Gait Authentication using Deep Neural Networks

Alexandru LOGHIN, Ştefan Mihai STRUGARI, Diana BOLOCAN
Department of Computer Science Alexandru Ioan Cuza, Iaşi
E-mails: loghinalexandru61@gmail.com, stefanmihaistrugari@gmail.com,
dianabolocan.db@gmail.com

Abstract. [small presentation of the article]

Keywords. *Machine learning, Deep Neural Networks, Android, Gait Authentication, Python*

I. Introduction

With advances in miniaturization techniques, performance of the mobile and portable devices is rapidly increasing. This enables to use such devices not only as communication tools but also in an applications from a security standpoint.This means that they can store and process valuable information such as financial or private data.

This paper intends to offer a way to protect against mobile devices theft using the data registered when the owner of the device walks. We desire to achieve this by creating a small android app, all open source in order to make available this technology to any user with an android device running Lollipop 5.0 and higher. In order to achieve this we also have a small constraint, the mobile device used in the gait authentication process has to have a built in steps counter in order for us to determine when the user is walking and then record the accelerometer data. The user can disable and enable this third-layer security feature at will and it can add as many owners of the device as he wishes. This paper presents a biometric authentication of individuals based on their gait. Gait is a person’s manner of walking in order to determine if the user’s device has been stolen. Unlike many other papers we tried to achieve this by using a regular day-by-day phone to prove the viability and ease of access to such biometric recognition technology.

II. State-of-the-art

1. What did others?

There are multiple papers that involve this kind of authentication. Some of them state that they used a device strapped to the lower leg of a person to measure the acceleration on all of the three axis in order to authenticate someone. More recent papers have been using smartphones to gain the accelerometer data since it is already integrated in the phone, but there are no commercial use application that use this kind of technology in order to add another security layer for the ordinary user.

2. Names in the field

The first person to introduce the idea of biometric gait authentication was Ailisto, H. in his most known paper Identifying People from Gait Pattern with Accelerometers done in 2005. Since then other big names have emerged in this field like Thingom Bishal Singha, Rajsekhar Kumar Nath and Dr. A. V. Narsimhadhan.

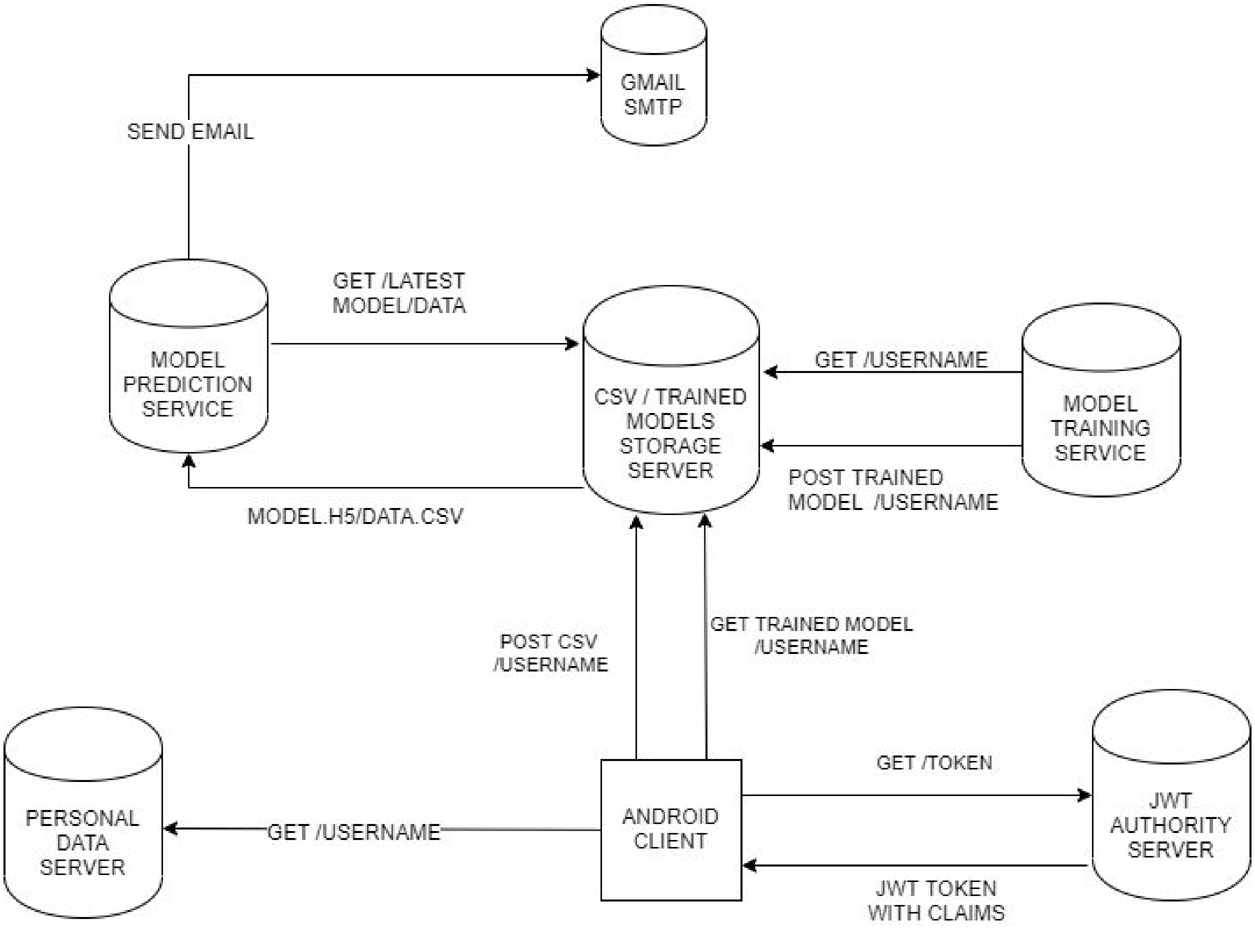
III. Our solution

1. Software Architecture

We opted for an SOA architecture with modular components in order to ensure scalability and performance. We split the whole monolithic ideea application into three important components.

- Data Collection : made possible by an android client that relies on the accelerometer and the step detector sensor
- Model Training : done by an independent service based on all the new and previous data collected for a certain user
- Model Prediction : also done by an independent service that runs the collected data on the previously mentioned trained model and takes action based on the result.

The architecture diagram below illustrates the application layout and also the modularity of the whole system. This ensures that on any number of users the whole aggregate will satisfy performance constraints.



2. Components

The most important component mentioned is the **Android Client**, build in order to collect the data in the most simple and convenient way. The client is available to use for anyone with an android phone and some basic sensors. Unfortunately because of some android limitations we could not make the whole system to work offline which would have greatly increased the usability of this approach.
From a security standpoint we use a service that generates **self signed tokens** in order to authenticate and authorize. As such only the user that has the phone and knows the credentials of the currently active account on the device can turn on/off the gait biometric authentication.

In order to ensure scalability because of the high volume of data we use a **real time communication queue** in order to notify the two workers. This approach allows us to add an infinite number of said workers with automatic load balancing from the said **queue**. Also the process of **model training/model prediction** is independent per each user so these processes can be done concurrently further increasing robustness and scalability.

The **Data Storage Server** is a **RESTful API** that stores the two major artefacts of our system, namely:

- The input data – represented by the CSV files that store a series of 3D coordinates detected by the android client running on the user's device. In this case, the responsibility of the server is to further provide this data to the **model training service**.

- The trained models – these are the files that contain the models obtained from **model training service.** This data will be directed into the **model prediction service**, to perform the actual gait authentication prediction for the specific user.

The files are stored on the local file system.

The aforementioned **real time communication queue** has a very important role in the flow of the data throughout the components of our application, standing at the base of our **event driven architecture**, assuring that the services are well **decoupled** or can be very easily **distributed** and **scaled**.

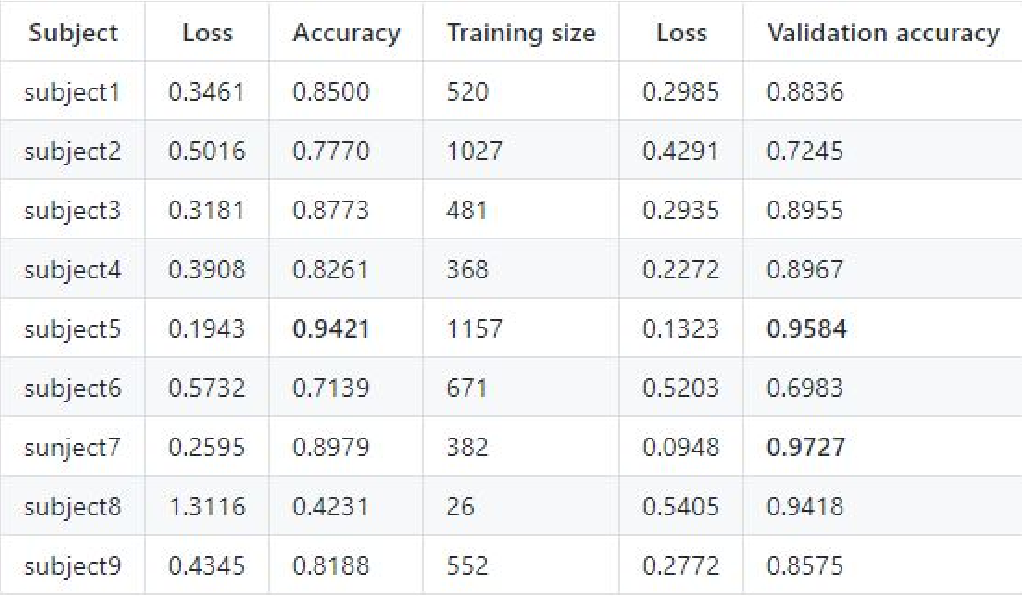
3. Data Preprocessing

Data Processing is done by running a python script with feature extraction tools that takes as arguments --raw_data_dir the path to the directory that contains csv files to be processed, --save_processed_data the path to the directory in which the results are saved as pickle files. The features contained by a sequence of tuples (**mean**, **median**, **magnitude**, **peak distance**) for consecutive intervals of input data in time. This features will be further used for the training and prediction fazes.

4. Gait Recognition Algorithm

The Gait Recognition algorithm is purely based on the execution of a well regularized feedforward neural network, by running another import python script that generates and traines the new model that takes as arguments --save_models_dir the path to the directory in which the model will be saved as .h5 where the username will be taken from --processed_data_dir the path to the directory that contains user directories of processed data.

IV. Results



Subject	Loss	Accuracy	Training size	Loss	Validation accuracy
subject1	0.3461	0.8500	520	0.2985	0.8836
subject2	0.5016	0.7770	1027	0.4291	0.7245
subject3	0.3181	0.8773	481	0.2935	0.8955
subject4	0.3908	0.8261	368	0.2272	0.8967
subject5	0.1943	**0.9421**	1157	0.1323	**0.9584**
subject6	0.5732	0.7139	671	0.5203	0.6983
sunject7	0.2595	0.8979	382	0.0948	**0.9727**
subject8	1.3116	0.4231	26	0.5405	0.9418
subject9	0.4345	0.8188	552	0.2772	0.8575

Fig [no]. Training results on validation size of 842 rows

We can see from the table that we managed to obtain a good accuracy on validation given the number of subjects. All the data used in this paper has been collected from the nine subject while they were walking on a flat surface. The collected data ranges from a 10 to 45 minute session walk, with the latter giving a better understanding of the subject's walking patterns.
The aggregate also is capable of dynamic adaptation, since the worker that handles the model training will always incorporate the new and old data gathered from the subject in a new and better describing model in order to keep the accuracy high and handle any noise coming from the environment, like running.

V. Other research

As stated in the cited papers below the usual approach for this authentication method is to extract data from the embedded accelerometer sensor from a user

phone and pass this data to a machine learning algorithm in order to determine if the user is who he claims he is (authorization) or to determine who the user is (authentication). As an article stated based on the experimental results from building a Random Forest based on both time and frequency, the resultant model delivers an accuracy of 0.9679 and Area under Curve (AUC) of 0.9822.
Some of the methods used include recording the data with a motion sensored strapped on the ankle, gathering data only on the two directions of the orthogonal axis. More recently there has been an interest in this field, many papers stating that a phone was used in order to record the data at a sample of 100 Hz per second, as we did in this experiment. However the data collection was only part of the problem, some of the papers state that multiple approaches for training on the said data was used such as SVM, Random Forest, Histogram Frequency and many more, including Neural Networks.

VI. Future work

Current limitations of the application are caused by some android limitations that won't allow us to make a fully working offline version of the system. In the future there should be an attempt to make a fully offline version of this application, compliant with the limitations determined by the phone's system.
Another aspect that is worth mentioning is that because we only used the accelerometer data recorded from the user's phone we needed to set a baseline of regulations such that we do not introduce noise in the captured data. These regulations stipulated how the user should hold their phone in the pocket in order for the android client to record unaltered accelerometer data. This rule was set in place for simplicity, in a future work one may also record the gyroscope data and attempt to normalize the accelerometer data with a rotation matrix such that phone orientation does not introduce noise in the data.

Regarding the model and the training process we need to mention that further optimizations can be done in order to speed up the whole training timing with better result and with a smaller sample data from the subject. For the moment we used a simple deep neural network but we mentions that a dynamically expandable network could help if there is a bottleneck regarding the NN due to the high number of users and features..

VII. Conclusions

Even thought we had a roughly small sample of data we managed to get great results on model accuracy. We wanted to prove that a simple smart phone can be used to add another layer of security besides the fingerprint, facial recognition and various type-in methods. Further investigation is required in order to determine if this method of biometric authentication is sufficient by itself for the smartphone security but as a third-party additional security it is definitely viable.

VIII. Biography

Singha, Thingom & Nath, Rajsekhar & Narsimhadhan, A.. (2017). ***Person Recognition using Smartphones' Accelerometer Data***.
Ahmad, Muhammad & Alqarni, Mohammed & Khan, Asad & Khan, Adil & Chauhdary, Sajjad & Mazzara, Manuel & Umer, Tariq & Distefano, Salvatore. (2018). ***Smartwatch-Based Legitimate User Identification for Cloud-Based Secure Services***. Mobile Information Systems. 2018. 1-14. 10.1155/2018/5107024.

IX. Links

Accelerometer Biometric Competition
RecuPlots and CNNs for time-series classification