DragID: A Gesture Based Authentication System

Junyoung Min

Department of Electrical Engineering

Graduate School of UNIST

DragID: A Gesture Based Authentication System

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Junyoung Min

06. 03. 2014 of submission
Approved by.

Advisor

Seyoung Chun

DragID: A Gesture Based Authentication System

Junyoung Min

This certifies that the thesis of Junyoung Min is approved.

06. 03. 2014 of submission

Signature

Advisor: Seyoung Chun

Signature

Committee Member: Kyunghan Lee

Signature

Committee Member: Giljin Jang, Kyungpook National University

Abstract

With the use of mobile computing devices with touch screens is becoming widespread. Sensitive personal information is often stored in the mobile devices. Smart device users use applications with sensitive personal data such as in online banking. To protect personal information, code based screen unlock methods are used so far. However, these methods are vulnerable to shoulder surfing or smudge attacks. To build a secure unlocking methods we propose DragID, a flexible gesture and biometric based user authentication. Based on the human modeling, DragID authenticates users by using 6 input sources of touch screens. From the input sources, we build 25 fine grained features such as origin of hand, finger radius, velocity, gravity, perpendicular and so on. As modeling the human hand, in our method, features such as radius or origin is difficult to imitate. These features are useful for authentication. In order to authenticate, we use a popular machine learning method, support vector machine. This method prevents attackers reproducing the exact same drag patterns. In the experiments, we implemented DragID on Samsung Galaxy Note2, collected 147379 drag samples from 17 volunteers, and conducted real-world experiments. Our method outperforms Luca's method and achieves 89.49% and 0.36% of true positive and false positive. In addition, we achieve 92.33% of TPR in case we implement sequence technique.

Contents

1 Introduction -- 9

2 DragID : A Gesture Based Authentication System -- 11

3 Features -- 13

3.1 Raw Data Process --- 13

3.2 Features 1~3: Radius and Origin --- 14

3.2.1 Definition -- 14

3.2.2 Extraction Method --- 15

3.2.3 Feature Analysis -- 15

3.3 Feature 4: Length of Drag --- 17

3.3.1 Definition -- 17

3.3.2 Extraction Method --- 17

3.4 Feature 5~8: Start and End Point of Drag --- 18

3.4.1 Definition -- 18

3.4.2 Feature Analysis -- 18

3.5 Features 9~11: Gravity --- 19

3.6 Feature 12~17: Touch Area -- 19

3.7 Feature 18~23: Velocity -- 20

3.7.1 Definition -- 20

3.7.2 Extraction Method --- 20

3.7.3 Feature Analysis -- 20

3.8 Feature 24~25: Perpendicular -- 21

3.8.1 Definition -- 21

3.8.2 Extraction Method -- 22

4 Classification Models of Machine Learning -- 23

4.1 Support Vector machine -- 23

4.2 K-Nearest Neighbor -- 27

4.3 Vector Quantization -- 28

5 Related Work --- 30

6 Experimental Result --- 32

6.1 Experimental Setting -- 32

6.2 Accuracy Evaluation --- 33

6.3 Real-world Evaluation --- 36

6.4 Comparison with Classification -- 38

6.5 Comparison with Existing Schemes -- 38

7 Limitations and Discussion -- 39

7.1 Limit of Posture -- 39

7.2 Size of Phones -- 39

7.3 Dynamic Update Learning -- 39

8 Conclusion -- 40

List of Figures

Figure 1 Posture of grabbing phone --- 9
Figure 2 Authentication system of DragID --- 11
Figure 3 Radius histogram-- 15
Figure 4 Origin coordinate histogram --- 16
Figure 5 Graph of drag data --- 17
Figure 6 Coordinate of start and end point -- 18
Figure 7 Gravity sensor value of the phone -- 19
Figure 8 Velocity magnitudes of drag --- 21
Figure 9 Perpendicular -- 22
Figure 10 Hyperplanes separating the two classes. --- 23
Figure 11 An optimal hyperplane -- 24
Figure 12 Example of k-NN classification --- 27
Figure 13 Accuracy evaluation table-- 33
Figure 14 Result of first experiment --- 34
Figure 15 Result of second experiment -- 35
Figure 16 ROC curve in accordance to SVM threshold --- 35
Figure 17 Histogram of SVM score for one model--- 36
Figure 18 CDF of the number of unlocked case-- 37

List of Tables

Table 1 List of API DATA-- 13
Table 2 Unstable Google API-- 14
Table 3 Comparison of classification -- 38
Table 4 Comparison of DragID with GEAT and [6]-- 38

1. Introduction

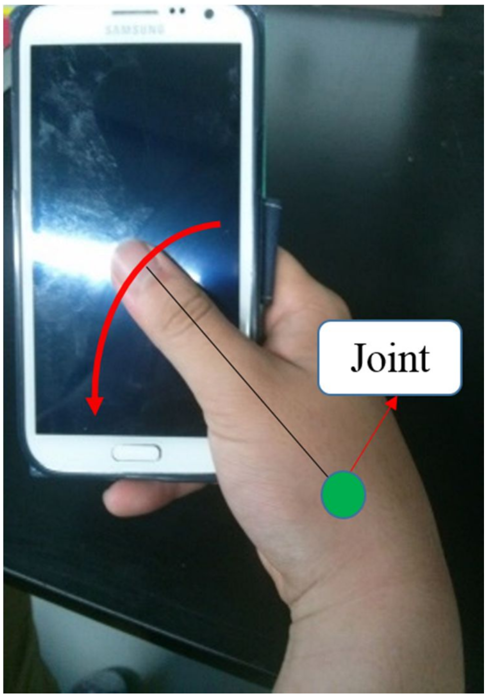


Figure 1: Posture of grabbing phone. Drag of arc form is generated because joint is fixed if rotate the finger.

After smartphone was developed, mobile market became increase rapidly. 6.2 billion mobile market was established in 2012, a portion of smartphone occupied 1.1 billion [1]. It is expected to go up 5.6 billion in 2019. Smartphones provide convenience for our daily lives. Touch screens became the user input technologies for mobile computing devices because those have user convenience and intuitive. Most applications are running on mobile devices such as desktop applications because mobile computing power is growing up. To deal with various functions on mobile devices, personal photos, e-mails, credit card numbers, passwords, corporate data are handled.

In case users lose smartphone including personal information, owner information are spilled. Security firm Symantec conducted a real-life experiment in five major cities in North America by leaving 50 smart phones in streets without any protection [2]. The results showed that 89% of finders accessed for personal related apps and information, 83% accessed for corporate related apps and information, 43% accessed online banking, and 50% contacted the owner and provide contact information.

Protecting private information is important on smartphone with possibility of loss or theft. The widely adopted methods are password/PIN/geometric pattern screens for unlocking devices. For example, iPhones use a 4-digit PIN and Android phones use a geometric pattern on a grid of points. That methods are based on configuration of users. These password/PIN/geometric pattern based unlocking schemes have three major weaknesses. First, they are vulnerable to sneak a look at unlocking such as shoulder surfing. It is very critical to shoulder surfing because they are subjective. If imposter directly see the shoulder surfing once, security should be unlocked immediately. Smartphones are used in outdoors where are exposure to shoulder surfing. Second, they are vulnerable to smudge attacks. Third, they are inconvenient for users to input frequently. In case of long length of the security system, performance is good. However, owner must remember that always. It is difficult to enter for owner it. When the button is located far distance on touch screen, it is necessary to use both hands or move hand largely.

We propose an authentication method that uses the features modeled joint, length of finger, shape of hand, and posture. There is no inconvenience to input in our method. It will only create a drag of circle shape by owner’s hand is stretched out and rotated with common posture. The owner don’t need to change the posture and use both hands for input. Our method calculate and authenticate user information from very simple gesture. It extracts hand shape and habit of act from the gesture. The features are Origin, radius, gravity sensor values, velocity, touch point, and perpendicular. Origin and radius are estimated values for length and the joints of the finger. The features are very effective to prevent copying and imitating.

We process features on machine learning algorithms with k-Nearest Neighbor (k-NN), Vector Quantization (VQ), and Support Vector Machine (SVM). SVM shows the best performance in those algorithm.

2. DragID: A Gesture Based Biometric Authentication System

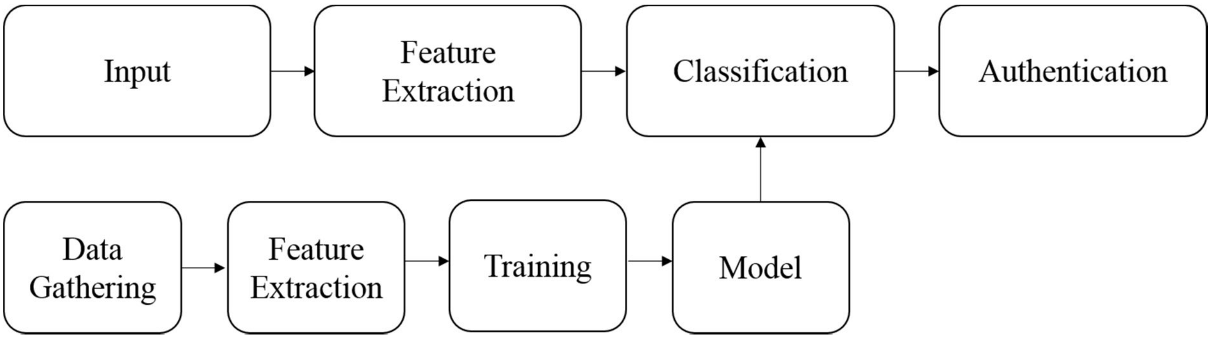


Figure 2. Authentication system of DragID

In this thesis, we propose DragID, a gesture and biometric based authentication methods to unlock smartphone by using touch screen. A gesture is a drag which is drawn by human hand shape, finger length, habit and grab poses. Figure 1 shows example of general posture to grab smartphone. Although unlocking methods such as password, PIN and geometric pattern lock have been adopted generally, it is vulnerable to shoulder surfing attack because it is subjective. DragID is based on hand shape and how to input. For authentication, DragID needs to make a model about owners. In order to collect training samples, the users performs to make the drags on touch screen. They use mobile phones in the default posture and hold it, and finger is rotated. Drag is generated on the shape of circle. After sufficient data is collected, then DragID makes the model. This model is stored in the mobile device for the classification. We use classification for authenticating whether owner or imposter which method is named SVM after extracted features from this sample drag. SVM classifies whether owner's or imposter's drags when performed it.

In comparison of existing secure unlocking scheme and DragID, It is impossible to imitate to owner's drag through shoulder surfing and smudge attack. In this thesis, we propose DragID that have 3 advantages. First, Biometric authentication such as fingerprints have a possibility of copy but our method don't. For example, iPhone use a fingerprint authentication. It take high-definition hardware for scanning fingerprint in order to authenticate owner exactly, but high-definition scanner show fingerprint authentication was incapacitated [3]. Second, it don't need to change posture during using phone. Our methods ask to conduct a drag while people usually have used in general posture that are showed in figure 1. Therefore, people don't need to use both hands in order to authenticate, just use in general grab posture. Third, our method only require fundamental hardware of smartphone such as gravity sensor and touch screen, not additional hardware.

3. Features

To use the classification, it is necessary to extract from the smartphone input for distinguishing users. We used android mobile phone Galaxy Note 2 on Samsung. Inputs are generated from touch screen and gravity sensor. For obtaining touch data from touch screen and gravity sensor, android API (Application Programming Interface) must be used. Android API is provided by Google. Android API catches touch data 17~18ms each in general and the value of gravity sensor with appointed interval.

We gather 5 values from smart phone sensors such as touch screen and gravity sensor and extract 25 features. For examples radius, origin, gravity, velocity, touch area, and perpendicular. The drag data obtained from API provided by Google can be expressed as follows.

Symbol	Description
$t = \{t_1, t_2, \cdots, t_n\}$	recorded time
$x = \{x_1, x_2, \cdots, x_n\}$	x coordinate of touch screen
$y = \{y_1, y_2, \cdots, y_n\}$	y coordinate of touch screen
$tma = \{tma_1, tma_2, \cdots, tma_n\},$	major area of contact on touch screen
$tmi = \{tmi_1, tmi_2, \cdots, tmi_n\}$	minor area of contact on touch screen
$wm = \{wm_1, wm_2, \cdots, wm_n\}$	major area of contact on finger
gx, gy, gz	gravity values

Table 1. Lists of API Data

3.1. Raw Data Process

In order to obtain touch data from touch screen, android API must be used. Android API catches touch data 17~18ms each in general. However, catch event is sometimes skipped because of scheduling of operation system in case smartphone is busy.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
General case	18	18	17	17	18	17	17	17	17	17	18	17	17	17	17	17	3
Unstable case	22	17	17	18	17	17	17	17	17	18	34	17	17	18	34	17	17

Table 2: This table shows unstable google API. Index values are time values of $t_{i+1} - t_i$. In case of between 11 to 12 and 15 to 16, catching event is skipped.

In that case, feature calculation take errors such as perpendicular. So we need to make a compensation for missing data.

```
Algorithm: Compensation Algorithm
$\mathrm{s} = \text{sampling interval}$
if $t_{i+1} - t_i > 2 * \mathrm{s}$
$t_{step} = \frac{t_{i+1} - t_i}{\text{floor}\left(\frac{t_{i+1} - t_i}{s}\right)}$
for($t = t_i; t \leq t_{i+1} - t_{\text{step}}; t = t_i + t_{\text{step}}$)
$x = \frac{t - t_i}{t_{i+1} - t_i} * (x_{i+1} - x_i) + x_i$
$y = \frac{t - t_i}{t_{i+1} - t_i} * (y_{i+1} - y_i) + y_i$
```

3.2. Features 1-3: Radius and Origin

3.2.1. Definition

In case people grab the phone and stretch thumb, and rotate as circle shape, drag is generated as circle shape. Since the rotation axis to the joint of the finger and fixed hand, unstable quarter circle is generated because of a change of touched area. To analyze this unstable circle, there are several ways to define it. The best example is circular regression. Circular regression methods have been used in many diverse applications. But it is improper to analyze real data point because it's fitting method based on error. Radius means distance between touched area and joint. Origin means joint.

3.2.2. Extraction Method

We extract origin and radius through the format find the original similar circle generated substantially. In order to use only real data point and make a deduction of circle, we use equation of circle.

$$r^2 = \left(x_i - x_{origin}\right)^2 + \left(y_i - y_{origin}\right)^2 \ (i = 1,2,\cdots,n)$$

r is radius and length of human finger, origin of x and y is joint of finger. n is the number of catching data from API. In order to solve this equation, need 3 points. If one drag have 40 points, the number of output will be C_3^{40}. The sets of C_3^{40} show Gaussian distribution histogram. We select r, x_{origin} and y_{origin} by taking the average of range of values that occur most frequently in those sets.

3.2.3. Feature Analysis

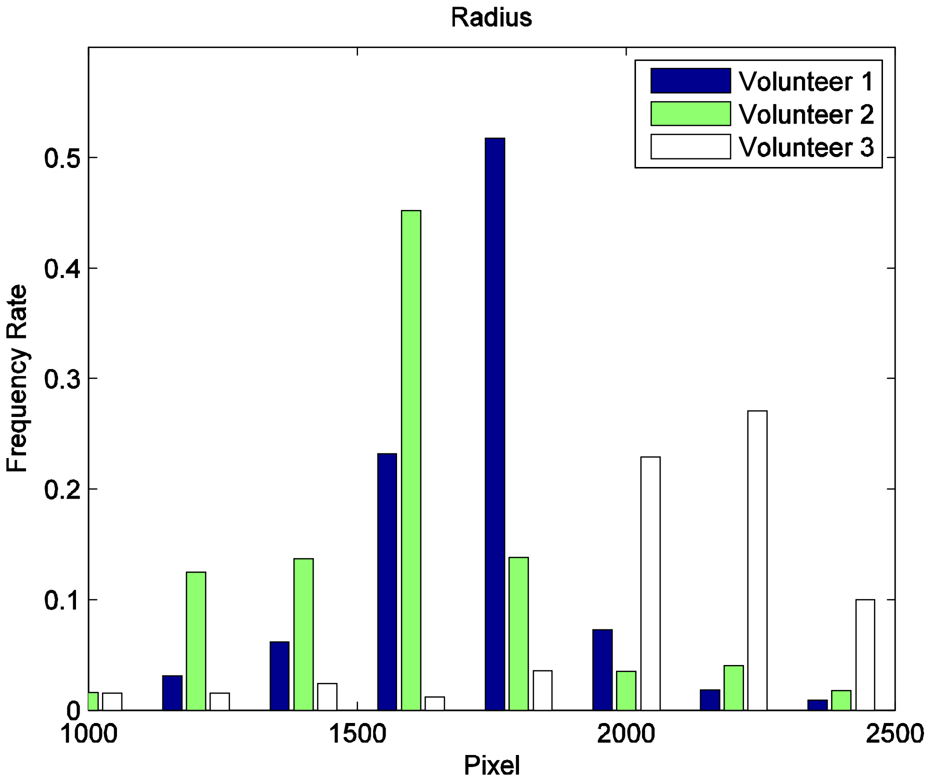
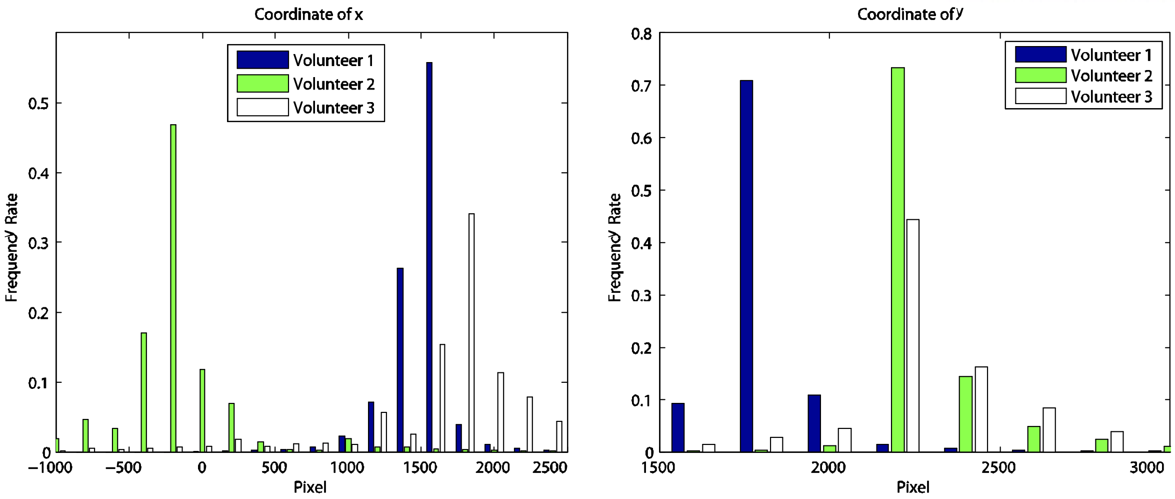


Figure 3: Radius histogram of C_3^n output which are calculated from one drag.



(a) (b)

Figure 4: origin coordinate histogram of C_3^n output which are calculated from one drag. (a) is origin of x. (b) is origin of y

Figure 3 and 4 shows the histogram of the values calculated from the single drag. They show volunteers have each other distribution because finger length and posture is different. (a) of figure 4 is located to a value less than 0. This can be inferred to be used with the left hand. To extract feature value, we take mode for the histograms.

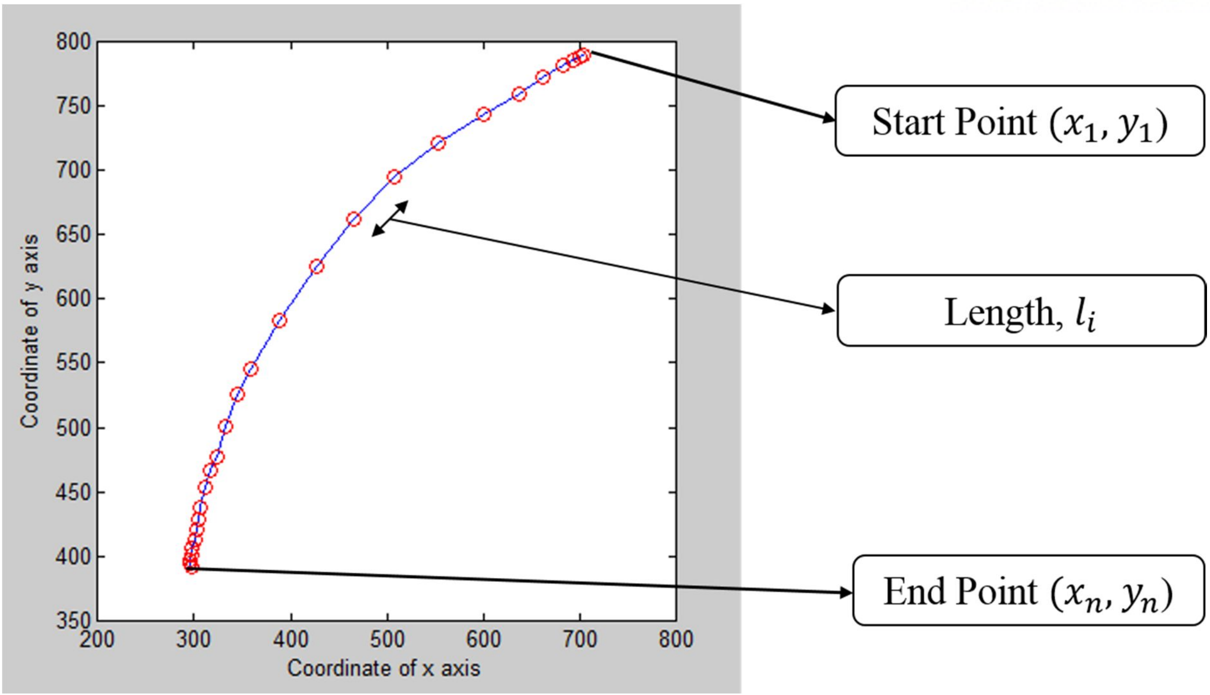


Figure 5: Graph of the drag data. Each red point is coordinate of $(x_i, y_i), (i = 1, \cdots, n)$

3.3. Feature 4: Length of Drag

3.3.1. Definition

The length of the drag is the length of the arc that is generated when people created once the drag. This is affected by shape of a hand and finger. As Posture of taking a mobile phone is similar and the shape of the hand is unchanged, this is one of the element that represents the people.

3.3.2. Extraction Method

This feature is Euclidean distance between sequential touch points. Length of blue line is feature.

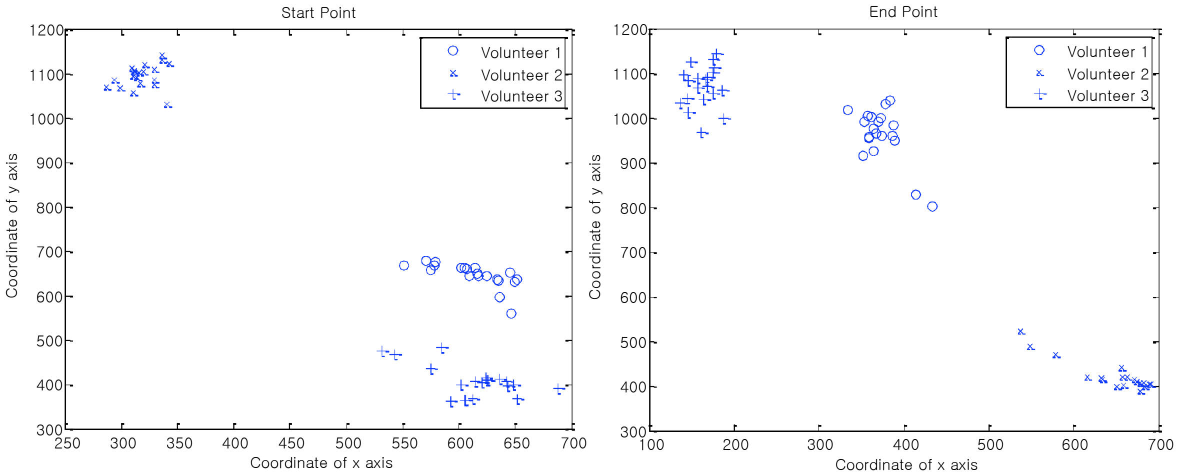
$$\sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

3.4. Features 5~8: Start and End Point of Drag

3.4.1. Definition

In case people stretched out finger on the mobile phone, start and end point of the drag is about the same because shape of finger is not changed. In the figure 6, the start and end points of three volunteers are different because of grabbed postures when people taking the phone. In addition, the attitude to take mobile phone when the people use it is almost the same. Therefore, the start and end positions are similar to each people. Therefore, there are four feature. : x_1, x_n, y_1, y_n

3.4.2. Feature Analysis



(a) (b)

Figure 6: Coordinate of start and end point. The symbols such as o, x, and + mean each volunteers.

3.5. Feature 9~11: Gravity

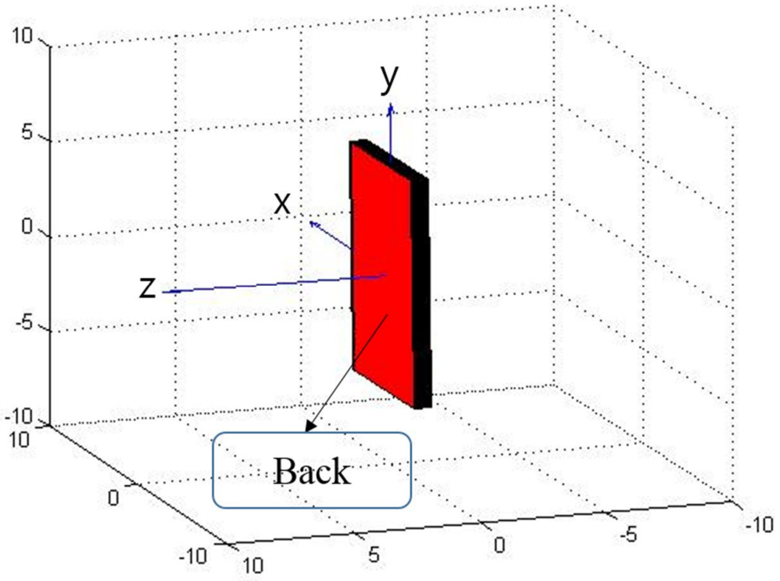


Figure 7: Gravity sensor value of the phone.

Gravity sensor is one of the fundamental hardware that smartphones have. Gravity values is represented three-dimensional shape representing x, y, and z. The sensor is capable of confirming that the telephone is located in any form. The value of the gravity sensor is able to check the mobile phone is how located. When people taking a mobile phone, posture is similar. Accordingly, the location and status of the mobile phone is an important factor to separate the people.

3.6. Feature 12~17: Touch Area (Mean, Variance)

Touch screen hardware provides information about the contacted area of finger surface. These is determined by the length of the longest and shortest of the contacted area. tma and tmi are the longest and shortest length of contacted area of touch screen in contact with it. wm mean the length of the longest finger contacted touch screen. Thus, wm is the same as the length of the finger. In addition, these values depend on the strength of the contact. These values present habit of act and shape of hand.

3.7. Feature 18~23: Velocity (Mean, Variance)

3.7.1. Definition

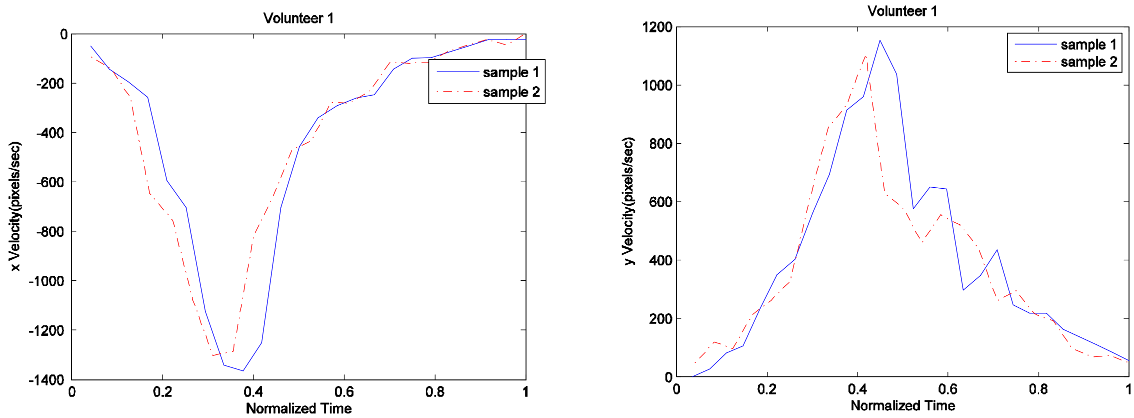
It is the speed that is created when you have created a drag until grabbed the mobile phone. (a) of figure 8 is a graph of the dragging speed of the x -coordinate. (b) of figure 8 is a graph of the dragging speed of the y coordinate. Form of increase or decrease rate is substantially similar when creating a drag on people each. These figures show that the shape of the movement of the drag is different for each people.

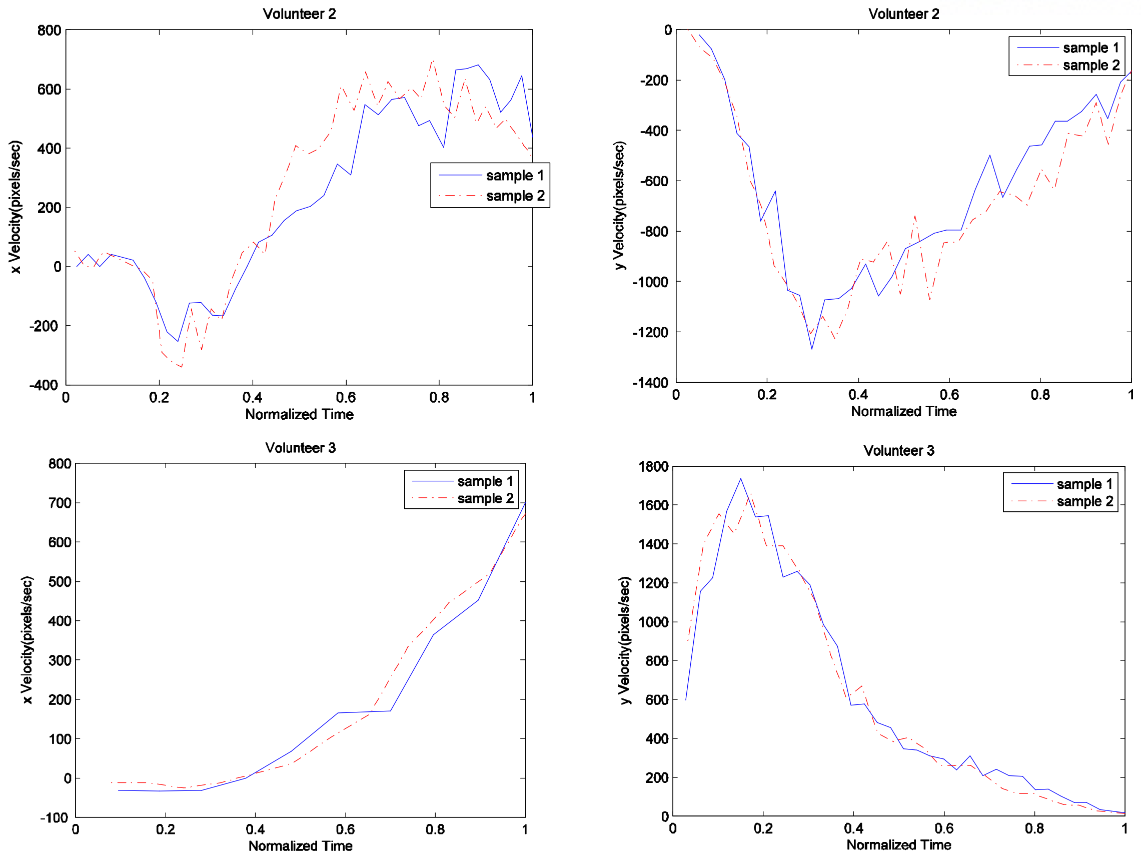
3.7.2. Extraction Method

$$vx_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i}, i = 1,2,\cdots,n-1$$

$$vy_i = \frac{y_{i+1} - y_i}{t_{i+1} - t_i}, i = 1,2,\cdots,n-1$$

3.7.3. Feature Analysis





(a) (b)

Figure 8: x and y velocity magnitudes of drag

Figure 7 shows change of velocity is different for each volunteer. They move finger with different speed until taking the gesture. These indicates that habits of act is reflected.

3.8. Feature 24~25: Perpendicular Length (Mean, Variance)

3.8.1. Definition

We assumed a set of perpendicular length is affected by shape and joint of finger. In the quarter of circle, we draw line between start and end coordinate point. The rest of points draw an orthogonal line for the line. In Figure, Straight line depicting “--“ shows connected line between start and end point. “-.” line means perpendicular. px and py are points intersected at the straight line. The sets of this perpendicular lines show how much bend when human make drag. It is possible to infer that it is the imbalance how much variance in the sets.

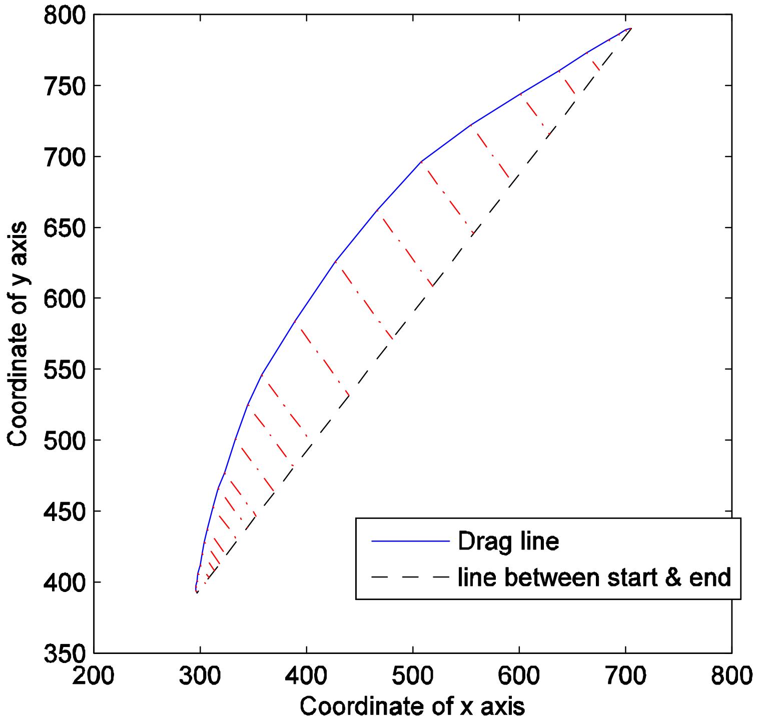


Figure 9: Perpendicular. “-.” means set of perpendicular line.

3.8.2. Extraction Method

$$px = \frac{s \cdot x_n - y_n + \frac{x_i}{s} + y_i}{s + \frac{1}{s}}, i = 2,3,\cdots,n-1,\ \text{where}\ s = \frac{y_n - y_1}{x_n - x_1}$$

$$py = \frac{-s \cdot x_1 + y_1 + s \cdot x_i + s^2 \cdot x_i}{s^2 + 1}, i = 2,3,\cdots,n-1,\text{where}\ s = \frac{y_n - y_1}{x_n - x_1}$$

$$plen = \sqrt{(px_i \ - x_i \ )^2 + (py_i \ - y_i \ )^2}$$

4. Classification Models of Machine Learning

Since the human can be learned, it is possible expand own knowledge continuously. As learning is the most important element that humans have, human make a decision through learning. Attempts to implement on computer by itself is a machine learning. In one field of artificial intelligence, machine learning is an algorithm for the computer to be able to learn. Machine learning algorithm, when the ability to enter training data into a computer, perform to predict what is to be kind of class based on the algorithm of any particular building a discrimination criterion. Core of machine learning is methods to approach by representation and generalization. Representation is an evaluation of the data. Generalization is in the treatment of data for an unknown.

4.1. Support Vector Machine

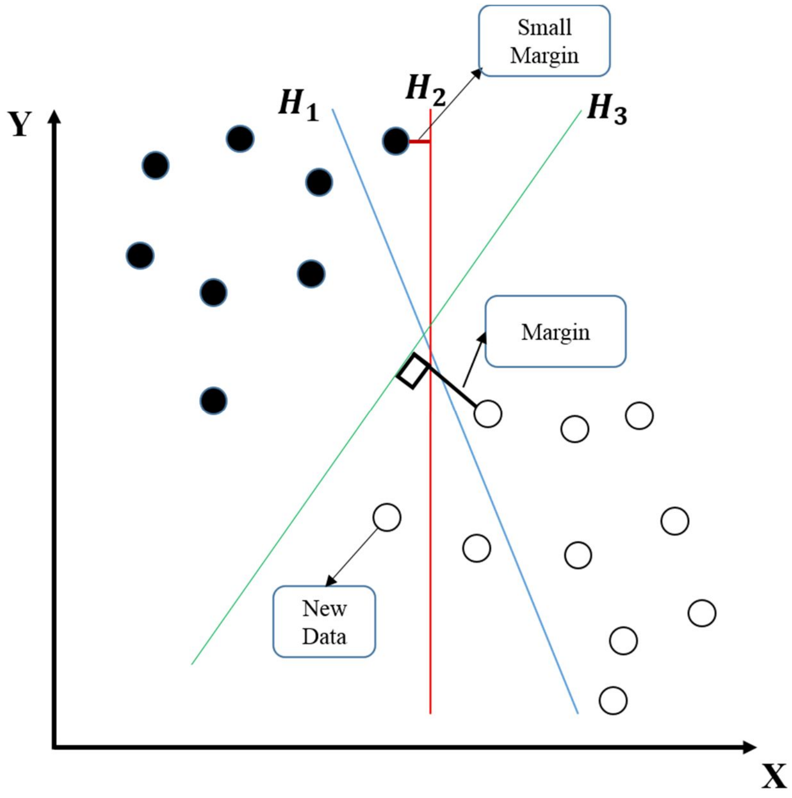


Figure 10: Hyperplanes separating the two classes. There are many hyperplanes that separate the two classes of data points.

Support vector machine is a supervised learning model that is used for regression and classification in machine learning. The SVM is proposed as Vector Networks in 1995 by Vanpnik [4] and currently widely used worldwide. SVM is that in the method of classifying the data with two categories, to find the optimal hyperplane that separates the two groups as far as possible for the given data. Because there are many hyperplane to classify the data belonging to two categories in general, SVM can search what hyperplane is optimal. For example, there are black and white data in Figure. The methods of separating the two categories are various such as H1, H2 H3 and so on. H1 is no quality to separate the two categories properly. H2 and H3 separate exactly. However, in cases of H2, efficiency is not so good as compared with the distribution of the data. For example, H2 have worse separation than H3, and H3 have the precise separation. In order to calculate hyperplane have the margin to separate farthest from two classes of given data, it should make the hyperplane that maximizes the margin. Defined as the margin the distance between the data points on these lines, the hyperplane of SVM bisects the center of the margin, which is the maximum size, and the point composed of the lines call support vector.

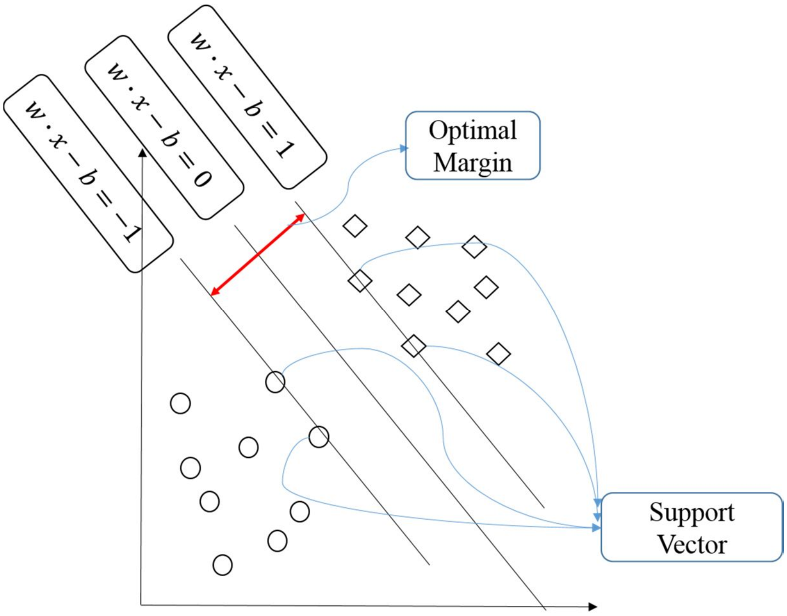


Figure 11: An optimal hyperplane separating the two classes.

The outermost data presented by square is support vector in each class on Figure2. A straight line passing through the support vector can be expressed as follows.

$$\mathrm{w} \cdot \mathrm{x} + \mathrm{b} = 1 \qquad (1)$$
$$\mathrm{w} \cdot \mathrm{x} + \mathrm{b} = -1 \qquad (2)$$

In this case, for each data belongs to class, it is whether above (1) or under (2), it must satisfy the following conditions in order to be located.

$$\mathrm{w} \cdot \mathrm{x}_{class1} + \mathrm{b} \geq 1 \qquad (3)$$

$$\mathrm{w} \cdot \mathrm{x}_{class2} + \mathrm{b} \leq 1 \qquad (4)$$

The distance between the two lines is ${}^{2}/_{|w|}$, $|w|$ should be minimized in order to maximize the margin. However, there is almost no case where training data does not exist between (1) and (2). In order to calculate hyperplane with the largest margin even if the other class is present in the function between these, slack variable is added to the equation. That is called soft margin method. Therefore, the following formula will be used.

$$\text{slack variable } \xi_i \geq 0, \mathrm{i} = 1, \cdots, \mathrm{l}$$

$$\mathrm{w} \cdot \mathrm{x}_i + \mathrm{b} \geq 1 - \xi_i, i = 1, \cdots, \mathrm{l}$$

Vanpink proposed to solve the following primal optimization problem.

$$\min \frac{1}{2} w^T \cdot w + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to } \; y_i(\mathrm{w} \cdot \phi(\mathrm{x}_i) + \mathrm{b}) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \mathrm{i} = 1, \cdots, \mathrm{l}$$

One of the important techniques of the SVM is able to solve the nonlinear classification problems by using the kernel. It make possible by implementing a slack variable and solve to some extent in the case of linearly separable problem, but it is limited because it uses as a classification boundaries of the linear hyperplane. By mapping the dimension of the input data on dimension of $\phi(\mathrm{x})$ that implement to increase the dimensions, nonlinear classification problems can be solved by using a simple linear classifier.

However, it considers a side effect of an increase in computational amount generated by increasing the dimensions. If the assumption dimension of the kernel is very high, computational amount of the high-dimensional vectors is not practical. SVM performs the kernel trick method in order to solve the problem of these operations. The kernel trick use the inner product of two vectors (i.e., $\phi(\mathrm{x}) \cdot \phi(\mathrm{x}_i)$) defined as k(x, y) instead of individual value of $\phi(\mathrm{x})$. k(x, y) is called the kernel function. The use of a kernel function can replace a two-dimensional calculation from the computation of high-dimension, it is possible to solve the problem of computational cost.

Type of kernel to be used in the SVM are as follows.

$$\text{polynomial kernel} : \mathrm{k(x,y)} = (x \cdot y + c)^d$$

$$\text{sigmoid kernel} : \mathrm{k(x,y)} = \tanh(k(x,y) + \theta)$$

$$\text{radian basis function} : \mathrm{k(x,y)} = \exp(-\|x - y\|^2/(2\partial^2))$$

Radian basis function is the most commonly used and called the Gaussian kernel. We implemented Gaussian kernel. Each kernel has $\partial, \theta, \mathrm{d}$ parameters. These have to be adjusted suitably according to the data.

4.2. K-Nearest Neighbors algorithm

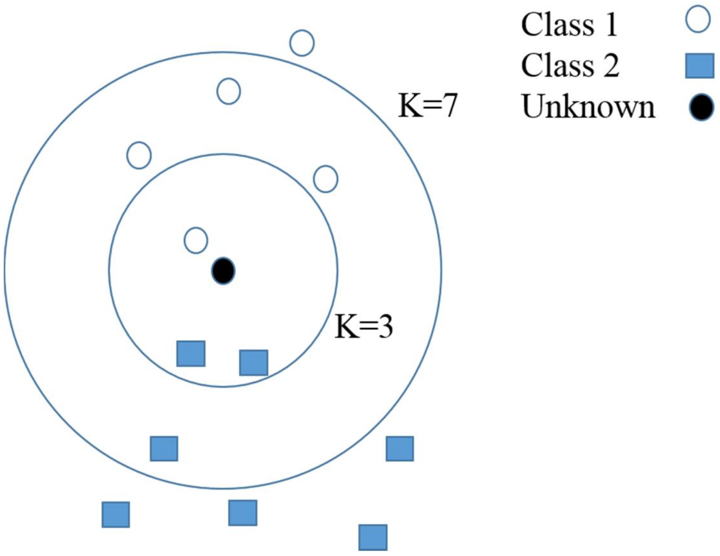


Figure 12: Example of k-NN classification. Unknown data compare a reference label data with the closest distance of the k. In the case of k=3, class1 and class2 are one and two. So unknown data is decided in class2. In the case of k=7, unknown data is decide in class1 by majority vote.

K-Nearest Neighbors algorithm (k-NN) is one of the supervised learning model using a non-parametric method, and is used for classification and regression in pattern recognition [5]. k-NN is a way to search for the K data in the ascending order of distance from the given data and select the number of k in ascending order. Then, the algorithm find the class of selected data by a majority vote. Therefore, k-NN classification method is referred to as the instance-based classification. To calculate the distance use Euclidean distance generally. For example, new input red is determined to green in case of k=3 in Figure. $x_j = \{x_1, x_2, \cdots\cdots, x_n\}$ is the training data with the n-dimension feature. $y_j = \{y_1, y_2, \cdots\cdots, y_n\}$ is a new input. As a result, to calculate the distance of each, the following methods are used.

$$\text{Euclidean distance}_j = \sqrt{\sum_{i=1}^{n} \left(x_{ij} - y_i\right)^2}$$

$$\text{Manhattan distance}_j = \sum_{i=1}^{n} \left|x_{ij} - y_i\right|$$

In k-NN, it is necessary to take time to calculate the distance of all learning data when obtaining the new data and selects the k adjacent data. All data must be stored in order to predict. Therefore, k-NN algorithm need the huge storage capacity and computation problem that takes a long time to compute. In addition, there is a problem but also to select the value of k. In case K is 1, it depends only on the data of the closest, it is sensitive to noise. If K is large, It becomes possible to refer the class of other, Therefore, the result is determined in accordance with the ratio data of each class occupied. Selection of the appropriate k depends on the data.

4.3. Vector Quantization for Clustering

Vector quantization(VQ) is a classical quantization technique quantized the sets of vector [6]. One of the unsupervised model, VQ is called by clustering. Clustering is a method of detecting a group with similar items from the data sets. Rather than finding the right answer, there is an interest in discovering the structure of the data set. Since VQ is no label for the classes, and is one of the unsupervised learning model. Quantization means to compress the input vector space, finds several representative patterns and assigns the input data to them called codebook. VQ has three problems. First, there is the learning of the codebook. That is how to find the optimal set of classification of a given data set. The Second is a quantization problem. That finds the closest code vector for the given vector. Third, there is Finding the closest codebook vector for a given vector.

The method to find optimal classification set is Quantization Function defined by the nearest neighbor rule according to some distance measure

$$q(x_n; Y) = \operatorname*{argmin}_{k} d(x_n; y_k)\,, where\ y_k \in Y$$

In general, vector space is assumed to Euclidian space. In order to solve the optimization problem to find the optimal codebook, using the Hierarchical clustering or k-means clustering. K-means algorithm, a clustering technique separating k sets from n objects based on distance. K-means clustering pick k cluster centers arbitrary when creating groups. Therefore, it is possible to provide different results in accordance with the initial conditions.

```
Algorithm: Agglomerative Hierarchical Clustering
Input $\mathrm{X} = [x_1, x_2, \cdots, x_n], \mathrm{Y} = \mathrm{X}$
$K_f$ : desired number of cluster
$q = [1,2,,\cdots,N], \mathrm{K} = \mathrm{N}$
$\mathrm{n} = [1,1,\cdots,1]$, 1 by $n$ matrix
While($K$ >$K_f$)
$\Delta(k,\mathrm{j}) = \frac{n_k n_j}{n_k + n_j} \|m_k - m_j\|^2, k \neq j, (k,j) \in q$
find the minimum distance $\Delta_{k,j}$
update $y_k = \frac{n_k y_k + n_j y_j}{n_k + n_j}$
  $n_k = n_k + n_j$
  delete $q_j, y_j, num_j$
  reduce $K = K - 1$
End
```

We implement Agglomerative hierarchical clustering implemented Ward's method. Ward's method use the distance between two clusters, k and j, is how much the sum of squares will increase when we merge them. n and y means the number of point included in each cluster and centers of the clusters. Δ is the merging cost of combining the clusters k and j. K_f of desired clusters are made by the distance measure of these and y_k of the clusters is to be codebook. When there is a new input x, its class label is decided by minimum distance between codebooks.

$$output = arg\min_{k} |x - y_k|^2$$

5. Related work

Luca proposed gesture based authentication on android phone that authenticate user by using geometric pattern and drawing timing [7]. By adding an implicit authentication layer to password pattern, they enhance its security. It operates in the mobile phone of Android, Nexus One. Luca's method use dynamic time warping (DTW) to analyze a time series of touch screen data such as all combinations of x coordinates, y coordinates, pressure, size and time. In time series analysis, DTW is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed [8]. Their work has limitations compared to our work. In the aspect of security, false positive rate is very important because a case that false positive is happened consider as occurring to serious problem in the characteristic of security .Their work show false positive rate is too higher about 21% than our method.

Shahzad proposed GEAT gesture based authentication on Windows phone [9]. Their work show performance is better than Luca's method. Their method is conduct given gesture is appeared on touch screen. GEAT operates in Samsung mobile phone of windows. GEAT implemented classification Support Vector Distribution Estimation (SVDE) with Radial Basis Function (RBF) kernel in libSVM.[10]. SVDE is called one-class SVM. SVDE was proposed by Scholkopf at al. (2001) for estimating the support of a high-dimensional distribution [11]. They measured parameters of SVDE such as γ, parameter for RBF, and ν, a parameter for SVDE by grid search with 10 cross validation on each training group. Cross validation was performed based on the true positive rate only. The training samples are only from legitimate user. For evaluation, they conduct Matlab simulations and real-world test. Real-world test was conducted based on shoulder surfing attack. In real-world experiments, results show TPR and FPR are 98.2% and 1.1%.

Saevanee proposed password user authentication by using additional information such as interval between sequential touch, duration and pressure is generated when user touch on the touchscreen [12]. Gaines showed different people have unconscious unique rhythm pattern when they typed [13]. Saevanee's method operates in the notebook touch pad and catch signals such as the measurement of force and value of keystroke dynamics when the user enters 10-digit number on the touch pad. Their work use k-NN classification method.

Meng proposed a behavioral biometric methods called touch dynamics [14]. Average touch movement speed and fraction of touch movements with direction, average single-touch time, average multi-touch time, number of touch movements per session, number of slow touches, touch duration and touch direction is used. In the method of Meng's, multi-touch is implemented for processing the plurality of fingers is on a touch screen at the same time such as the method Shahzad proposed. Meng's method creates the authentication signature for the legitimate user, make model, and compare new input with model. Experiment operates in the touch screen mobile phone of Android. In order to classify the user, they used NaïveBayes, Decision tree, the Radial Basis Function Network and Back Propagation Neural Network.

Y.Niu proposed gesture based authentication [15]. It is a gesture that uses the forearms and wrist that is different from the gesture of the above. This is similar to motion. While user holding the mobile phone performing the gesture, the user can tap the screen with the thumb. They have the main target to capture a biometric quantity of muscle memory and physical characteristics of the specific user by using the gesture. Classification used a DTW. The experiment was carried out in mobile phone of Android.

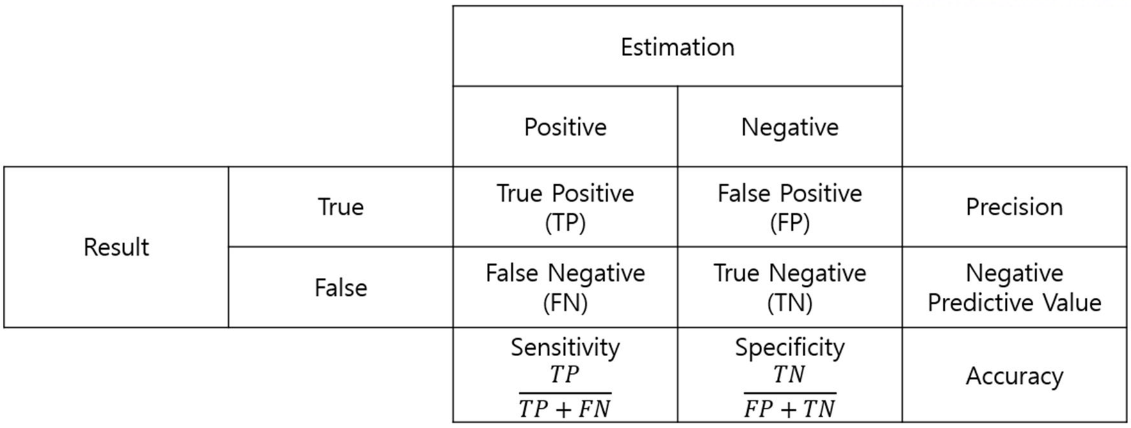
6. Experimental Results

6.1. Experimental Setting

We assumed that the drags of the people represent the person. To test this, experiments were carried out twice. We developed drag catching and authentication program on Samsung Galaxy Note 2. In case user try to drag on touch screen, the program gather each touch point, time to record touch point, touch area on touchscreen, touch area on finger, and gravity sensor. We took advantage of Google API for extracting touch data, it brings the value of once in 17~18ms in general.

First, we did experiments that occurred in a short period of time. We recruited volunteers of 15 people. This experiment made the drag of 25 times with 100 times. Second, we found 17 volunteers, the experiment was conducted over 5~7 days. We ask volunteers conduct to make 30 drags with short break time and 2 times at least with long interval. We need to check drags are consistent when time goes by and posture are changed. We ask to perform a drag grabbed in a posture that has been used usually to volunteers. Third, we conducted real-world test. We found 32 volunteers. We show video to make the drag in case of success to unlock for 2 owners to imposters, then they tried to imitate the drag.

In this section, We show the experimental results of DragID. The first, our evaluation is reported from accuracy and the ROC curve using Matlab simulations. ROC (Receiver Operating Characteristic) curve is a graph that show relationship between sensitivity and specificity. For the ROC curve, definition shown in the following figure is required.



		Estimation		
		Positive	Negative	
Result	True	True Positive (TP)	False Positive (FP)	Precision
	False	False Negative (FN)	True Negative (TN)	Negative Predictive Value
		Sensitivity $\frac{TP}{TP+FN}$	Specificity $\frac{TN}{FP+TN}$	Accuracy

Figure 13: Accuracy evaluation table.

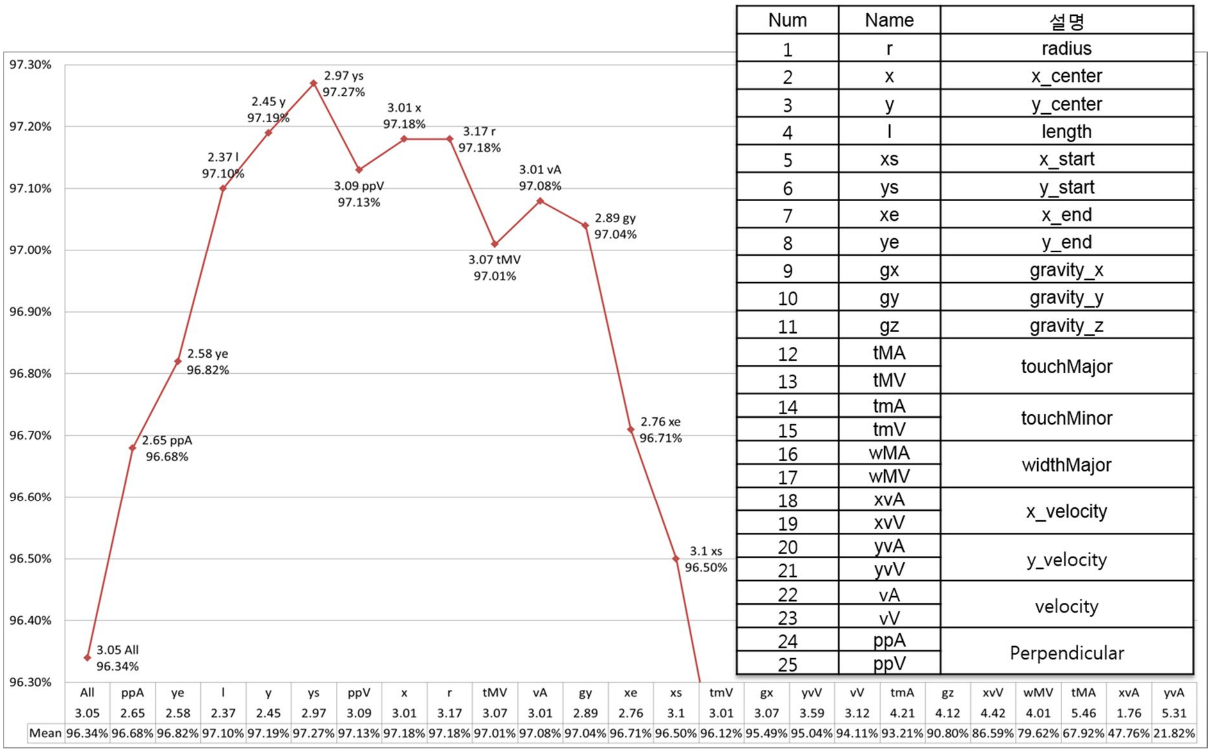
In this figure, condition is status should be predicted, outcome is a result judged by machine learning. Therefore, means to be correctly classified. False means to be incorrectly classified. Positive and negative mean identified and rejected respectively. X axis of ROC is FPR (False Positive Rate). Y axis of ROC is TPR (True Positive Rate). TPR is same as sensitivity. FPR is same as $1 - \text{specificity}$. The more graph is drawn near the top of the left, the more classification performance is superior.

Second, it searches for an optimum parameter values of SVM corresponding to TPR value using a grid search. Third, We present transition of FNR (False Negative Rate) and TPR through the adjustment of the threshold SVM.

6.2. Accuracy Evaluation

In the first experiment, it was undertaken to investigate whether can drag representing human. The simulation was done in Matlab. Result of simulation was made average accuracy of 5-fold cross validation with the data divided into five equal parts. Figure 8 shows the True Positive of 96.34% when using the features of 25.

We implemented SBS (Sequential Backward Selection) for importance of feature. SBS is an optimization technique for selecting features automatically in case features of the n number are given. Feature selection algorithm is important in classification. This reduces the dimensionality of the feature space, improves the speed and performance. Performance of the classifier can be reduced by redundant feature if feature space is large. Performance is improved on 97.27% through the SBS.



Num	Name	설명
1	r	radius
2	x	x_center
3	y	y_center
4	l	length
5	xs	x_start
6	ys	y_start
7	xe	x_end
8	ye	y_end
9	gx	gravity_x
10	gy	gravity_y
11	gz	gravity_z
12	tMA	touchMajor
13	tMV	
14	tmA	touchMinor
15	tmV	
16	wMA	widthMajor
17	wMV	
18	xvA	x_velocity
19	xvV	
20	yvA	y_velocity
21	yvV	
22	vA	velocity
23	vV	
24	ppA	Perpendicular
25	ppV	

	All	ppA	ye	l	y	ys	ppV	x	r	tMV	vA	gy
	3.05	2.65	2.58	2.37	2.45	2.97	3.09	3.01	3.17	3.07	3.01	2.89
Mean	96.34%	96.68%	96.82%	97.10%	97.19%	97.27%	97.13%	97.18%	97.18%	97.01%	97.08%	97.04%

	xe	xs	tmV	gx	yvV	vV	tmA	gz	xvV	wMV	tMA	xvA	yvA
	2.76	3.1	3.01	3.07	3.59	3.12	4.21	4.12	4.42	4.01	5.46	1.76	5.31
Mean	96.71%	96.50%	96.12%	95.49%	95.04%	94.11%	93.21%	90.80%	86.59%	79.62%	67.92%	47.76%	21.82%

Figure 14: Result of first experiment. This graph show change by SBS. Coordinates of x and y mean removed features and True Positive Rate.

In a second experiment, we looked for a volunteer of 17 people to visit more than 10 times a week to make a drag data. To confirm that the position using the mobile phone is similar as time goes by, volunteers collected over a long period of time. We divided into seven sets in accordance with sequential time. We performed to verify the accuracy by using the cross-validation.

SVM output is assigned to 0 in this experiment. TPR shows 89.49%. TPR of CV1 is low because the experiment was performed prematurely. It is possible to check the increase in TPR as CV number goes up to. Sequence is a method for increasing the TPR and reducing the FPR to prevent the output of the SVM occurred false positive by chance. If the True Positive occurs three times consecutively, SVM output should be regarded as the true positive of one. If SVM outputs are wrong three times consecutively then reject. In case wrong is happened once or twice in third, give the user a chance again with a pass. TPR shows 92.33% to confirm the rise using sequence technique.

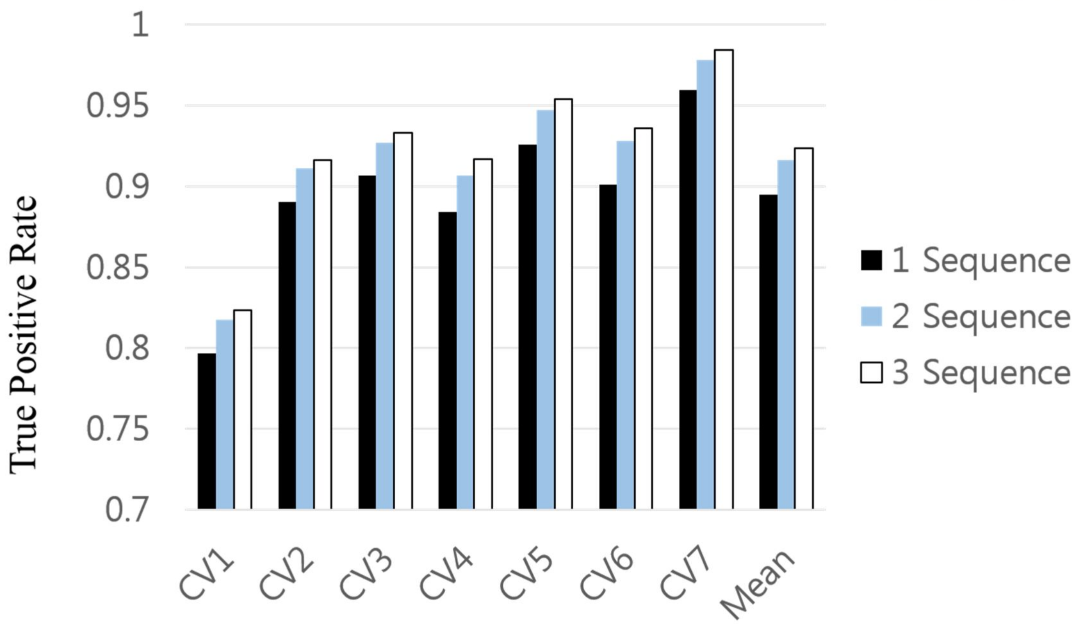


Figure 15: Result of second experiment. "CV" means cross-validation.

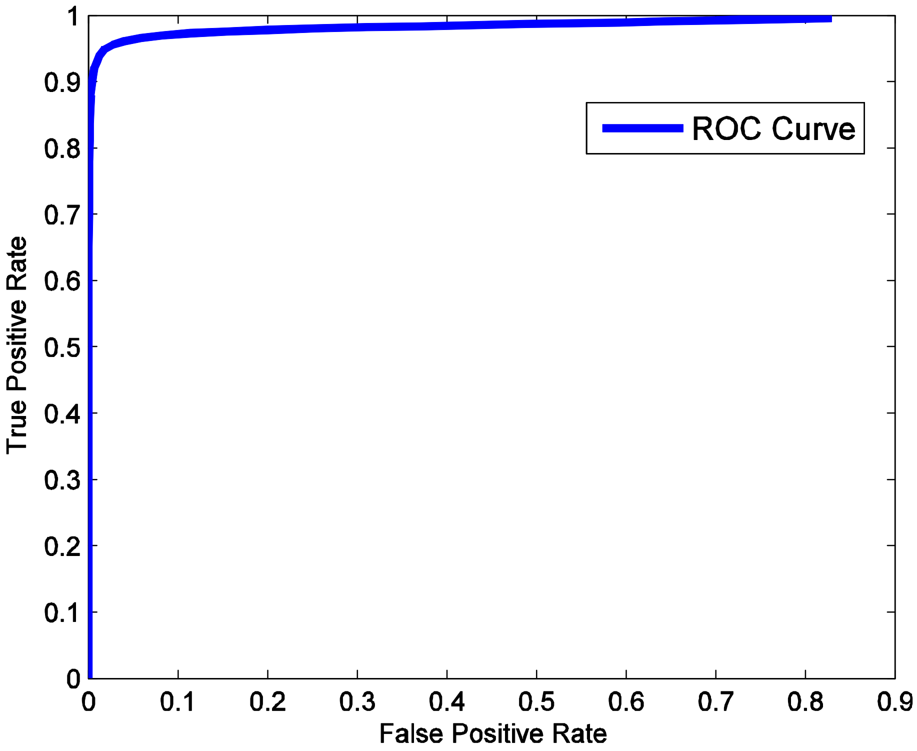


Figure 16: Change of TPR and ROC curve in accordance to SVM threshold.

Figure 16 is ROC curve for the user included training. In authentication, low false positive is more important than true positive. SVM is a classifier very good in aspect of ROC curve. True negative is 99.64% in case the SVM threshold is 0. The SVM classify TN well.

6.3. Real-word Evaluation

For Real-world Evaluation, we found volunteers of 17 and 19 people for two models respectively. In the experiment, it is executed in the Samsung galaxy note2 on Android. To create the models, two people made training data for 10 to 15 minutes. For each set, we took a video of 30 times drag generation process to unlock for legitimate users. We asked volunteers to watch the video and to create a 50 drag similar to the drag seen. Figure15 is the result of SVM when 19 people were trying to shoulder surfing attack for a single model. Shoulder surfing attack means looking over someone's shoulder to get information [17]. Figure 15 shows TPR and FPR turn out to be 91.65% and 3.05%.

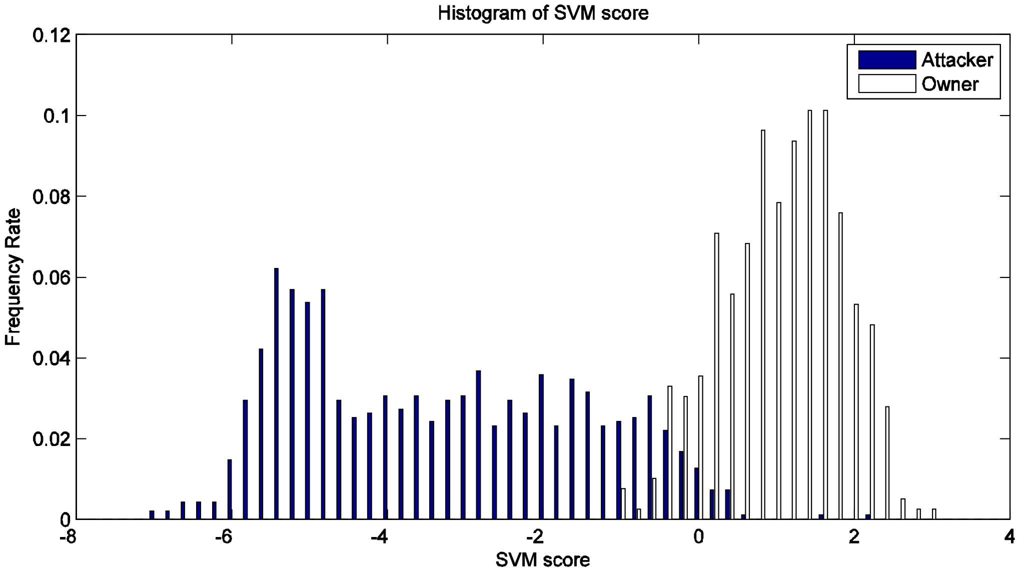
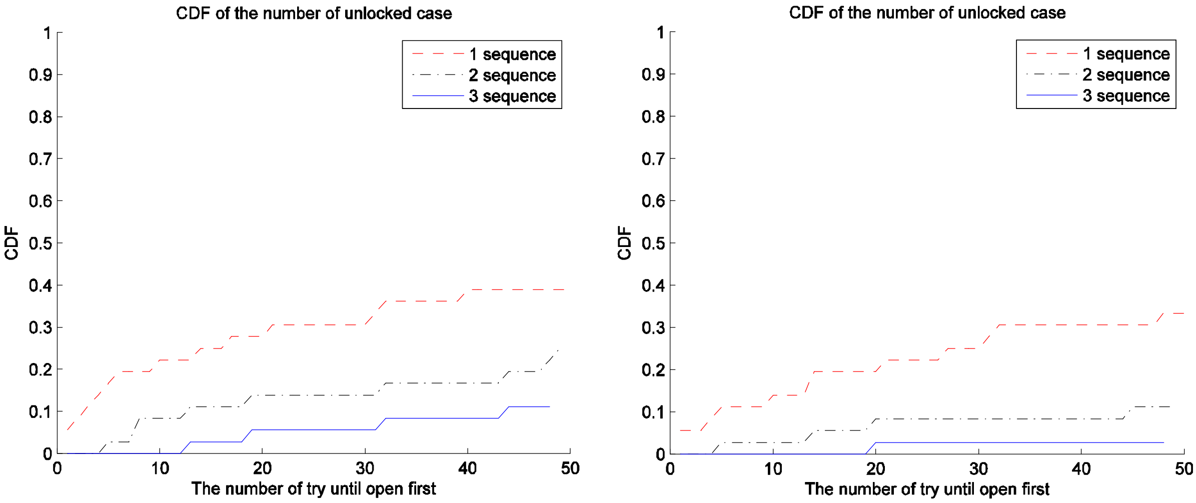


Figure 17: Histogram of SVM score for one model. SVM threshold is 0. Blue bars located at over 0 are false positive.

In order to analyze the experiments with the actual application for the method proposed by us, we must consider following. First, we have to consider again ROC curve defined by the first experiment. In the training data, false positive does not actually exist because false positive is generated when other people imitated the training data after collecting the training data. So there is a limit to collect training data. When it comes to viewing in the ROC curve, collecting false positive data improve the performance. However, it is difficult in a real situation. Second, it is the definition of performance. In case of taking place FP once, it is very fatal. Therefore, it is necessary to define the number of try first time unlock (i.e. false positive) occurred when an imposter tried to unlock rather than to be defined in relation between FPR and TPR. Figure 16 shows the CDF of the unlock case depending on the threshold. (a) shows that the unlock occurs once when imposters try 13 times at least in case of using the techniques of three consecutive. Unlock was happened 11.11%. In case threshold is adjusted to 0.3 in order to reduce the occurrence of FP, the unlock occured with a probability of 2.78% when imposters tried 20 times at least. When imposter try imitation, output access to the threshold often because there are many features. It can be seen in the hope to improve performance by adjusting the threshold.



(a) (b)

Figure 18: CDF of the number of unlocked case. (a) is CDF of threshold=0. (b) is CDF of threshold=0.3

6.4. Comparison with Classification

Classification	3-NN	5-NN	7-NN	9-NN	SVM
True Positive Rate	28.94%	31.35%	32.55%	33.31%	89.49%

Table 3: Comparison of classification. NN and SVM mean “Nearest Neighbor” and “Support Vector Machine”

Table 2 shows the results of TPR of SVM and NN method. NN shows very low TPR compared to the NN SVM. It does not consider the weight of the distance corresponding to the range of each features and just uses Euclidean distance.

6.5. Comparison with Existing Schemes

Method	Gesture	TPR	FPR
Luca et al.	Swipe down-1 finger	98%	50%
GEAT	Swipe down-1 finger	95.71%	10.71%
DragID	Circle shape drag	91.65%	3.05%

Table 4: Comparison of DragID with GEAT and [6]

Although Luca's method and GEAT was also possible to use a variety of gesture, but they used the similar simple gesture like circle shape drag. Table 3 shows the comparison of the TPR and FPR obtained when utilizing simple gestures. As performance due to fall in comparison with the other gesture, GEAT didn’t usie the gesture. It shows excellent performance with the authentication using a simple gesture.

7. Limitations and Discussion

7.1. Limit of Posture

The experiments are performed with four positions because we guess the value of the feature change when using a mobile phone in various poses. DragID use hardware such as gravity sensor and touch screen. Gravity sensor shows mobile phone is placed in any state when the people take the mobile phone. In general, the value of the gravity sensor is similar until the posture people grab mobile phone. However, posture is varied because smartphone is the mobile device. It is possible to lean or lie when using for people mobile phone. If you use a mobile phone with leaning or lying posture, the value of the feature changes because the position of the arm and wrist position are different. If input is generated during people don't use on trained posture, performance should decrease. Thus, data that are used in various positions is required.

7.2. Size of Phones

Methods such as size considerations are necessary if for the application of other phones. There are difference of phones such as bezel, screen, button and so on. External appearance of the smartphone has an effect on the value of the feature. DragID uses as features such as end point and start. If smartphones changes and the shape of a human hand does not change old feature values need to change. Because the experiment was performed in the same size phone on Galaxy Note2, we did not consider the screen size in this experiment. Methods such as taking into account the size is necessary in order that it is implemented on other equipment.

7.3. Dynamic Update of Learning

In order to improve the performance, FP or FN data is highly efficient. However, acquisition of data of the False is virtually impossible. In a real environment, confirmation whether owner or imposter is not possible when the input occurs. It is hard to take dynamic update because there is a difficulty of owner confirmation.

8. Conclusions

In this thesis, we introduced classification, performance evaluation, and the DragID authentication system to analyze the pattern of the drag to model the people. We hypothesized posture taking mobile phones is substantially same. The patterns of the drags vary from person to person because the shape of the hand does not change. To prove this, the drags are possible to represent the people from the first experiment. We showed 96.34% of TP from the simple experiment. In the second experiment, it was shown that posture taking phone over time is almost same. Real-world experiment shows higher performance for shoulder surfing attack of imposters. Strengths of DragID is as follows. A period of collecting training data is short. In the real-world experiments, we showed a good performance for shoulder surfing attack by using the training data of 10-15 minutes. If we got more training data, DragID should have better performance. We expect DragID will have higher performance by collecting data and updating the model when users unlock every time in application. However, a drawback of DragID system is difficult to collect FP data in actual environment.

REFERENCES

1. "Global mobile statistics 2014 Part A: Mobile subscribers; handset market share; mobile operators", http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#subscribers.
2. "The symantec smartphone honey stick project".
3. "Chaos Computer Club breaks Apple TouchID", http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid
4. C. Cortes, V. Vanpnik, "Support-Vector Networks", Machine Learning, 20, 273-297, 1995
5. T. Cover, P. Hart, "Nearest Neighbor Pattern Classification", Information Theory, 1967
6. T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning", Springer Book, Second Edition, 2009
7. D. Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you! Implicit Authentication based on Touch Screen Patterns", in Proc, ACM(CHI), 2012.
8. M. Muller, "Dynamic time warping", Information Retrieval for Music and Motion, Springer Book, 2007
9. M. Shahzad and A. X. Liu, "Secure Unlocking of Mobile Touch Screen Devices by Simple Gestures – You can see it but you can not do it", in Proc, ACM(Mobisys), 2013.
10. C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines", ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp.27:1-27, 2011.
11. B. Scholkopf, J. C. Platt, J. Shaew-Taylor, A. J. Smola, R. C. Williamson "Estimating the Support of a High0Dimensional Distribution", Journal of MIT Press, vol. 13, no. 7, pp. 1443-1471/
12. H. Saevanee, "User Authentication using Combination of Behavioral Biometrics over the Touchpad acting like Touch screen of Mobile Device", ICCEE, 2008.
13. R. Gaines, W. Lisowski, S. Press and N. Shapiro, "Authentication by keystroke timing: some preliminary results", Rand Report R-2560-NSF, Rand Corporation California, 1980.
14. Y. Meng, D. S. Wong, R. Schlegel, and L.-for Kwok. "Touch Gestures Based Biometric

Authentication Scheme for Touchscreen Mobile Phones”, ICISC, 2014.

15. Y. Niu and H. Chen, “Gesture Authentication with Touch Input for Mobile Devices”, MOBISEC, 2011.