Spring 5-20-2019

Smartphone Gesture-Based Authentication

Preethi Sundaravaradhan
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the [Artificial Intelligence and Robotics Commons](#), [Graphics and Human Computer Interfaces Commons](#), and the [Information Security Commons](#)

Recommended Citation

Sundaravaradhan, Preethi, "Smartphone Gesture-Based Authentication" (2019). *Master's Projects*. 701.
DOI: <https://doi.org/10.31979/etd.ykqn-vum5>
https://scholarworks.sjsu.edu/etd_projects/701

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Smartphone Gesture-Based Authentication

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Preethi Sundaravaradhan

May 2019

© 2019

Preethi Sundaravaradhan

ALL RIGHTS RESERVED

The Designated Project Committee Approves the Project Titled

Smartphone Gesture-Based Authentication

by

Preethi Sundaravaradhan

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2019

Dr. Mark Stamp	Department of Computer Science
----------------	--------------------------------

Dr. Katerina Potika	Department of Computer Science
---------------------	--------------------------------

Professor Fabio Di Troia	Department of Computer Science
--------------------------	--------------------------------

ABSTRACT

Smartphone Gesture-Based Authentication

by Preethi Sundaravaradhan

In this research, we consider the problem of authentication on a smartphone based on gestures, that is, movements of the phone. Accelerometer data from a number of subjects was collected and we analyze this data using a variety of machine learning techniques, including support vector machines (SVM) and convolutional neural networks (CNN). We analyze both the fraud rate (or false accept rate) and insult rate (or false reject rate) in each case.

ACKNOWLEDGMENTS

I would like to express gratitude to my advisor Dr. Mark Stamp for his patience, motivation and guidance on this project. My thanks to the committee members Professor Fabio Di Troia and Dr. Katerina Potika for their encouragement. I would also like to thank volunteers and my colleagues for helping me with data collection for this project. I will forever be thankful to the Almighty, my parents, and friends for the immense support and guidance received throughout my journey.

TABLE OF CONTENTS

CHAPTER

1	Introduction	1
2	Background	3
2.1	Authentication systems	3
2.2	Comparison of authentication techniques	4
2.2.1	Error evaluation in biometric authentication	5
2.3	Machine learning techniques	6
2.3.1	Convolutional neural networks	7
2.3.2	Principal component analysis	7
2.3.3	Support vector machines	8
2.4	Attacks on gesture-based techniques	9
2.5	Accelerometer data	9
2.6	Data collection	10
2.7	Data preprocessing	13
2.7.1	Features	13
3	Experiments and Results	15
3.1	Support vector machines	15
3.2	Convolutional neural networks	24
3.3	Discussion	26
4	Conclusion and Future Work	29
	LIST OF REFERENCES	30

LIST OF TABLES

1	Summary of experiments and results	28
---	--	----

LIST OF FIGURES

1	Confusion matrix	6
2	Derivation of ERR from ROC	6
3	An overview of PCA	8
4	An overview of SVM [1]	9
5	3D acceleration space of authentic user	11
6	Screenshot of Firebase database with sample data	11
7	An iOS application for data collection	12
8	Mean of x , y and z axes for type 1 data	16
9	ROC for SVM on mean of type 1 data	17
10	Median of type 1 data along x -axis	17
11	Median of type 1 data along y -axis	17
12	Median of type 1 data along z -axis	18
13	ROC for SVM on median on type 1 data	18
14	Mean of type 2 data across x , y and z axes	18
15	ROC for SVM on mean values of type 2 data	19
16	Median of type 2 data along x axis	19
17	Median of type 2 data along y axis	20
18	Median of type 2 data along z axis	20
19	ROC for SVM on median values of type 2 data	20
20	Magnitude of type 1 and type 2 data	21
21	Principal components of type 1 data	22

22	Principal components of type 2 data	22
23	Confusion matrix of SVM on type 1 data	23
24	Confusion matrix of SVM on type 2 data	23
25	Sample images of different user signatures	24
26	Sample images of forged signatures	24
27	CNN architecture	25
28	ROC curve for type 1 data	25
29	ROC curve for type 2 data	26
30	Confusion matrix for CNN on type 2 data	27
31	Comparison of final results	28

CHAPTER 1

Introduction

Authentication is an integral part of security to any digital system to provide privacy and protection. Several methods of authentication have evolved over years. The most ubiquitous form of user authentication is passwords. Another form of authentication is using the biometric features of users. These features must be distinguishable and unique to the user. Biometric authentication can be divided into two categories based on whether the users are identified by their physical features or their behavioral patterns [2]. According to Liu et al., a new type of behavioural biometric authentication system inspired from handwritten cheques evolved. This system involves signing gestures in the air similar to drawing signatures with pen.

One such system named OpenSesame claims to achieve a high-level of security and robustness with a mean false positive rate of 15% and false negative rate of 8% [3]. Opensesame evaluates the hand gesture actions of users without any restrictions. Thus, it does not take into account the scenario where an intruder observes the hand movements, and imitates the movements to gain access to system. This scenario is similar to signature forging in cheques.

Several machine learning techniques have been experimented in the past to recognize a user signature from accelerometer sensor data. Most commonly used techniques include Hidden Markov Model (HMM) [4] [5], Support Vector Machine, Recurrent Neural Networks, and Dynamic Time Wrapping (DTW) [6] [7]

This research explores and compares the effectiveness of gesture-based authentication system under unrestricted and forging scenarios. Effectiveness is measured in terms of accuracy of user identification and intruder detection. Also, this research explores convolutional neural networks and principal component analysis for analysis accelerometer data. The idea to use convolutional neural networks is inspired from a

previous work that deploys this technique to authenticate a user using pressure sensor data [8]. Most research in gesture-based authentication systems focuses on detecting the shape and stroke of the pattern [9]. This research also explores other statistical features such as mean, median, and magnitude without explicitly identifying the shape of the signature.

The report is structured in the following manner. Chapter 2 gives a background on available authentication systems, machine learning techniques used for gesture recognition, details on working of these techniques, and data collection setup. Chapter 3 gives a detailed explanation on the experiments and results of this research for classifying user signatures. Chapter 4 discusses the most effective techniques and features found, and finally discusses the future scope of this research.

CHAPTER 2

Background

2.1 Authentication systems

In this section, various methods used to authenticate a user to a computer system are discussed. A machine can authenticate a user by means of something they know, such as password, or by something they have such as an RFID (radio-frequency identification), or by biometric features unique to the user such as fingerprint. These are popularly summarized in the security domain as ‘something you know, something you have or something you are’ [10].

Widely used authentication systems are the knowledge-based mechanisms like PIN and password [11]. This system involves the user to record numeric or alphanumeric characters of significant length in a suitable keypad. This system is easy to implement since authentication is a simple string comparison of the existing password to the provided value. This ease of implementation and cost of computation is the reason for the popularity and wide usage of password based authentication system.

Another approach to authentication is using users biometric features. This can be categorized into physical biometrics and behavioural biometrics.

Physical biometrics allow verifying the user with regard to what he/she is. Some of the popular physical biometric features include iris, fingerprint, hand geometry, and face. Authentication involves scanning the biometric feature of the user and matching this image to a user.

Behavioral biometrics, also known as physiological biometrics, allow verifying the user with regard to how he/she behaves [9]. According to Liu et al., these features depend on upon the knowledge and/or habits of the user. An example of knowledge based physiological attribute is handwritten signatures, that are captured as images and image-based pattern recognition techniques are used to authenticate users [4].

An example of habit based physiological attribute is, keystroke dynamics. Keystroke dynamics is based on the assumption that, the time taken to type two different keys is unique for each user [12]. This system involves time-series analysis of timing data between pre-recorded timings and current timings of typing each key in a keyboard. In their paper, Liu et al., discusses another example of habit based attribute namely, gait authentication systems [13]. These systems leverage the speed and motion pattern of users.

An interesting and novel user authentication system based on gestures captured using accelerometer sensor was proposed by Huang [9]. This system comes under the category of both knowledge and habit based behavioural biometrics. Here, the 3D accelerations are captured when the user waves a device containing the sensor in air/space. The sensor data is a sequence of x , y , and z acceleration values. This sequence is then associated to the user using pattern matching techniques.

2.2 Comparison of authentication techniques

The knowledge-based systems using passwords and PINs suffer from issues such as usability [10]. Humans can find remembering passwords difficult. Also, the basic assumption of password-based system is that, passwords cannot be duplicated or stolen [14]. But in reality, this assumption does not hold well. Other issues with passwords include misuse, that is, users can pass off copies to others intentionally. Above all, this system works only when, keyboards or other devices are available to capture the password characters. In small devices like IoT wearables and smart watches, this type of authentication is not user-friendly [14].

Physical biometrics like fingerprint, will need special sensors to capture the input. For face recognition, high precision cameras may be needed and are not usable in poorly lit environments. In addition to hardware cost, computational costs of these

techniques are also high [3].

Physiological biometrics such as keystroke dynamics and gait based authentication techniques are heavily criticized by several literature for being error-prone and demonstrated to be not useful for real world authentication [4] [9].

Gesture-based authentication requires only an accelerometer and most modern smart devices have built-in accelerometer. Also, this sensor is cheap, light-weight and does not boost the cost of the whole device. Finally, this is computationally 10-times cheaper than face-recognition and other similar types of physical biometrics [9]. This is suitable for small devices such as smart watches.

2.2.1 Error evaluation in biometric authentication

Two types of errors can occur in authentication systems. The rate at which an intruder may be erroneously recognized as an authentic user is referred to as *fraud rate*. An authentic user may be incorrectly rejected as an intruder. The rate at which this type of misauthentication happens is the *insult rate* [10]. In machine learning terminology, the fraud rate is referred to as false acceptance rate (FAR). Similarly, insult rate is referred to false reject rate (FRR).

The confusion matrix for the user and intruder classification is in Figure 1, from which the FAR and FRR formulas can be derived as

$$\text{FAR} = \text{FPR} = \text{FP}/(\text{FP} + \text{TP})$$

and

$$\text{FRR} = \text{FNR} = 1 - \text{TPR} = \text{FN}/(\text{TP} + \text{FN})$$

There is a trade off between the fraud rate and the insult rate wherein, decreasing the fraud rate would lead to increase in insult rate and vice versa. The *equal error rate* (EER) is the rate at which the fraud rate and the insult rates are balanced. This is a measure of effectiveness of biometric systems. A low equal error rate implies that

		True Conditons	
Predicted Labels		User	Intruder
	User	TP	FP
	Intruder	FN	TN

Figure 1: Confusion matrix

the system has higher accuracy [15].

The ERR can be derived from ROC easily by tracing the point where the values of FPR and (1-TPR) are equal. This method is depicted in Figure 2 [16].

Figure 2: Derivation of ERR from ROC

2.3 Machine learning techniques

Most biometric systems need to approximate the biometric feature observations using various pattern matching or machine learning techniques. Gesture-based authentication is also one such system which needs the aid of machine learning algorithms to closely approximate the signatures.

2.3.1 Convolutional neural networks

Convolutional neural networks (CNNs) is a type of deep neural network that makes an explicit assumption that the input will be an image and structures the network weights accordingly. Because of this property, CNNs have been applied successfully in image classification and numerous other pattern recognition tasks with exceptionally high accuracy [17]. The hidden layers in CNNs are a series of convolution layers, pooling layers and fully connected layers. The functions of these layers are explained as follows [18].

- Convolution layer collects the information from the local region of the image.
- Pooling layer down samples the image so as to reduce the number of parameters and computations in the network.
- Fully connected layer computes the class scores. The number of neurons in this layer must be equal to number of categories of images

Though the gestures are real-valued tri-axial data points, they can be transformed to images by concatenating the acceleration values from raw accelerometer along the 3 axes [19]. Then, the image is constructed by normalizing the values and mapping them to series of color codes for each pixel in the image [8]. To save these pixels as an image, a python library for imaging is used [20].

2.3.2 Principal component analysis

Principal component analysis (PCA) is a mathematical technique used to transform a complex data to a lower dimensional space called principal components. These principal components hold the compressed and important information from the higher dimensional space. The first principal component holds the data with maximum variability, and next principal component accounts for the remaining variability and so on. Since in-air signatures have features like direction of movement and acceleration,

the principal component analysis is an interesting technique to explore in this research. Figure 3 shows the principal components in a scatter plot of multivariate Gaussian distribution. Here the arrows represent the direction of two principal components or eigenvectors, with the length of the arrow equal to its respective eigenvalues [21].

PCA is sometimes used for visualization of the data by projecting the magnitude of the first two principal components i.e., the eigenvalues in the x -axis and y -axis respectively [22].

Figure 3: An overview of PCA

2.3.3 Support vector machines

Support vector machine is a supervised learning technique which separates two classes by constructing an optimal hyperplane. This technique operates by projecting the data in higher dimensional space [23]. Figure 4 shows a hyperplane separating two classes in higher dimension, the dots represent one class and squares represent another class. Here, the hyperplane, which is shown as a line, separates these classes. SVM is originally designed for binary classification. It can be extended to multiclass classification by using the "OneVsRest" strategy [24]. The one-vs-rest is a heuristic

where one classifier is trained per class and each of these one-class classifiers are fitted against all the other classes.

Figure 4: An overview of SVM [1]

2.4 Attacks on gesture-based techniques

Attacks on any authentication system start with stealing the unique token that authenticates the user. In the case of password based authentication, attacks will be to steal and use the password. In the case of biometric authentication, attacks involve copying the biometric features [25]. However, forging any biometric token such as fingerprint or facial feature has proved to be hard. The gesture-based authentication scheme using accelerometer pattern, is harder to tamper than the facial recognition. This difficulty to tamper is due to the search space of the pattern [4]. According to Guse., in addition to knowing the pattern itself, the imposter needs to know the angle, speed, and relative area in which the pattern is drawn. His research assumes that nobody can copy the signature [25] [13].

2.5 Accelerometer data

The accelerometer is a sensor in smartphone, that can capture the gesture i.e., waving action of the user. Accelerometer measures the movements of the phone, in terms of acceleration relative to freefall along the x , y and z axes. The unit of

measurement is g , where

$$g = 9.8m/sec^2$$

on earth. For example, if the smartphone is placed flat on the ground, the accelerometer will read $0g$ along the x and y axis and $1g$ along the z axis [26].

When the device is moved, the acceleration along the three axes are measured as a sequence of tri-axial data points represented by

$$(x_t, y_t, z_t),$$

where x , y , and z represents the acceleration along these axes and t denotes the time. Typically, accelerometers allow the sampling time to be user defined. In our research, the sampling rate is fixed at 50ms. This means, every 50ms the acceleration of x , y and z axis will be recorded. Thus, if a gesture lasts for 2 seconds, the accelerometer records 40 data points represented as a collection

$$\{(x_0, y_0, z_0), \dots, (x_i, y_i, z_i), \dots, (x_{39}, y_{39}, z_{39})\}$$

This signature is formally defined as [3]

$$S = \{(x_{t_0}, y_{t_0}, z_{t_0}), (x_{t_1}, y_{t_1}, z_{t_1}), \dots, (x_{t_n}, y_{t_n}, z_{t_n})\}$$

On connecting the signature datapoints in 3D space, the shape of the signature pattern can be reconstructed. The Figure 5 depicts a signature shaped like an 'S' drawn in air.

2.6 Data collection

Data collection in smartphone is required for the two platforms: Android and iOS. This is done to maximize the ease of the data collection process independent of smartphone platform. For Android platform, a custom application was created. This application automatically uploads data to a cloud-based database namely Google Firebase [27].

Figure 5: 3D acceleration space of authentic user

Sample data and database interface of collected data is shown in Figure 6. The data is stored in JSON (JavaScript Object Notation) format. Here, the timestamp at which the signature collection is started, forms the root of the JSON tree and every tri-axial data point recorded in the 50ms interval forms the children of this tree.

Figure 6: Screenshot of Firebase database with sample data

For the iOS platform, a publicly available application named Accelerometer is used [28]. A screenshot of the application is shown in Figure 7. The signature is plotted as a time series curve. Here acceleration along each axis is represented along the y -axis and time along the x -axis of the graph. Accelerometer values in the 3D x -axis represented by green curve, y -axis by the red curve and z -axis by the blue curve.

Figure 7: An iOS application for data collection

During the data collection process, user performs the following steps.

- Click the 'Start measuring' button on the app (The start button toggles to stop and records the acceleration).
- Move the smartphone in air to draw the signature or gesture
- Click on 'Stop measuring' button.

The above step records and loads the data to the database in cloud in JSON format.

Data is collected from 102 users and divided into two sets. A smaller dataset with 10 user data and the larger dataset containing all the user data.

2.7 Data preprocessing

In this research, small dataset containing 10 users data is used. Here, the user 0 is an authentic user. Other 9 users generated 20 samples of unrestricted gestures (Type 1) and 20 samples of gestures imitating the user 0 (Type 2).

- Type 1: An unrestricted pattern chosen by the user which is a unique signature for the user.
- Type 2: A common pattern that the users observe and forge.

2.7.1 Features

The following statistical and structural features of the tri-axial data from accelerometer are considered in our experiments [29] [30].

- Mean: The mean values of the data for each of the three axes is calculated and this is done for every signature of a user. This gives us three mean values for each signature over three dimensions.
- Median: The median values of the data for each of the three axes is calculated similar to mean. Consequently, this gives us three median values for each signature over three dimensions

- Magnitude: The magnitude of a signature is the average of the root mean square of the tri-axial data [31]. The magnitude m is calculated as

$$m = \left(\sum_{k=1}^s \sqrt{x_k^2 + y_k^2 + z_k^2} \right) / s$$

where s is the signature length.

- Velocity: The velocity of the smartphone is calculated as the difference of consecutive data points along each of the three axes.

$$\text{speed}[x][i] = \text{data}[x][i] - \text{data}[x][i - 1]$$

$$\text{speed}[y][i] = \text{data}[y][i] - \text{data}[y][i - 1]$$

$$\text{speed}[z][i] = \text{data}[z][i] - \text{data}[z][i - 1]$$

CHAPTER 3

Experiments and Results

The process of authentication can be viewed as two classification challenges. First, from the unrestricted signature dataset, the machine learning techniques should be able to distinguish the users. Thus, it is a multiclass classification problem where number of classes is equal to number of users. Secondly, in the forging scenario, one user must be classified as real user and the rest as intruders. This is a binary classification problem. We consider the accuracy for the two categories of data mentioned above and compute the equal error rate for the machine learning technique with highest accuracy.

This chapter uses the following techniques to analyze the usability of the features discussed in Chapter 2, and discusses the results for the above two scenarios:

- Support vector machines
- Principal component analysis
- Convolutional neural networks

3.1 Support vector machines

The visualizations of mean values of 9 signatures along the x , y and z axes are shown in Figure 8, by plotting mean values on x -axis and sample id in y -axis. Each user is assigned a unique color. The results show that, mean values along x -axis are different for the two users (dark-blue and purple) and shows considerable overlap for the remaining users. Mean values along z -axis is similar to x -axis showing significant difference for two users (light-blue and purple). Furthermore, mean values of y -axis show the largest difference among at-least four users (dark-blue, light-blue, yellow, and purple).

Combining these tri-axial mean values as features, a SVM is trained to fit this data. SVM is implemented using sklearn python library [32]. On using SVM multiclass

classification [33], on all the user signatures we get a mean training accuracy of 64%. The ROC curve for this set is shown in Figure 9. Accuracy values for individual users from ROC curve, reflects the observations from the tri-axial dataset. Thus, we can infer that signatures of the user dark-blue and user purple highly different from other users. Thus, they have an average accuracy of 100%. Moreover, user lightgreen and user dark-green had significant overlap in mean values along all the three axes. Thereby, making their ROC values similar. This pattern can also be related to user orange and user blue. Thus, observing the diagonal from top left to bottom right corners of the ROC curve, the equal error rate from this system for user aqua is highest and the users dark-blue and purple have lowest EER. Hence, SVM is capable of constructing separable hyperplanes among users fairly well.

Figure 8: Mean of x , y and z axes for type 1 data

Similarly, median values of type 1 data across the x , y and z are visualized in Figures 10, 11 and 12. The same users (user dark-blue and user purple) who were distinguished using the mean were also classified correctly using median. The corresponding ROC curves are shown in Figures 13. Also, observing the diagonal of the ROC curve, user orange has highest EER, and users dark-blue and purple have lowest EER. The mean accuracy taken for all users is 61%. Thus, median feature gives 3% lesser accuracy compared to mean.

We repeat the same experiments for forged signatures (type 2) by the 9 users.

Figure 9: ROC for SVM on mean of type 1 data

Figure 10: Median of type 1 data along x -axis

Figure 11: Median of type 1 data along y -axis

Visualizations of mean values across x , y and z axes are shown in Figure 14. Here we can visually distinguish only user orange and user green, along the y -axis. Users are not distinguishable in x and z domains, as there is considerable overlap. On using

Figure 12: Median of type 1 data along z -axis

Figure 13: ROC for SVM on median on type 1 data

SVM, we observe the same. The ROC for the mean values on type 2 data is shown in Figure 15. The EER for dark-blue is observed to be highest. Here, the accuracy for the user orange and user green are 100%. However, the mean accuracy of identifying the forged signatures is 54%. This is lower compared to the accuracy on type 1 data.

Figure 14: Mean of type 2 data across x , y and z axes

Figure 15: ROC for SVM on mean values of type 2 data

Similarly visualizations for median values across x , y and z axes on type 2 data are shown in Figures 16, 17 and 18. Again SVM classifies the 2 users (user orange and user green) with AUC of 100%, which is same as the results from mean. SVM gives an average accuracy of 56%. This is lower than the accuracy of type 1 data. The ROC for the median values of type 2 data is shown in Figure 19

Figure 16: Median of type 2 data along x axis

It is observed that accuracy of classifying unrestricted signatures is better than the accuracy on signatures which were forged. Also, it is difficult to conclude whether mean or median is a better feature. Overall the features mean and median produce less than 70% accuracy. This is lower than any of the previous research explored in the previous chapter.

Figure 17: Median of type 2 data along y axis

Figure 18: Median of type 2 data along z axis

Figure 19: ROC for SVM on median values of type 2 data

Next, we consider the magnitude as a feature described in Chapter 2. Figure 20 shows a comparison of magnitudes of both types of signatures. The magnitude of most users are similar. Difference between the magnitudes of users 1, 2, 6, 8 and 9 is less

than 10. This is very less to be able to differentiate the users. Also, the magnitude difference between type 1 and type 2 data is not co-related. That is, user 1 has the same magnitude for both type 1 and type 2 data. User 3 has higher magnitude for type 2 data. And finally, user 6 has slightly lower magnitude for type 2 data. Though this implies users are not able to imitate the magnitude of a real user, this observation also means the real user cannot be distinguished from intruders. Observing this trend, we conclude that the magnitude of tri-axial data is not a distinguishing feature of users and this feature is not explored further.

Figure 20: Magnitude of type 1 and type 2 data

Lastly, to use the velocity as a feature, the principal components of the accelerometer data is extracted using PCA. This method gives us the eigenvalues and eigenvectors for each signature. The top two maximum eigenvalues of the corresponding principal components extracted from the signatures are plotted along the x -axis and y -axis respectively. The ratio of magnitude of principal component vectors of each users is clustered together as observed in the Figures 21 and 22. SVM is used to classify the users using this eigenvalues. This approach gives us an average accuracy of 87.8% on type 1 data and 65% on forged type 2 data. Clearly, the SVM continues to perform bad in identifying forged signatures.

Figure 21: Principal components of type 1 data

Figure 22: Principal components of type 2 data

The confusion matrix shows the various user signatures predicted by SVM versus the true user in Figure 23. Here user 1, user 2 and user 3 has a consistent signature and is predicted correctly for 5 test dataset. The other users are misclassified atleast once. The values along diagonals have higher votes, this implies users are classified correctly most of the times. The prediction is based on maximum voting strategy. If a test data gets equal votes for all classes, then it is not classified into any category.

The confusion matrix in Figure 24 shows the forged signatures predicted by SVM versus the true user. Here user 0 is the real user and all other users are imitating user 0. Attempts of user 4, and user 6 are classified as real user. This implies these

Figure 23: Confusion matrix of SVM on type 1 data

users are able to break the system.

Figure 24: Confusion matrix of SVM on type 2 data

Support vector machines are then used to classify users based on velocity without reducing the dimensionality by PCA. This is done by concatenating the x , y and z axis data into one feature vector. This give 99.7% accuracy for type 2 data and 100% accuracy for type 1 data on the small dataset of 10 users. On repeating the same experiment with large dataset of 102 users, we get 100% on both type 1 and type 2 data.

3.2 Convolutional neural networks

Velocity is the best performing feature so far as shown in Section 3.1. Considering this result, we do not explore mean, median, magnitude any further and explore only velocity. We convert the tri-axial data to image as discussed in 2. Figure 25 shows sample images from 7 user signatures of Type 1 data. Here all the images are easily distinguishable, implying the users are distinguishable. Figure 26 shows sample images from 7 user signatures of forged signatures. Here though the images are distinguishable, a white streak in the central part of the image is common to 6 of 7 users. We can assume that these users are able to forge some features of the signature. CNN is used to classify the users on both the types of data and analyze if we can support this assumption.

Figure 25: Sample images of different user signatures

Figure 26: Sample images of forged signatures

The architecture of CNN used for image classification is shown in Figure 27. It has 3 convolution layer, each followed by a max-pool layer and finally a fully connected layer [34]. Keras framework on python is used for implementation of this CNN [35].

Figure 27: CNN architecture

For uniquely identifying the users, which is a multiclass classification problem, the CNN consists of 9 nodes in the fully-connected dense layer. The number of nodes in the dense layer must be equal to the number of classes or users. For this experiment, the dataset is split into training set and validation set. Training data set consists of 17 images from each user and the validation data set consists of 3 images from each user. On training the CNN with this training dataset, an average accuracy of 97% is obtained. On testing the model using the validation set, we get the results as shown in the ROC curve in Figure 28. Here, 5 users are classified with an accuracy of 100% and the least accuracy for a user is 88%. This shows CNN performs better than SVM.

Figure 28: ROC curve for type 1 data

For classifying type 2 data, the CNN architecture is reused with the last fully-

connected layer replaced to have only 2 nodes, since this is a binary classification problem. The training and validation set are split with the training set consisting of 5 images from 8 imposters and 15 images from real user, and validation data consisting of 46 imposter user images and 6 real user images. The average training accuracy of the model is 86%. The ROC curve for this experiments is shown in Figure 29. Here the mean accuracy for the user is 83%.

Figure 29: ROC curve for type 2 data

The confusion matrix for the CNN classifier that compares the user and all intruders is shown in Figure 30, where label 0 is the real user and label 1 is intruder. From this figure, it is clear that a user false reject rate is 0% but false acceptance rate is 7%.

CNN performs much better on distinguishing the users when the signatures are distinct. Also, the average accuracy of the system is lower on type 2 data, implying this system is slightly poor in detecting forged signatures.

3.3 Discussion

Since the dataset, and machine learning techniques are different from the previous researches [3], we analyzed our models under both restricted and unrestricted signature patterns to be able to compare the performance of our authentication system under

Figure 30: Confusion matrix for CNN on type 2 data

the two scenarios. For the available data, SVM on accelerometer data gives best results than the other techniques explored in this section. Average of mean, median and magnitude are features that do not hold all the properties of the signatures. PCA identifies the dominant plane in which the user has moved the most, thus it achieves good results in identifying distinct signatures, but it is poor when the intruder has observed the dominant plane and imitates the user. The accelerometer data in the image domain holds the sequence of changes in velocity over smaller intervals and preserves the pattern. This could explain the reason for good results with CNN. And when using the same accelerometer data without any dimensionality reduction, SVM achieves the best results.

Now that we have established velocity as a feature gives best results. The equal error rate of CNN is found by observing the ROC curve in Figure 29. From the discussions in Chapter 2, the EER is the point where the black dotted line intersects the ROC curve. Hence EER for this system is 0.25. The results of all the experiments conducted is summarized in Table 1

The results of these techniques are plotted in bar graph with accuracy along the x -axis. From the graph shown in Figure 31, SVM and CNN classifier on velocity has

Table 1: Summary of experiments and results

Technique	Feature	Distinct signatures	Forged signatures
SVM	Mean	0.64	0.54
SVM	Median	0.61	0.56
SVM	Velocity	1.00	0.99
PCA+SVM	Velocity	0.87	0.65
CNN	Velocity	0.97	0.88

the highest accuracy.

Figure 31: Comparison of final results

CHAPTER 4

Conclusion and Future Work

The aim of this project was to explore the effectiveness of accelerometer based in-air signatures as a gesture based authentication mechanism. A smartphone application was built and accelerometer tri-axial data was collected from users. Both the built application and the user data can be reused for further experiments.

Of the several features explored, velocity of the user has been found to be the most effective in differentiating users. Mean, median and magnitude of the sensor data seems to be inefficient at differentiating users. Convolutional neural network (CNN) was designed and trained to classify the user data. This technique is found to be a good classifier with a high accuracy of 97% at identifying users and 86% accuracy at detecting imitations. The EER for this system was 0.25. Another technique explored was SVM's multiclass classifier. The accuracy of SVM with velocity is better than CNN for identifying user and as well as detecting imitations. The equal error rate is 0 in this case. In all cases, the accuracy of identifying different patterns is better than detecting imitations. From these experiments we can conclude that, identifying unique user signature is feasible but the techniques and features explored in this research are not enough to effectively identify imitations of signatures.

This research can be further expanded by collecting longer signatures, more samples, and testing against the machine learning techniques experimented in Chapter 3. In addition, the technique of extracting principal components and using the eigenvalues as features in tri-axial data looks promising. Future research could explore this further by using these features on other classification techniques like k -nearest neighbors, k -means clustering or k -medoids clustering techniques.

LIST OF REFERENCES

- [1] Savan Patel, “SVM Theory,” [Online; accessed April 26, 2019]. [Online]. Available: <https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>
- [2] S. M. Ganesh, P. Vijayakumar, and L. J. Deborah, “A Secure Gesture Based Authentication Scheme to Unlock the Smartphones,” in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, Feb 2017, pp. 153–158.
- [3] L. Yang, Y. Guo, X. Ding, J. Han, Y. Liu, C. Wang, and C. Hu, “Unlocking Smart Phone through Handwaving Biometrics,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1044–1055, May 2015.
- [4] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra, “Analysis of pattern recognition techniques for in-air signature biometrics,” *Pattern Recognition*, vol. 44, no. 10, pp. 2468 -- 2478, 2011, Semi-Supervised Learning for Visual Content Analysis and Understanding. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320311001683>
- [5] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.
- [6] S. Mitra and T. Acharya, “Gesture Recognition: A Survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.
- [7] J. Yang, Y. Li, and M. Xie, “MotionAuth: Motion-based authentication for wrist worn smart devices,” in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, March 2015, pp. 550–555.
- [8] M. S. Singh, V. Pondenkandath, B. Zhou, P. Lukowicz, and M. Liwicki, “Transforming sensor data to the image domain for deep learning - An application to footstep detection,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 2665–2672.
- [9] C. Huang, Z. Yang, H. Chen, and Q. Zhang, “Signing in the Air w/o Constraints: Robust Gesture-Based Authentication for Wrist Wearables,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

- [10] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [11] S. K. Behera, D. P. Dogra, and P. P. Roy, “Localization of signatures in continuous air writing,” *IEEE Region 10 Symposium (TENSYP)*, 2017, pp. 1–5, 2017.
- [12] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “uWave: Accelerometer-based personalized gesture recognition and its applications,” *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657 -- 675, 2009, perCom 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1574119209000674>
- [13] C. Liu, G. D. Clark, and J. Lindqvist, “Guessing Attacks on User-Generated Gesture Passwords,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 1, pp. 3:1–3:24, Mar. 2017. [Online]. Available: <http://doi.acm.org/10.1145/3053331>
- [14] C. Liu, G. D. Clark, and J. Lindqvist, “Where Usability and Security Go Hand-in-Hand: Robust Gesture-Based Authentication for Mobile Systems,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’17. New York, NY, USA: ACM, 2017, pp. 374–386. [Online]. Available: <http://doi.acm.org/10.1145/3025453.3025879>
- [15] M. Stamp, *Information Security: Principles and Practice*, 2nd ed. Wiley Publishing, 2011.
- [16] R. Tronci, G. Giacinto, and F. Roli, “Dynamic Score Combination: A Supervised and Unsupervised Score Combination Method,” in *Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition*, ser. MLDM ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 163–177. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03070-3_13
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: <http://doi.acm.org/10.1145/3065386>
- [18] Andrej Karpathy, “Convolutional Neural Networks,” [Online; accessed April 26, 2019]. [Online]. Available: <http://cs231n.github.io/convolutional-networks>
- [19] W. Jiang and Z. Yin, “Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks,” in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM ’15. New York, NY, USA: ACM, 2015, pp. 1307–1310. [Online]. Available: <http://doi.acm.org/10.1145/2733373.2806333>

- [20] Fredrik Lundh, “Pillow,” [Online; accessed April 26, 2019]. [Online]. Available: <https://pillow.readthedocs.io/>
- [21] Wikipedia, “Principal Component Analysis,” [Online; accessed April 26, 2019]. [Online]. Available: https://en.wikipedia.org/wiki/Principal_component_analysis
- [22] Michael Galarnyk, “PCA using Python,” [Online; accessed April 26, 2019]. [Online]. Available: <https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>
- [23] M. Stamp, *Introduction to Machine Learning with Applications in Information Security*, 1st ed. New York, NY, USA: Chapman and Hall/CRC, September 2017, pp. 98–115.
- [24] scikit-learn, “OneVsRest Classifier,” [Online; accessed April 21, 2019]. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html>
- [25] D. Guse, “Gesture-based User Authentication on Mobile Devices using Accelerometer and Gyroscope,” Master’s thesis, Berlin Institute of Technology, 2011.
- [26] Fitbit, Inc, “Accelerometer Sensor Guide,” [Online; accessed April 26, 2019]. [Online]. Available: <https://dev.fitbit.com/build/guides/sensors/accelerometer>
- [27] Google, “Firebase,” [Online; accessed April 25, 2019]. [Online]. Available: <https://firebase.google.com>
- [28] DreamArc, “Accelerometer,” [Online; accessed April 25, 2019]. [Online]. Available: <https://itunes.apple.com/us/app/accelerometer/id499629589>
- [29] T. Hai Nguyen, P. Pham, C. Q. Ngo, and T. Tam Nguyen, “A SVM Algorithm for Investigation of Tri-Accelerometer Based Falling Data,” *American Journal of Signal Processing*, vol. 6, pp. 56–65, 01 2016.
- [30] M. Zhang and A. A. Sawchuk, “A Feature Selection-based Framework for Human Activity Recognition Using Wearable Multimodal Sensors,” in *Proceedings of the 6th International Conference on Body Area Networks*, ser. BodyNets ’11. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 92–98. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2318776.2318798>
- [31] T. Bishal Singha, R. K. Nath, and A. V. Narsimhadhan, “Person Recognition using Smartphones’ Accelerometer Data,” *arXiv e-prints*, p. arXiv:1711.04689, Nov 2017.

- [32] P. Saimadhu, “SVM classifier implementation in Python with scikit-learn,” 2017, [Online; accessed April 21, 2019]. [Online]. Available: <http://dataaspirant.com/2017/01/25/svm-classifier-implementaion-python-scikit-learn/>
- [33] J. Suriya Prakash, K. Annamalai Vignesh, C. Ashok, and R. Adithyan, “Multi class Support Vector Machines classifier for machine vision application,” in *2012 International Conference on Machine Vision and Image Processing (MVIP)*, Dec 2012, pp. 197–199.
- [34] Francois Chollet, “Keras Blog,” [Online; accessed April 25, 2019]. [Online]. Available: <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>
- [35] Keras, “Keras Sequential Library,” [Online; accessed April 21, 2019]. [Online]. Available: <https://keras.io/models/sequential/>