Walking Style Authentication(WSA)

Jianbo Pei
Computer Engineering

Jingyuan Wang
Computer Engineering

Bowen Sun
Electrical Engineering

Abstract

As our project name suggests, the idea of our project is to design a new method of biometric authentication based on the unique walking style of each individual. Although there already exists many biometric authentication methods, most still need to have users perform some actions to do the authentication. Our project is intended to design a new method so that users do not need to perform any action for authentication but still achieve reliable security. This paper is divided into several sections, such as introduction, related work, methodology, results and conclusions. It mainly covers the details of our idea, the methodology we constructed to achieve our goal, the results with our current approaches, and conclusion with the future work to complete our authentication method.

1 Introduction

Research on new methods of authentication has been conducted for a very long time and still continues. Authentication is a big topic in the security domain. In Security in Computing, the author states that authentication is used to prove the asserted identity. Also, the author summarizes that we could employ the authentication based on three qualities, which are something the user knows, something the user is, and something the user has. One example of something the user knows is the password created by users. For something the user has, we can use tokens. One example for tokens is the credit card with magnetic strip. Something the user is would mainly focus on the biometric properties of human body. Since mobile devices, especially smartphones, have been growing so rapidly, new methods of biometric authentication have been developed for the purpose of better security and convenience. We have seen smartphones equipped authentication with fingerprint, hand geometry, retina and iris, facial features and so on. The biggest advantage of biometric authentication over others is that the biometric features are with you all the time, cannot be lost, stolen, and always available.[1]

Our project is to develop a new biometric authentication method based on the unique walking style of each individual. As we know that it is highly unlikely two persons would walk like the same. Therefore, we will extract the useful features from human walking movement and develop a reliable authentication method using machine learning algorithms, such as Linear Regression and SVM. In this paper, we will cover related works of our project, the approaches we took to developing the authentication method, the corresponding results from the progress of our approaches, and future works we need pursue to strengthen our authentication method.

1.1 Flow Chart

Before directly jumping into the detailed description of our project, we first look at a flow chart which clearly layouts our project from an abstraction level.

The Figure 1 below shows the flow for the system initialization. In order to make our project work in real life, we need to build our system in the first place. First, we need to find a way to collect the training and testing data for later machine learning algorithms. Second, after we have obtained the data we need, we continue to process data and add labels in order to fit the machine learning algorithms. Third, once we have everything we need for machine learning algorithms, we will build machine learning models from the data we collected. Last but not least, we need to test our system for better accuracy and reliability.

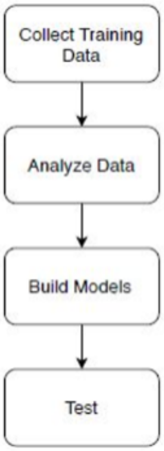


Figure 1: System Initialization

2 Related Work

Supervised learning is where you have input variables (X) and output variables (Y) and you use an algorithm to learn the mapping function from the input to the output.The goal is to approximate the mapping function so well that when you have new input variables (X) that you can predict the output variables (Y) form that data.[2]

Classification problem is one standard formulation of supervised learning: the learner is required to learn (to approximate the behavior of) a function which maps a vector into one of several classes by looking at several input-output examples of the function. Inductive machine learning is the process of learning a set of rules from instances.

Supervised learning can be applied to predict things, like house price. Which means if you know the factors which can determine the house price and you can build the model predicting the house price by using supervised learning.

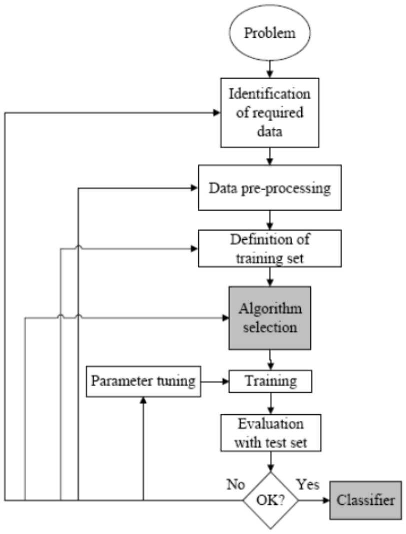


Figure 2: The Processes of Supervised Learning

Because, in our project, after the authentication system gains the accelerators' data of the person, it can predict the label of the person so that it can know whether the person is in the dataset of the system and then do the correspondent action. It can also be seen as a classifier, that is, what the system needs to learn and do is to classify the dataset into two different parts one of which is the data with the specific feature that can allow the person who have the feature into the gate and another one is the data with no specific feature.

Thus, supervised learning is the proper algorithm to build the model of our walking style authentication.

3 Methodology

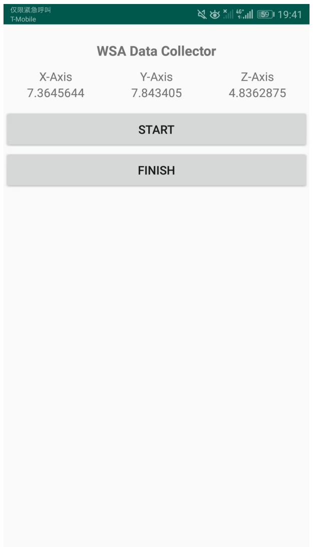
In this section, we will cover the details of approaches as we develop our authentication method. First, we will describe what features we discovered as useful, how we extract and process these features. Second, we will use linear regression algorithm to develop machine learning models from these features.

Third, we will use another algorithm SVM in order to compare these two machine learning algorithms.

3.1 Feature Extraction

The features we planned to extract are the acceleration rates from x, y, and z three axis and steps count. However, in later building our machine learning models, because steps count for working at a time is almost identical, the steps counts do not make much difference on the results. Therefore, we only use acceleration rates in later data processing.

In order to extract acceleration rates data, we designed an android application using Android Studio environment based on an existing project[3]. The App is installed on an android phone with an accelerometer sensor built in. With this app, the acceleration rates data can be saved on the phone for every time walking measurement. Once start button pressed, the accelerometer service routine will be invoked, and the data will be measured and saved in an open file. Once finish button pressed, accelerometer service routine will be deactivated, and the data file will be closed. Figure 3 shows the interface of our App.

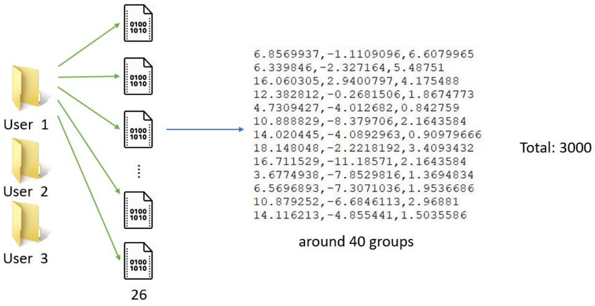


WSA Data Collector

X-Axis	Y-Axis	Z-Axis
7.3645644	7.843405	4.8362875

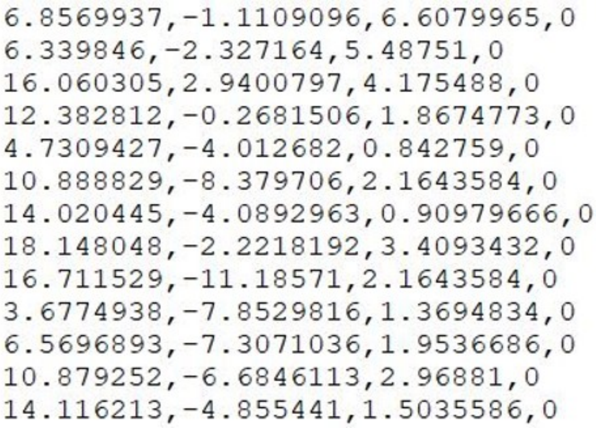
Figure 3: App Interface

In our project, we treat one group of data as each reading on x, y, and z three axis. For later building our models, we collected around 3,000 groups of data as our dataset. The dataset combines each of our three group members holding the phone to walk from one end of a certain 10-meter path to the other end for total 26 times. Each time generates one data file and each data file contains about 40 groups of data.Figure 4 shows the abstraction structure of data files.



```
6.8569937,-1.1109096,6.6079965
6.339846,-2.327164,5.48751
16.060305,2.9400797,4.175488
12.382812,-0.2681506,1.8674773
4.7309427,-4.012682,0.842759
10.888829,-8.379706,2.1643584
14.020445,-4.0892963,0.90979666
18.148048,-2.2218192,3.4093432
16.711529,-11.18571,2.1643584
3.6774938,-7.8529816,1.3694834
6.5696893,-7.3071036,1.9536686
10.879252,-6.6846113,2.96881
14.116213,-4.855441,1.5035586
```

Figure 4: Data Files Abstraction



```
6.8569937,-1.1109096,6.6079965,0
6.339846,-2.327164,5.48751,0
16.060305,2.9400797,4.175488,0
12.382812,-0.2681506,1.8674773,0
4.7309427,-4.012682,0.842759,0
10.888829,-8.379706,2.1643584,0
14.020445,-4.0892963,0.90979666,0
18.148048,-2.2218192,3.4093432,0
16.711529,-11.18571,2.1643584,0
3.6774938,-7.8529816,1.3694834,0
6.5696893,-7.3071036,1.9536686,0
10.879252,-6.6846113,2.96881,0
14.116213,-4.855441,1.5035586,0
```

Figure 5: Labelling Example for Linear Regression

The raw data we collected cannot directly be used by linear regression and SVM, so we wrote our Python programs to integrate the data and add labels. Labels we used for our project is 1 and 0. 1 represents the users who are in the system and 0 represents the users who are not. For linear regression, we combine the data with labels and divide for training and testing. For SVM, we combine the data which has label 1 and data which has label 0. One example shown as Figure 5 is the result after adding labels for linear regression. '0' after each line is the label we added.

3.2 Linear Regression

Linear regression is perhaps one of the most well-known and well-understood algorithms in statistics and supervised learning[4].The representation of linear regression is an equation that describes a line that best fits the relationship between the input variables (X) and the output variables (Y), by finding specific weightings for the input variables called coefficients (θ). For example, $\hat{Y} = \theta_0 + \theta_1 X$.

For every change in X, Y will change by the same amount no matter how far along the line you are. The X is transformed by the same θ_0 and θ_0 at every point.Besides, the reason why we are dealing with $\hat{Y}$, an estimate about the real value of y, is because linear regression is a formula used to estimate real values, and error is inevitable.

Linear regression with only one input variable is called Simple Linear Regression. With more than one input variable, it is called multiple linear regression. In our project, we use multiple linear regression algorithm, because the input variables (X) are more than one. As we can see from Figure 6, these three columns represent as accelerators from different directions (x,y,z).

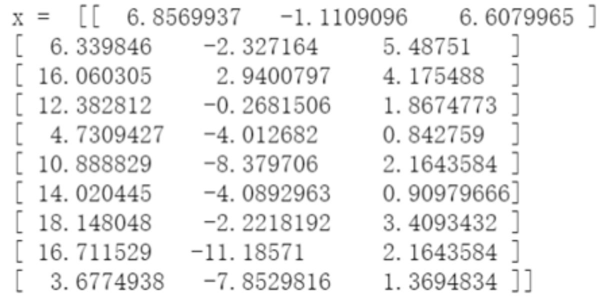
Thus, the mapping function is represented as:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$h_\theta(x) = \theta^T X$$

There are two significant concepts in linear regression which make the predict more accurate.

3.2.1 Cost Function

The cost function helps us to figure out the best possible values for θ which would provide the best fit line for the data points. Since we want the best values for θ, we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.



```
x =  [[  6. 8569937    -1. 1109096     6. 6079965 ]
 [  6. 339846     -2. 327164      5. 48751    ]
 [ 16. 060305      2. 9400797     4. 175488   ]
 [ 12. 382812     -0. 2681506     1. 8674773 ]
 [  4. 7309427    -4. 012682      0. 842759   ]
 [ 10. 888829     -8. 379706      2. 1643584 ]
 [ 14. 020445     -4. 0892963     0. 90979666]
 [ 18. 148048     -2. 2218192     3. 4093432 ]
 [ 16. 711529    -11. 18571       2. 1643584 ]
 [  3. 6774938    -7. 8529816     1. 3694834 ]]
```

Figure 6: Input: Accelerators from different directions (x,y,z)

$$J(\theta_0, \theta_1 ... \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$h_\theta(x^{(i)})$: Predicted value
$y^{(i)}$: Actual value

We choose the above function to minimize. This cost function is also known as the Mean Squared Error (MSE) function.

3.2.2 Gredient Desent

Gradient descent is an iterative optimization algorithm to find the minimum of the cost function. The idea is that we start with some values for θ and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.

After doing the derivation of θ repeatedly, the function is shown below.

$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} ((h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)})$$

α: Learning rate

The choice of correct learning rate is very important as it ensures that Gradient Descent converges in a reasonable time.If the learning rate is very large, Gradient descent can overshoot the minimum. It may fail to converge or even diverge, which is shown in Figure 7.If the learning rate is very small, Gradient descent will take small steps to reach local minima and will take a longer time to reach minima, shown in Figure 8.

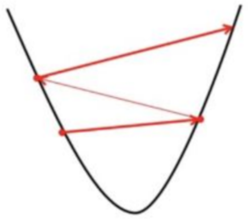


Figure 7: Gradient descent with very large α

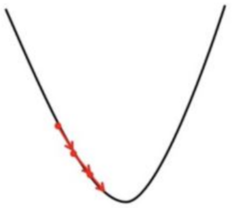


Figure 8: Gradient descent with very small α

3.3 SVM

Support vector machine (SVM) is a supervised learning method. Given labeled data, it would establish a learning model and divide all the data into two parts[5]. Assume that the data we collected is with amount n. After labeling them with 0 and 1, the machine knows how to separate them with a hyperplane according to their labels. Define margin to be the shortest distance from the hyperplane to the labeled points as shown in Figure 9. Support vector machine would look for the most suitable hyperplane with largest margin[6].

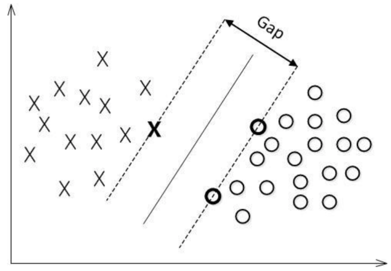


Figure 9: Support Vector Machine

3.3.1 Data Processing

Before testing and training data for separation, data should be processed in the certain form:

$$[x_1, y_1, z_1, x_2, y_2, z_2, ..., x_n, y_n, z_n]$$

Where x_n, y_n, z_n represent the acceleration velocity for three dimensions.

The next step is to label data manually. Attach label 1 to people with authorization and 0 to others. Four file.txt files are created separately for label 1, label 0, data of label 1 and data of label 0 correspondingly. Programmed with Python, we import svm from sklearn and set the code to distinguish the four files and then use them.

3.3.2 Training and Testing

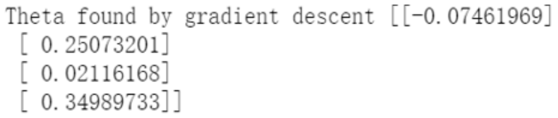
In order to make support vector machine satisfy the requirements of our project, we then set parameters in the Python code, in which gamma and C are the most critical. For instance, gamma = auto which means its value is 1/n features[7]. The C is a penalty parameter of error term. It trades off correct classification of training and testing data against the maximization of margin. With larger value of C, machine would force more punishment for misclassification. Nevertheless, if we set a smaller value of C, setting lower C is at the cost of the training accuracy. After finishing all the parameters setting. After being fitted, the model can be applied to predict new values.

For convenience, we divide each file for separately training and testing with certain percentage: [40, 50, 60, 70, 80, 90]. Then we mix all the label files together and data files as well for both training and testing. With various percentages of training and testing data sets, we obtain a graph of accuracy.

4 Results

4.1 Multiple Linear Regression

In our project, we let the learning rate be 0.01 and the times of iteration be 2000. After optimizing the algorithm by using gradient descent method, we can estimate the best values of θ, which can be seen in the Figure 10 below.



```
Theta found by gradient descent [[-0.07461969]
 [ 0.25073201]
 [ 0.02116168]
 [ 0.34989733]]
```

Figure 10: The values of θ

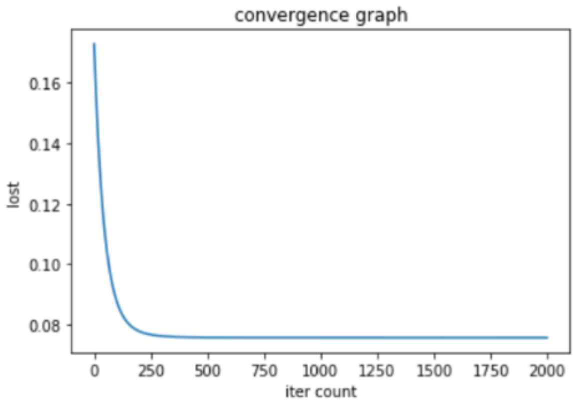


Figure 11: The convergence graph about lost and the count of iteration

In Figure 11 which presents the relationship between lost and the count of iteration, we can realize that at the beginning, as the count of iteration increases, the loss value decreases. After about 250 times, the value of lost converge to 0.075. The accuracy of this Multiple Linear Regression is about 75 percent.

4.2 SVM

As is shown in figure 12, with various percentages of training data, the accuracy of authentication is different. The percentage of training data and authentication accuracy is not simply linear relation. To acquire better recognition performance, we can collect a great deal of acceleration velocity and let them train and test with diverse percentages, and then analyze the most appropriate proportion between testing data and training data. We can utilize this proportion to optimal our model. Obviously, from the figure below, when training data takes up 80 per cent of the whole data set, the accuracy is the highest which is 88.2 percent. Compared with the Linear Regression Algorithm, SVM makes a better performance.

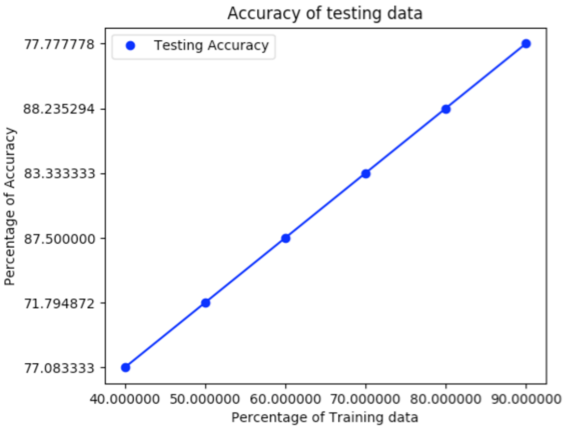


Figure 12: Accuracy with SVM

5 Conclusion

In this paper, we have elaborated our project with related work, the details of our approaches, and the expected results. By illustrating the approaches we took and the corresponding results, we have shown the feasibility and reliability of our new authentication method employed in the future. However, there still are much work for us to do to completely finish our authentication method so that it can work in real life. Also, we will discuss the future work and the corresponding security aspect in following paragraphs.

The Figure 13 below shows how the authentication process work in real life. We designed to have users unlock the phone to trigger the authentication process. After the authentication process starts, the phone will start to collect useful data for further authentication. The collected data will then be transmitted to the host system for comparison. If the model built from the data just collected highly matches the model stored in the system, we can authorize the entrance of the user. Otherwise, the user will be blocked from entering and the system will send out a warning message to the security.

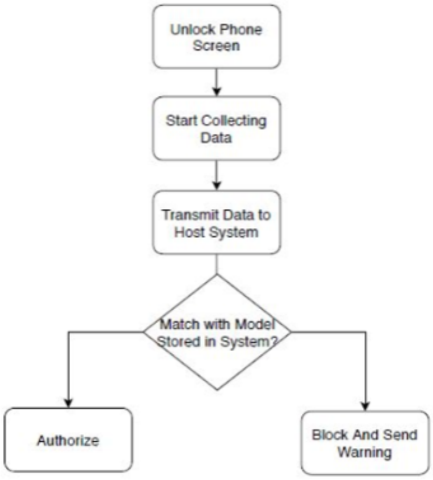


Figure 13: Authentication Process in Real Life

5.1 Future Work

5.1.1 Accuracy improvement

Due to the limitation of time, only 3000 groups of data form three people are collected as dataset which is not large enough. There exists the possibility that the walking style of the three people are so different to separate them easily. Thus, the accuracy is higher than it actually should be. When other person whose acceleration velocity is rather similar with the one with authentication is taking Android phone and walking for authentication, it would be quite likely that the system shows he or she has the permission.

As a result, our accuracy would be improved a lot if we expand the amount of data for training. What is more, we can also try some other algorithms that is more advanced than Linear Regression and SVM for the improvement of performance.

5.1.2 Data transmission

In our project, the approach to transfer acceleration velocity is to manually operate. It is too awkward and time-wasted. What we plan to do is designing an application which can realize transferring data from phone to computer system automatically once finishing one record. Then we just need to directly process the data received by computer and use them to train and test.

5.1.3 Security

As for our project, there exists a latent security problem. If the attacker holds the phone of the person with authorization and maliciously replaces the information about acceleration velocity with his own ones, he could successfully obtain the permission and enter the room with secret. The method we came up with to deal with this particular security problem is performing encryption. We can either encrypt the file through setting password for people who want to open it or encrypt the single data inside the file.

Appendix

Our project code has been uploaded to Github. If you are interested in checking out our code, you can follow this link:
https://github.com/nickpei/EECS221-Security-Project

References

[1] Charles P. Pfleeger Shari Lawrence Pfleeger and Jonathan Margulies. *Security in computing.* 2015.

[2] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 2012.

[3] William J. Francis. *A Quick Tutorial on Coding Androids Accelerometer.* 2011. https://www.techrepublic.com/blog/software-engineer/a-quick-tutorial-on-coding-androids-accelerometer/.

[4] Adi Bronshtein. *Simple and Multiple Linear Regression in Python.* 2017. https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9.

[5] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *Proceedings - International Conference on Pattern Recognition*, 2004.

[6] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998.

[7] Jayavardhan Ravi. *EEG-Data-predection.* 2016. https://github.com/jayavardhanravi/EEG-Data-predection.