E-Bias - A Next Generation User Authentication System using Brain Activity

Tarun Chinmai Sekar
1213181929
tsekar@asu.edu

Soundarya Sankarasubramanian
1213119620
ssanka24@asu.edu

Sandhya Chandrasekaran
1215159426
schand61@asu.edu

Aravind Shanmugam
1213174948
ashanmu5@asu.edu

***Abstract*—Authentication of users is a very important part of today's mobile focused world. Traditional authentication mechanisms such as pass code, pin code etc are flawed and are prone to breaches. Bio metric based authentication systems such as Facial recognition, Fingerprint recognition have become commonplace these days. In this project, we present a novel authentication mechanism that identifies the user based on Electroencephalogram data (EEG) collected from a Neurosky device. The system uses various machine learning models to learn the user's characteristics from the EEG signal. The processing of signals is offloaded to a Cloud Server or a Fog Server. The system has an algorithm to determine which server to offload the process to. The system also allows users to register themselves, triggering the application to train the models with the new user's data.**

***Index Terms*—Biometric authentication, EEG, Offloading techniques**

I. INTRODUCTION

In the recent years, with the rapid evolution of technology, smart phones have become so vital that it has become a critical part of one's life, storing sensitive information of users. So, it is of utmost importance to safeguard the privacy of users. In order to provide security while not impacting ease of use, several new authentication methods such as Biometric authentication using facial recognition, gesture recognition, fingerprint recognition etc have been developed. This project presents a next generation user authentication system that authenticates users based on their thinking patterns. The data is obtained from a neurosky device, and is aggregated by the mobile device. The patterns in the user's brain signals are identified and learned using machine learning.

The user logs into the application by providing his user name, .edf file and selects one of the machine learning algorithms for processing. We have implemented an algorithm which, after a significant amount of training, decides between fog server and cloud server to perform authentication The coding is done in Python so Django framework has been used to handle requests and responses to and from the device.

II. PROJECT SETUP AND PIPELINE

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

Android Device

Network	Local
Model	One Plus 6T
OS	Android 9.0
Processor	Qualcomm Snapdragon 845
RAM	8 GB

Fog Server (Local Server)

Network	Local
Model	Macbook Air
OS	MacOS Mojave 10.14.3
Processor	Intel i5 Dual Core
RAM	8 GB

Cloud Server

Server	Google compute Engine
Model	n1-standard-8
OS	Ubuntu 18:04 10.14.3
Processor	8 vCPUs
RAM	30 GB

III. CLOUD AND FOG SERVER SETUP

The machine learning model interfaces with the android app through a Django web application. The Django web app is written in python and handles the requests from the app, parses them and passes it to the model. The prediction results from the model are transformed to Json Objects and sent to the app. Both the cloud and fog servers are setup identically. The servers are configured to listen on port 8000 for incoming connections. Both servers are configured with python 3.7, and the following python packages are installed.

- Django v2.2
- django-cors-headers v2.5.3
- djangorestframework v3.9.3
- numpy v1.16.3
- pandas v0.24.2
- pyEDFlib v0.1.14
- python-dateutil v2.8.0
- pytz v2019.1
- scikit-learn v0.20.3
- scipy v1.2.1
- six v1.12.0
- sqlparse v0.3.0

IV. DATA PREPARATION

The Electroenchephalogram (EEG) data is collected from 64 different electrodes of Neurosky device. This creates 64 channels for the data and data is in the .edf format. Before extracting the features, cleaning up of data or data preprocessing is performed using zero mean and unit variance. Then, Fast Fourier Transform (FFT) is performed on the data and it is labelled based on what subject it belongs to. Around 80 percent of the data is taken as training set. Typically, out of 14 runs for each user, 12 or 13 runs are taken as training set while the remaining are considered for test set. After applying FFT, the features are passed onto four different machine learning models. The algorithms implemented are SVM, Naive Bayes, KNN and Stochastic Gradient Descent (SGD).

V. IMPLEMENTATION

The following tasks has been implemented as per the project specifications:

A. *User Authentication*

The application has a login functionality and a registration module. The login interface allows the user to enter his user name, as showed in Fig 1. As we do not collect data from Neurosky device directly, the user inputs .edf file during login. In addition to that, user chooses a machine learning algorithm from the four techniques implemented. In addition to the requirements, we have also implemented a registration functionality where the user can upload the edf file, along with their name. The uploaded sample is added to the training dataset, and the model is re-trained with the new sample. This enables the application to learn better and improve its accuracy. Moreover, after the authentication is complete, the results are stored in the local database, and is charted as displayed in Fig. 2.

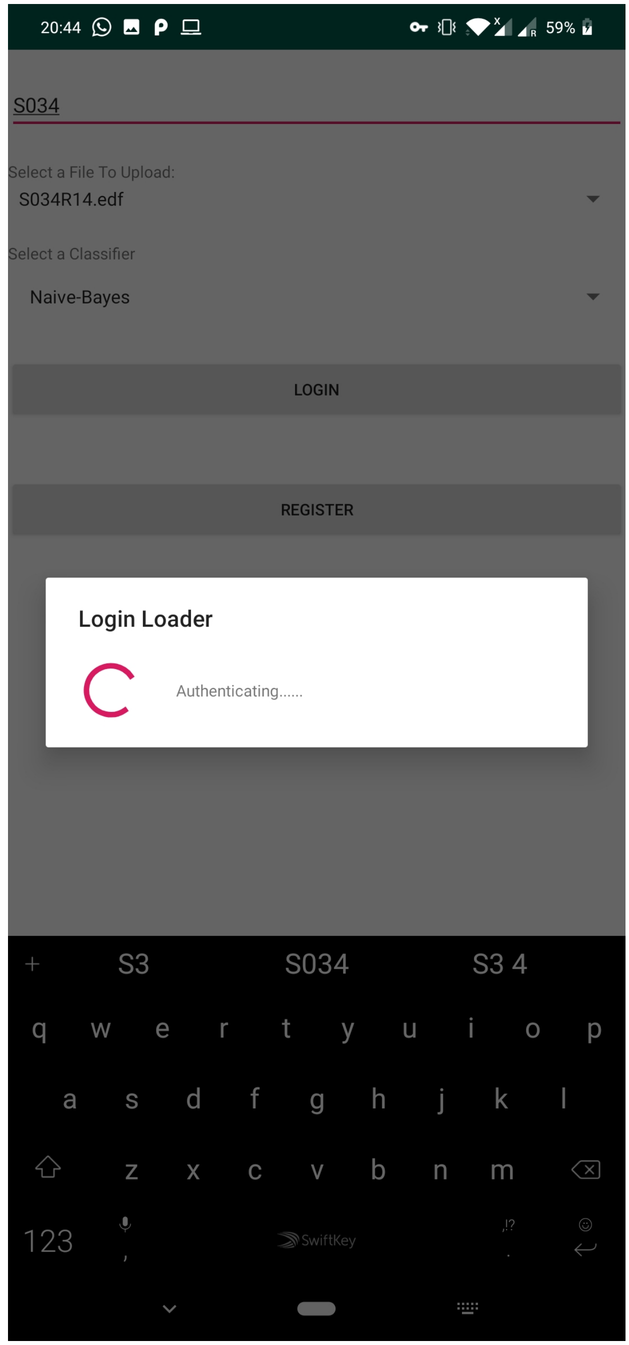

Fig. 1. Login Screen

B. *Setting up fog and cloud servers*

The fog server is setup on a laptop on the same network as the device. The cloud server is setup on Google cloud. The Django app is run as a service on both the devices, and Nginx is used to reverse proxy the requests to the Django backend. The model is first trained on both the cloud and fog servers and the models are saved to binary files. When the user attempts to authenticate, the saved model is loaded, and the prediction is performed. Once the prediction is performed, the uploaded file is removed and the results are returned to the app. Python virtual environment is used to setup an identical environment on both the servers.

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

When the application is launched, an http request is sent to both cloud and fog servers. This response is parsed and we calculate the response time (latency) for both servers. This latency is stored and is considered as the main criteria for selection between fog and cloud server. The results of the selection, as well as the metrics are stored in a Realm database Object on the device. After authentication is complete, the time taken to authenticate etc are also stored in the realm object, and dumped to the db. The algorithm checks for presence of any previous database dump. If such a dump exists, then the average execution time for both cloud and fog servers is calculated using those data. When average execution time and latency of cloud server is less than that of fog server, then cloud server is selected. In all other cases, fog server is chosen. If a database dump doesn't exist, we compare only the latencies. If cloud latency is less than latency of fog server, then cloud server is chosen to perform authentication. The choice of server is displayed to the user as shown in Figure 3.

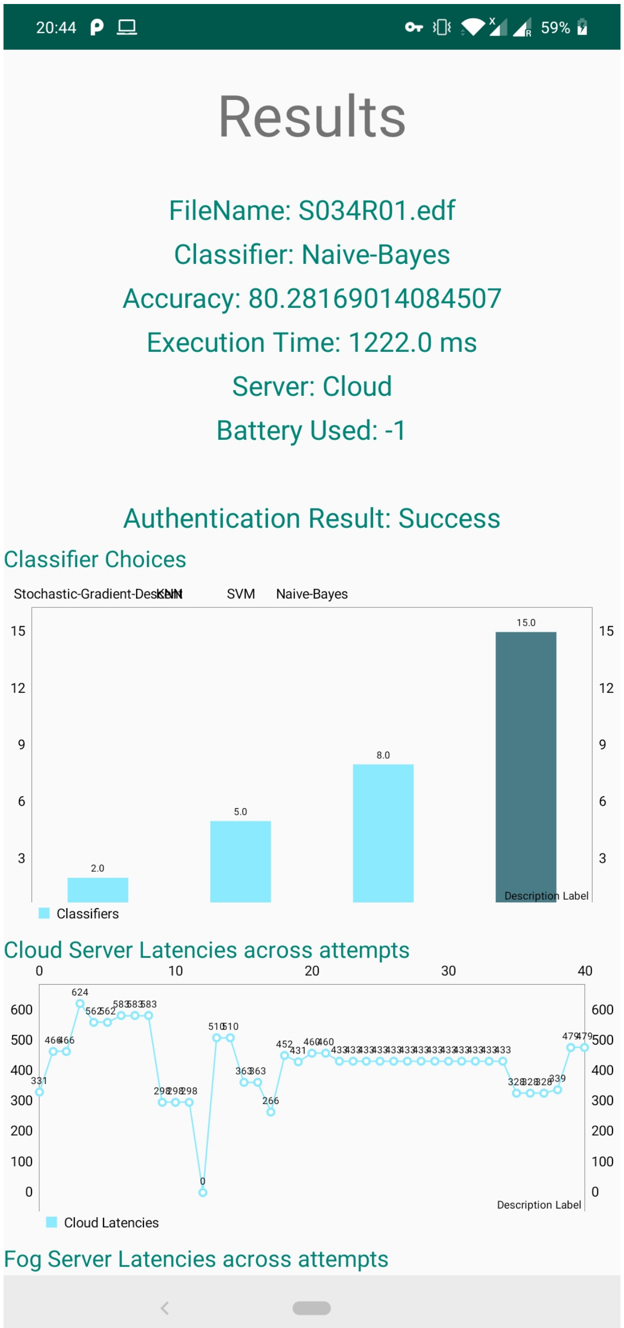

Fig. 2. Login Results

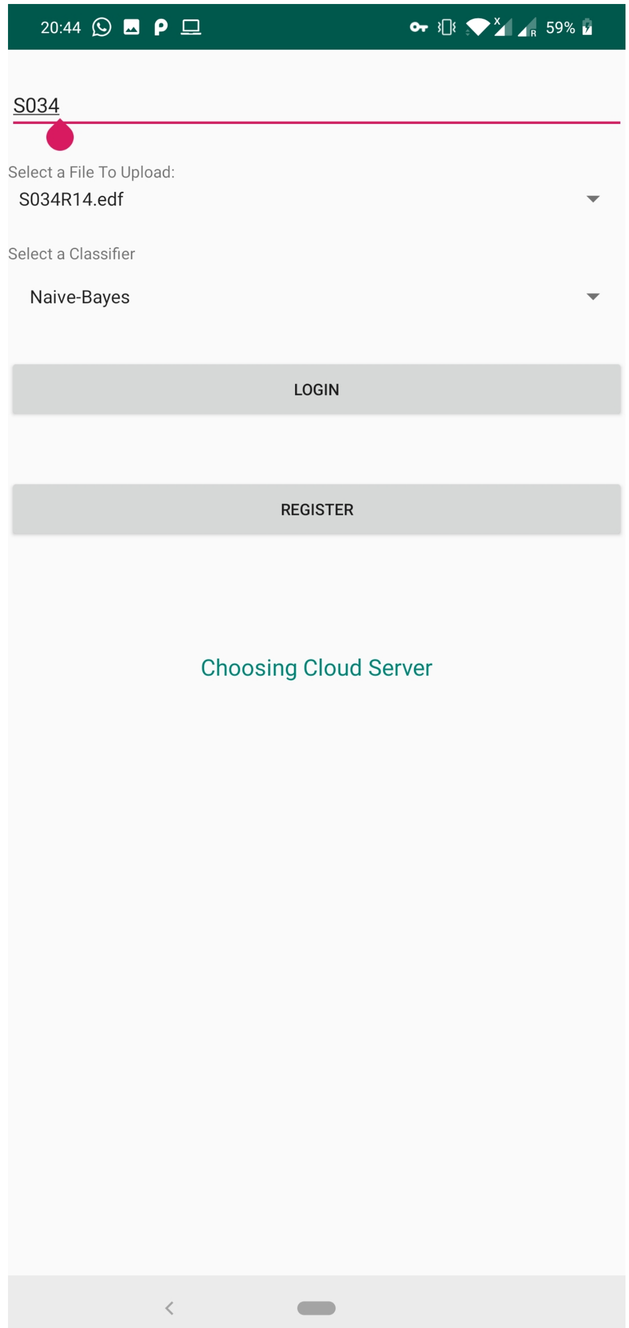

Fig. 3. Server Choice Displayed

D. Latencies comparison

We have compared the latencies of the two servers, Cloud and Fog and have visualized the result below in the Figure 4. It can be seen that the latency of the fog server is almost always lesser than that of the cloud server.

E. Execution Time Comparison

Comparing the execution time for both the cloud and fog servers, it can be seen from Figure 5 that the execution time of cloud is slightly better than that of fog, despite having a higher latency.

F. Power usage Comparison

Running these algorithms on the cloud and fog servers, there was no noticeable power consumption difference seen on the android device.

G. Accuracy Comparison

The different algorithms perform with different accuracies, and the highest accuracy achieved by the system is 92.8%.

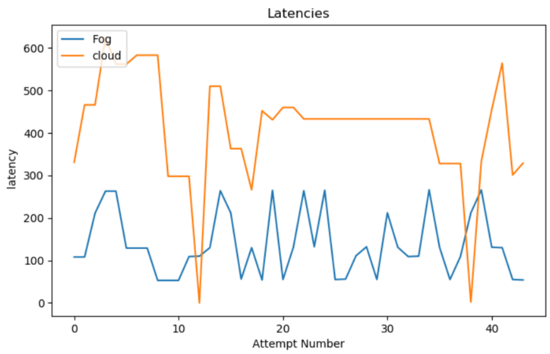

Fig. 4. Comparison of latencies

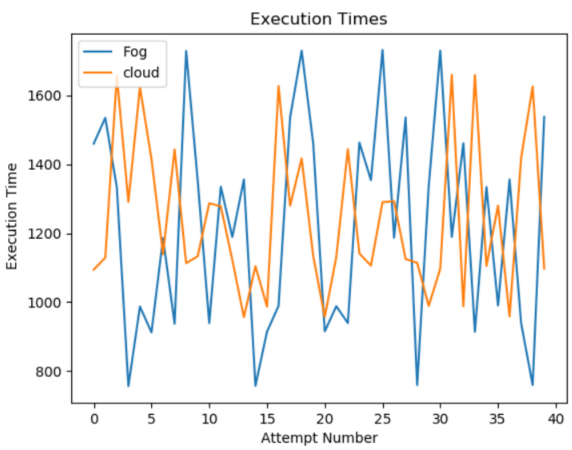

Fig. 5. Comparison of Execution Times

VI. LIMITATIONS

The project suffers from a usability limitation. The main issue is that for this to become viable, the users must have access to a neurosky sensor and be able to authenticate it. This will cause many users to avoid this type of authentication due to the additional cost incurred. Moreover, the system has to retrain for every new user registered. This is not scalable, as a heavy influx of users to the application will cause the training pipeline to choke, and overload the servers. Moreover, the data available to the system is very limited and is hard to find all patterns with such limited information.

VII. CONCLUSION

The system provides a novel way of authenticating users, and opens up exciting new applications. The EEG based authentication can also be used to augment existing authentication mechanisms until the system becomes robust and stable.

VIII. DIVISION OF TASKS

Below is the list of tasks divided among team members.

Task 1	Understand data such as format and distribute the data to each member	Sandhya
Task 2	Use pre-processing algorithms provided by IMPACT lab to clean data	Soundarya, Tarun
Task 3	Write code to extract frequency domain feature from EEG signal	Soundarya, Aravind
Task 4	Compute the Fast Fourier transform of the Signals	Tarun, Soundarya
Task 5	Write Naive Bayes classifier based algorithm	Sandhya
Task 6	Evaluate its accuracy	Aravind
Task 7	Develop Login UI	Tarun
Task 8	Develop Register UI	Sandhya
Task 9	Develop Server Backend for Registration	Tarun, Soundarya
Task 10	Backend process for mobile application	Tarun, Sandhya
Task 11	Evaluate its execution time and power consumption	Aravind, Sandhya
Task 12	Develop SVM model	Tarun, Sandhya
Task 13	Develop KNN model	Soundarya, Tarun
Task 14	Develop SGD model	Sandhya, Aravind
Task 15	Setup cloud server and Fog server	Tarun
Task 16	Figure out the measured communication delay between fog, cloud, and the mobile	Sandhya,Soundarya
Task 17	Develop algorithm for best offload tactic taking in the account for execution time, power consumption, and communication delay	Tarun, Soundarya
Task 18	Compare latency authentication for fog VS Cloud VS Mobile	Aravind, Sandhya
Task 19	Compare power consumption for fog VS Cloud VS Mobile	Aravind, Tarun
Task 20	Testing end to end Integration and Authentication flow	Tarun, Aravind

REFERENCES

[1] Sohankar, Javad Sadeghi, Koosha Banerjee, Ayan K.S. Gupta, Sandeep. (2015). E-BIAS: A Pervasive EEG-Based Identification and Authentication System. 10.1145/2815317.2815341.

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

[3] https://www.mathworks.com/matlabcentral/answers/223973-how-to-normalize-60x3-vector-to-zero-mean-and-unit-variance

[4] https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/

[5] Wang, Xiao-Wei Nie, Dan Lu, Bao-Liang. (2014). Emotional state classification from EEG data using machine learning approach. Neurocomputing. 129. 94106. 10.1016/j.neucom.2013.06.046.

[6] Mller, Klaus-Robert Tangermann, Michael Dornhege, Guido Krauledat, Matthias Curio, Gabriel Blankertz, Benjamin. (2008). Machine learning for real-time single-trial EEG-analysis: From brain-computer

interfacing to mental state monitoring. Journal of neuroscience methods. 167. 82-90. 10.1016/j.jneumeth.2007.09.022.