BrainID
CSE 535: Mobile Computing

Giriraj NoLastName
MS CE, CIDSE
Ira A. Fulton Schools of Engineering
Arizona State University
ggirira2@asu.edu
1213350721

Sarvesh Patil
MS CE, CIDSE
Ira A. Fulton Schools of Engineering
Arizona State University
sspati13@asu.edu
1213353386

Nidhi Dubey
MS CE, CIDSE
Ira A. Fulton Schools of Engineering
Arizona State University
ndubey2@asu.edu
1213031246

Aravind Manoharan
MS CE, CIDSE
Ira A. Fulton Schools of Engineering
Arizona State University
amanoh10@asu.edu
1213332547

***Abstract*— An Android application that implements phone-based authentication. This application authenticates the user based on Brain Signal Data which is collected from a Neurosky device. The user has to enter userID, type of algorithm and where to authenticate the user. Our app has five inbuilt algorithms which will use the current users data against all users dataset present on the server (fog or cloud) and would authenticate the user.**

I. INTRODUCTION

Brain sensing and associated cognitive applications are fast becoming pervasive in nature due to the advent of wireless low cost easy-to-wear brain sensors such as Neurosky sensor that connect to mobile phones. This enables seamless access to a persons brainwaves which contains information that is unique to a person, nearly impossible to impersonate without invading personal space, and chaotic over time. In recent times, we have seen new biometric authentication systems like TouchID, FaceID and Iris scanner. Bank transactions are based on these new authentication systems. Hence we are experimenting on a new type of biometric authentication system based on EEG signals. We have taken reference from this paper[1] published by IMPACT Lab and we are trying to implement an algorithm called adaptive offloading which decides whether to implement the authentication in Fog server or Cloud server. Our intention is to make the authentication more faster with a performance improvement.

II. AUTHOR KEYWORDS

Authentication, Machine learning, Biometric authentication, EEG, Brain Signals, Fog, Cloud.

III. PROJECT SETUP & PIPELINE

Our architecture consists of three systems as shown in Figure 1. The configurations of these three systems are as follows:

1) Android device:
   Network: Local (Set up on the same server as the Fog server)
   Model: OnePlus 5
   OS: Android 8.1 (Oreo)
   Processor: Snapdragon 835
   RAM: 8 GB
   Application Name: EEG Group 31
2) Fog Server (Local Server)
   Network: Local (Set up on the same server as the Android device)
   Model: Acer Swift 3
   OS: Ubuntu 16.04 LTS
   Processor: Intel Skylake Core i7-6500U CPU
   RAM: 8 GB
3) Cloud server
   Type of server: AWS EC2 Instance (North California Data center)
   OS: Ubuntu 16.04 LTS
   Processor: AMD64
   RAM: 1 GB

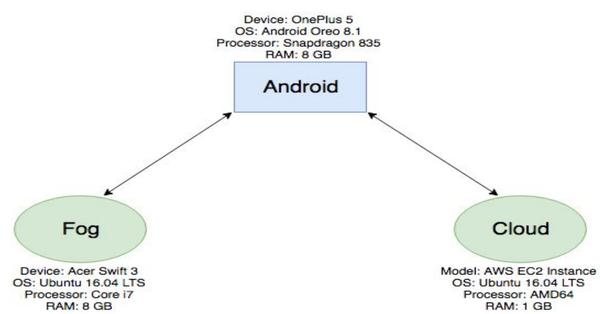


Fig. 1: Pipeline

IV. CLOUD AND FOG SERVER SETUP

To set up the web server on both the back-ends (Fog and Cloud), we have used nodejs server. A nodejs instance is running on both fog and cloud. We are using port 3000 in nodejs script to listen for any requests. The server request contains the URL to choose a particular back-end and there are two arguments which are:

1) Algorithm ID
2) User ID by name

The script running web server parses these arguments and executes the appropriate machine learning algorithm script with the users datafile. Then it returns back the result to the callee i.e. whether or not user can be authenticated.

V. DATA PREPARATION

We have collected the EEG data using the Neurosky device provided by the IMPACT lab. Each person has collected 2 minutes of data for 5 times for each mental state (open eyes and close eyes) for each day of the 7 days. The sensor collects data at the rate of 512 samples per second and since its a 2 minutes file, we have 61440 samples per file. Later these files are put together along with the class label of that persons name along with the mental state in which the data has been collected. Since these values are raw data from the sensor and also each instance has a dimensionality of 61440, we applied FFT feature extraction method in MATLAB and reduced the dimensionality to 720 features. This could reduce the computational time very much. Hence, our dataset looks like 70 instances per person and each instance has a class label of whose data and the type of mental state (some examples of the class label are aravind_open, sarvesh_close).

VI. IMPLEMENTATION

We have implemented the tasks for the project based on the category of the work which are as follows:

A. User Authentication

The user selects his user name from the drop down list. For each user name selected, the respective persons EEG data which be chosen. Then the user needs to select his/her choice of server for authentication. For better results, we suggest to choose automatic mode as it authenticates whichever server is faster.

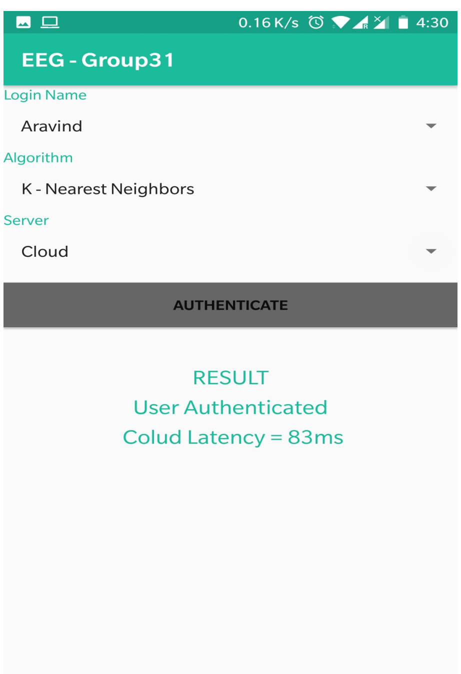


Fig. 2: Authentication Success

Once the authentication completes in any of the server, it returns a binary message to the phone whether the authentication is accepted or declined for the respective user. The success and the failure message is shown in Figure 2 and 3.

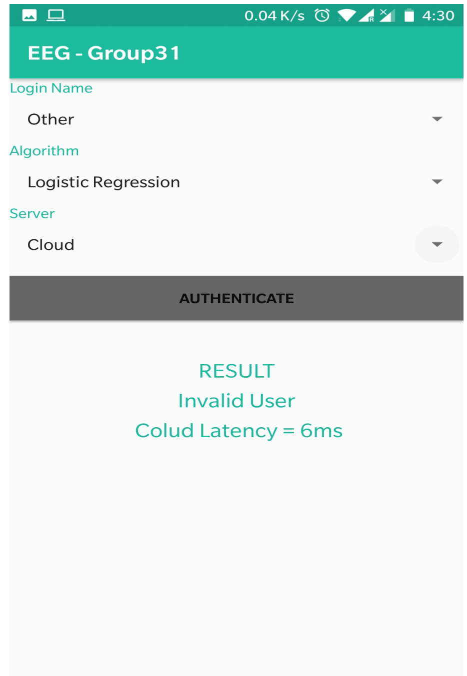


Fig. 3: Authentication Failure

B. Setting up Fog and Cloud server

As part of a back-end setup, we have created a cloud and a fog server for our application.

For cloud setup, we have created EC2 instance running Ubuntu 16.04 server in Amazon Web Services cloud in North California Data center.

The IPV4 public IP generated for this cloud server is used from our application to make requests to the web server. The authentication happens in Cloud as shown in Figure 4.

Similarly for fog setup we are using a machine connected in the same LAN network as that of mobile on which the application is being tested. This fog server is also running the same Ubuntu 16.04 OS like cloud. The authentication happens in Fog as shown in Figure 5.

To ensure all device are able to connect seamlessly to either of the back-end, we have opened TCP port 3000 in firewall to allow all connection coming to the web server.

On our back-end we have created a separate directory/server which holds all the code-base of machine learning algorithms which we are using along with the user dataset files and this directory also serves as route in URL requests that we make.

C. Developing an adaptive offloading algorithm to decide which server to use

The automatic mode uses an offloading algorithm to decide if a user should be authenticated on the cloud or fog

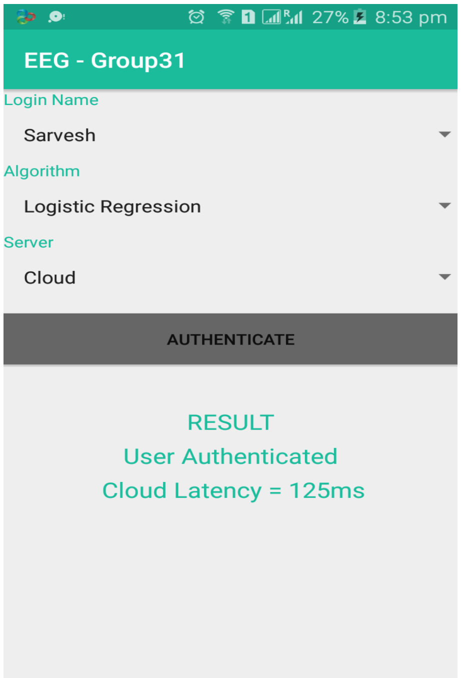


Fig. 4: Cloud Authentication

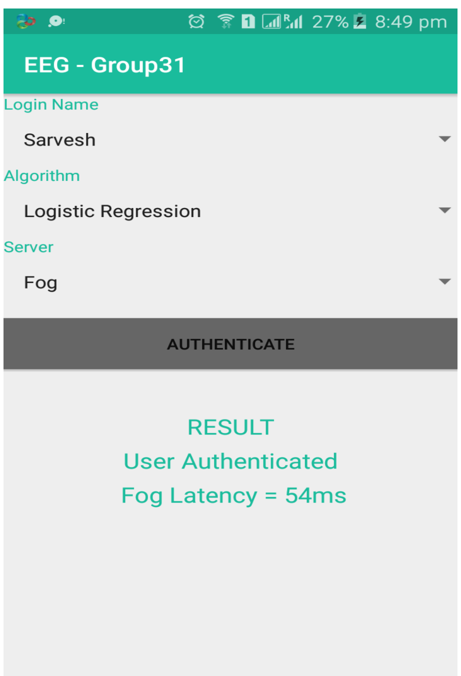


Fig. 5: Fog Authentication

server as shown in Figure 6. Both the server have required files such as the machine learning algorithms and the user dataset for the authentication. As and when a request is made from the phone, the required algorithm and the dataset will be called and the aggregate of the parameters are retrieved from cloud and fog server which let's the algorithm to decide which of the two servers should be used for authentication. After each authentication, the parameters are compared based on their latencies and whichever server has less latency gets selected.

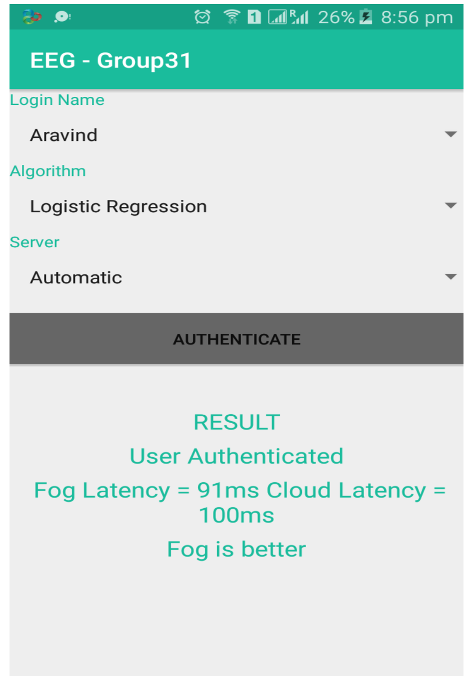


Fig. 6: Automatic mode

D. *Latencies comparison*

We have listed in Table I the comparisons of latencies on Fog, Cloud and Mobile . The Mobile can use only Support Vector Machines and Naive Bayes algorithm for the authentication. Both the server can use Support Vector Machines, Naive Bayes, Decision Tree, Logistic Regression and K - Nearest Neighbors for the authentication.

ALGORITHMS	FOG	CLOUD	MOBILE
Support Vector Machines	36	88	210
Naive Bayes	63	97	996
Decision Tree	41	96	N/A
Logistic Regression	54	85	N/A
K - Nearest Neighbors	34	84	N/A

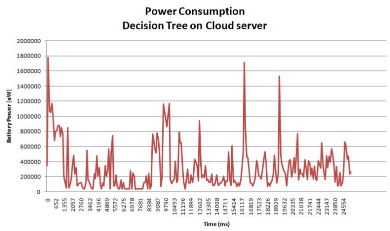
TABLE I: Latency Comparison in ms(time)

E. *Power Consumption comparison*

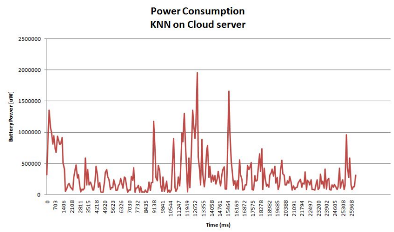
We have calculated Power Consumption for each algorithm and for each type of server and plotted against time. The power consumption vs Time plots for each algorithms which are implemented in Cloud are shown in Figure 7. The power consumption vs Time plots for each algorithms which are implemented in Fog are shown in Figure 8. The power consumption vs Time plots for each algorithms which are implemented in Mobile are shown in Figure 9.

VII. LIMITATIONS

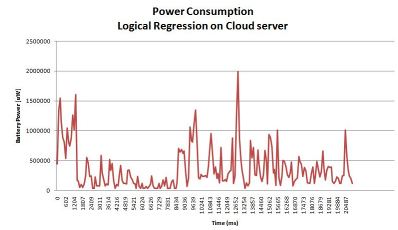
Our algorithm authenticates users based on the data which have been already collected from the same users. Hence our



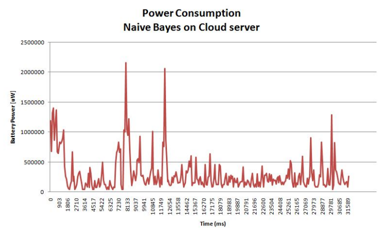
(a) DECISION TREE



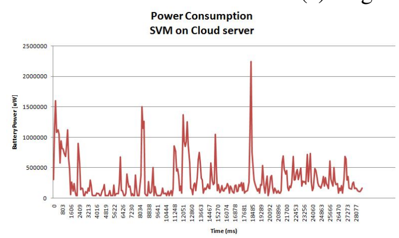
(b) KNN



(c) Logistic Regression

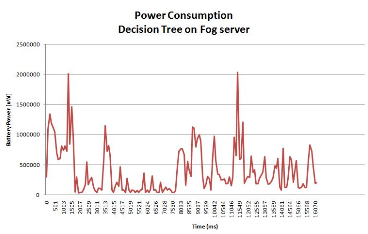


(d) Naive Bayes

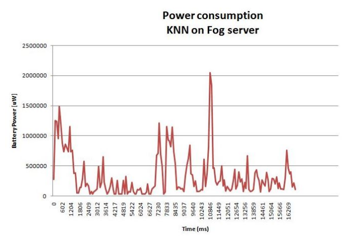


(e) SVM

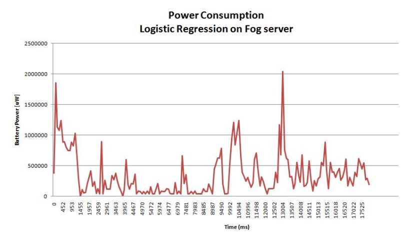
Fig. 7: Cloud



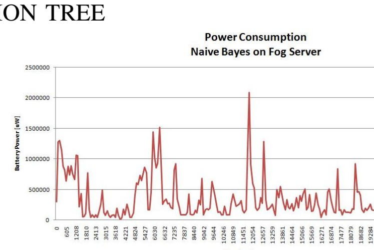
(a) DECISION TREE



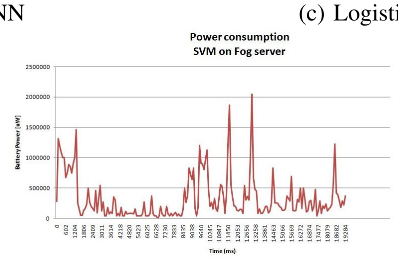
(b) KNN



(c) Logistic Regression

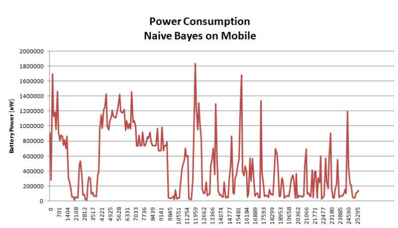


(d) Naive Bayes

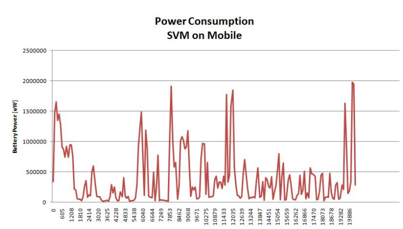


(e) SVM

Fig. 8: Fog



(a) Naive Bayes



(b) SVM

Fig. 9: Mobile

algorithm has been trained with the similar kind of data for each user, so we are able to get pretty good results. If we were able to test our authentication system with the real time data, then there might be more error than what we are getting now. Its just like the TouchID or FaceID which are available now, none of these systems are as accurate as entering a simple pass code, but these systems are available and are been widely used because they respond much quicker and are hassle free.

VIII. CONCLUSION

For this project, we have developed an application that authenticates a user based on his EEG signals. The authentication of a user is a time consuming activity and hence our algorithm intelligently decides whether to complete the authentication either on fog or cloud server based on the user data. We have conducted extensive experiments using commercially available wearable Neurosky brain sensor. Large scale implementation of this scheme in systems such as bank transactions, and border control is envisioned in the future. This application brings an intuitively new concept to the table which adds another method of biometric authentication to the list of secure authentications.

IX. COMPLETION OF TASKS

There were various tasks involved in this projects. The tasks were equally divided among the team members as listed in Table II

S NO.	TASK DESCRIPTION	ASSIGNEE
1	Collecting EEG signals from the Neurosky Device	All
2	Implementation of FFT feature extraction method	Aravind
3	Create a UI for User Authentiction	Aravind
4	Testing the UI4	Nidhi
5	Setting up & configuring Cloud server in AWS	Sarvesh
6	Setting up & Configuring Fog server in the same LAN network as that of smartphone	Sarvesh
7	Setting up NodeJs server on both the server to listen for requests from the applications	Sarvesh
8	Implemention of Nave Bayes and Support Vector Machines in Mobile	Giriraj
9	Implementation of Nave bayes in the servers	Aravind
10	Implementation of Support Vector Machines in the servers	Nidhi
11	Implementation of Decision Tree in the servers	Nidhi
12	Implementation of KNN in the servers	Sarvesh
13	Implementation of Logistic Regression in the servers	Giriraj
14	Calculate the accuracy for each algorithm in both the server	Nidhi
15	Calculate Latency for both Cloud and Fog servers	Giriraj
16	Record power consumption of all the algorithms in Fog, Cloud and Mobile	Aravind
17	Compare the latency of Cloud and Fog server	Giriraj
18	Implementing algorithm to decide where to authenticate the user	Giriraj

TABLE II: Tasks Description

ACKNOWLEDGMENT

We would like to thank Dr. Ayan Banerjee for giving us the opportunity to work on this project, which is valuable experience to us during the course of Mobile Computing. We also thank the IMPACT Lab and the TA, Junghyo Lee who provided the Neurosky device for data collection and helped us in collecting the data. We learned many concepts and technologies as part of this project.

REFERENCES

[1] Javad Sohankar, Koosha Sadeghi, Ayan Banerjee, and Sandeep K.S. Gupta. 2015. E-BIAS: A Pervasive EEG-Based Identification and Authentication System. In Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '15). ACM, New York, NY, USA, 165-172. DOI=http://dx.doi.org/10.1145/2815317.2815341

[2] https://www.physionet.org/pn4/eegmmidb/

[3] Xiao-Wei Wang, Dan Nie, Bao-Liang Lu. 2014. Emotional state classification from EEG data using machine learning approach.

[4] Klaus-Robert Mller, Michael Tangermann, Guido Dornhege, Matthias Krauledat, Gabriel Curio, Benjamin Blankertz. 2008. Machine learning for real-time single-trial EEG-analysis: From braincomputer interfacing to mental state monitoring.

[5] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June 13).

[6] Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R. BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. IEEE Transactions on Biomedical Engineering 51(6):1034-1043, 2004. [In 2008, this paper received the Best Paper Award from IEEE TBME.]