User-Authentication on Wearable Devices Based on Punch Gesture Biometrics

Guan-Cheng LIANG[a], Xiang-Yu XU and Jia-Di YU

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

Abstract. Due to commoditization and convenience, wearable technology are interwoven with our daily life. However, privacy sensitive data stored on those devices such as personal email, message can be easily stolen. Most devices require a PIN input to unlock. However, this mechanism is vulnerable to shoulder surfing attack. Thus many novel authentication approaches have been proposed to solve this problem. And biometric-based methods have been adopted by many researchers because of the efficiency and excellent performance. In this paper, we propose a new biometric-based authentication system. We focus on how the user performs a straight punch gesture subconsciously. By analysis the acceleration data from the smartwatch when user performing the gesture, we are able to profile the user. And we authenticate the user according to the biometrics of this action. This mechanism is light-weighted and do not require user to remember any secret code. We develop an authentication system on Samsung Gear Fit 2 and conducted a real-world experiment on 20 volunteers. And we collected 13000 gesture samples to evaluate our system. Results show that our system can achieve a classification accuracy of at least 95.45%. In attacking scenario, our system can achieve an equal error rate lower than 4%. The maximum number of samples required by a well-trained classifier is 25.

1 Introduction

Wearable devices have growing popularity in recent days. Products such as smartwatch and smartband are inexpensive and provide good usability. People are allowed to access social media, e-pay apps through these equipment. However the personal information within brings us the security concern. Attackers may steal the device to snoop sensitive information which results in fatal aftermath. Hence recognizing the identity of the user has great significance.

Unlike smartphone or tablets, wearable devices are usually equipped with small touchscreen, which is only sufficient to perform PIN input. This scheme is defenseless against shoulder surfing attack. Recently, Huawei or Apple enable smartwatch and smartphone to unlock each other after they are paired [1, 2]. This approach leads to more severe consequences for both devices once either the smartphone or wearable device is stolen. Due to resource constraints for wearable devices, researchers turned their focus to biometric-based authentication. For example, resonant properties of human hand are used to establish a secure channel between two devices [4]. However this method is sensitive to the location of the hardware on human hand. Meanwhile Gait based recognition suffers from long training process[5–7].

[a] Corresponding author: liangguancheng@sjtu.edu.cn

For practical use, the authentication system should be light-weighted, e.g. no extra hardware involved and easy to use. Therefore we focus on using hand gesture as a unique behavior of user for authentication. However previous gesture based system requires user to perform very complicated gestures which can be quite cumbersome[3].

In this work we identify the user based on the biometrics generated by punch gesture. Since each person has a unique way of doing a punch gesture which reveals the information of one's identity. And this gesture depends on wingspan, strength, and personal habit et.al. Moreover, the punch gesture is a subliminal action that doesn't need to remember.

There exist several challenges of realizing this system. First, the accelerometer data contains some noise that may cause uncertainty in our system. Thus we apply a low pass filter to remove high frequency noise. Second, we need to identify the features within the recorded data we should use to distinguish different users. To this end, we conduct a close analysis on the collected data to extract useful features for authentication. Third, the attacker may imitate the gesture performed by the legitimate owner to get access to the device. Thus we evaluate the effectiveness of our system in such scenario. The results show that our system can effectually defeat against the imitation of the attackers.

In particular, the legitimate owner wearing a device performs a punch gesture while we collect the readings from accelerometer. Then we use the collected data to extract training dataset and train an effective classifier offline. Later the data is collected from the unidentified user who wears the device, if it matches the pattern of the legitimate user, the user is authorized. In real world the user can perform a punch gesture at any place any time.

We summarize our contribution in this work as follows:

- We conduct a close analysis on the accelerometer data generated by punch gesture to demonstrate biological diversity among different users.
- We propose and implemented a novel authentication system based on biometric of punch gesture.
- We evaluate our system on collected traces of 20 volunteers during 20 days.

The rest of the paper is organized as follows: Section II reviewed the related work on wearable authentication. Section III conducted a close analysis on biometric of punch gesture. In Section IV we present the implementation of our system. Then we perform evaluation for authentication and display the result in Section V. Finally, we draw a conclusion in Section VI.

(a) Throw (b) withdraw

Figure 1. Punch gesture.

2 Related work

In this section we review previous works on biometric-based authentication system for wearable devices.

Since wearables are always attached to user's body, gait data can be extracted to identify user. Before smartwatch and other wearables are available, researchers have been studying gait based authentication using wearable motion sensors such as accelerometer or gyroscope[5, 7]. In those works, a wearable sensor is located on a specific area on human body to collect motion data[8–10]. Most recently, a gait based authentication system on smartwatch has been proposed [6]. However gait is

inconsistent when user walks on different kind of surfaces, resulting in low true positive rate. And time consuming training and recognition process are needed for these methods, which is not user-friendly.

Meanwhile, using hand gesture as a biometric for authentication is a promising solution. The sweep or swing of action user while holding the mobile device can be used as biometric for authentication [12, 14]. Yet it's not suitable for wearables since the device is not fixed on the wrist. Continuous gestures of the user, as an out-of-band communication channel, are used to pair wearables and other devices [3]. However those gestures are too complex to perform. For simplicity and usability, we use simple punch gesture as biometric to authenticate the user in this paper.

3 Data collection and analysis

In this section, we give a definition of the punch gesture first. Then we describe the data collection procedure with a set of volunteers. And we conduct a close analysis on the collected accelerometer data.

3.1 Defining punch gesture

The punch gesture is a simple 3D space gesture. We notice that there exist different types of punch gesture in martial arts and combat sports. However, we only adopt the straight punch gesture in this paper. And we assume everyone perform this gesture with the right hand. A person can perform a straight punch gesture as follows: First stand straight and hold the fist within the right chest area. Then throw and withdraw the fist with the way he or she feels comfortable. We present the punch gesture in Fig. 1. The trajectory of throwing and withdrawing almost follows a straight line. This gesture possesses many advantages. First, everyone can perform this gesture in a subconscious way without any specific instruction. Second, it isn't limited to any specific scenario to perform this gesture.

3.2 Collecting data

We developed an application on the Samsung Gear Fit 2 for accelerometer data acquisition. The data are stored locally on the smartwatch. We set the sample rate of the accelerometer to 100 Hz. This

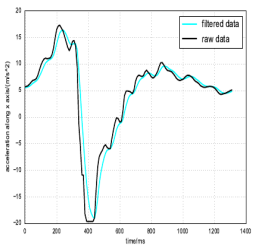
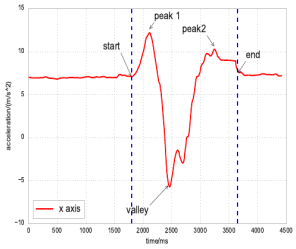
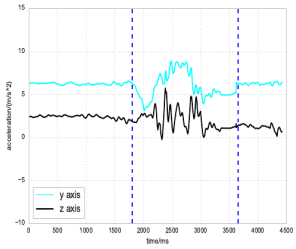


Figure 2. Low pass filter.



(a) x axis



(b) y and z axis

Figure 3. Acceleration data of volunteer 5 performing punch gesture

application will record the accelerometer data when the volunteer performs the gesture. After one gesture is finished, an assistant will press a button on the smartwatch's touchscreen to save the data while the volunteer holds the hand still. The recorded data file is defined as one observation. After 10 observations are complete, the assistant will press a button on the touchscreen to exit the application. The collected data will be transferred to the laptop for further analysis. We recruit 20 graduate students (8 female, 12 male) in the CSE department for our experiment. We label each of them with a unique number. Every day each of them were asked to perform the punch gesture for 10 times, and all observations are labelled with their number. We execute this experiment for 10 days. Together we collected 2000 observations for analysis.

3.2 Analysing data

Accelerometer readings are always containing noise, since the wearable device can sometimes be slippery on the wrist. Thus we need to eliminate noise to obtain more accurate data. We apply a low pass filter on the readings since human action always produce low frequency noise. As we depicted in Fig. 2, the filtered data are smoother than the original data.

We randomly choose 5 out of 20 volunteers, denoted as {V1, V2, V3, V4, V5}, to demonstrate our observation. In our settings, the moving direction of the fist keeps consistent with the x-axis of the accelerometer. Since the moving direction of the fist is almost a straight line, the acceleration produces the most significant change on the axis parallel to this direction. As we depicted in Fig. 3, when V5 perform the punch gesture, we observe a remarkable shift along x-axis while the readings from y and z axis presents inconsistency. This originates from the fact that the acceleration change on y and z axis comes from the slipping on the direction perpendicular to the punch direction. Thus the readings from x-axis retain clearness while the readings from y and z axis suffer from randomness. Therefore we choose the readings along x-axis to detect the beginning and finish time of the gesture by calculating the energy levels[11]. We denote the beginning and the finish time as T0 and TN. We plot 10 segmented observations of V3 in Fig. 4. And one observation for each volunteer is depicted in Fig. 5. We observe that all volunteers share some similar features while at the same time can be distinguished from each other.

First, there exist two peaks and a valley on the segmented data. We label them as ρ_1, ρ_2 and υ. As we can see in Fig. 4, the punch gesture can be divided into two phases, throwing and withdrawing. The volunteer first accelerates to push the fist, and then decelerates in order to change the moving direction at the end of the former phase. Thus we observe an increase follow by a reduction between T_0 and $T_{..}$ Similarly, the second phase of withdrawing the fist experiences an almost symmetrical process. This compliance with our intuition since withdrawing a fist is symmetric with throwing. However, we make some observations on Fig. 4 and Fig. 5. First, we notice that the value of ρ_1, ρ_2 and υ is different among all volunteers. Second, the speed of and decreasing the acceleration shows diversity for different volunteers but maintains consistency for the same volunteer. Third, the duration of the gesture of the same volunteer may varies between gestures, yet is still a good indicator of different volunteers. Finally, when a volunteer starts the gesture, the initial acceleration along 3 axes indicates how the volunteer

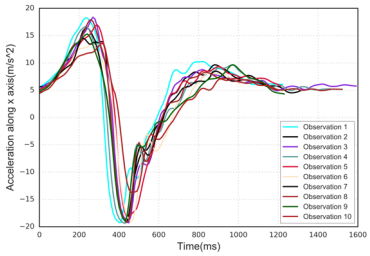


Figure 4. Ten observations of Volunteer 3

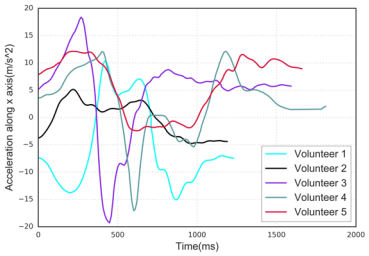


Figure 5. Observation of 5 Volunteers

prefers to hold the fist on still. We define this initial position of the fist as a triple $<acc^I_x, acc^I_y, acc^I_z>$. Therefore, we can leverage the biometrics to recognize the legitimate user and attackers.

4 System design

In this section, we design an authentication system for wearable devices to identify the legitimate user who owns the device and prevent attackers from accessing the device. We only use the acceleration data from accelerometer that is available on all wearable devices.

4.1 System overview

The system architecture consists of two parts. We first collect observations from the legitimate owner

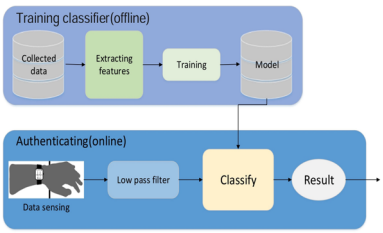


Figure 6. System architecture

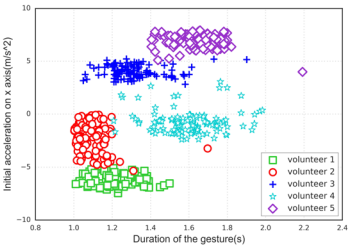


Figure 7. Distribution of feature tuple $<T,acc^{I}_{x}>$ for 5 volunteers

and extract training dataset to learn a classifier in the offline part. Then we utilize the classifier to recognize the identity of the user in online part. We depict the work flow of our system in Fig. 6.

In the offline part, we train an efficient classifier for the legitimate owner. First we collect sufficient observations. For this purpose the legitimate owner is asked to wear the device and perform the gesture for several times. Then we extract effective features from each observation to form a feature vector. Afterwards, the dataset is fed to a machine learning framework thus a classifier for the legitimate owner is obtained.

In the online part, a well-trained classifier is stored on the device ready for authentication. The user is asked to perform the punch gesture to unlock the device. The authentication system first collects the readings of accelerometer and applies a low pass filter to remove noise. Then the feature vector is extracted from the readings. Finally we authenticate the user by feeding the feature vector to the classifier of the legitimate owner. If the result is positive, then the user is authorized to access the device. Otherwise it stays locked for security. As more observations are collected in authenticating process, we add these data to the training dataset to improve the performance of the classifier. However we will show that a small amount of observation is enough for training a useful classifier.

4.2 Feature extraction

Table I. Feature list

Feature	Description	Feature	Description
$acc^{v}_{x,1}$	Value of peak 1	T	Duration of punch gesture
$acc^{v}_{x,2}$	Value of peak 2	acc^{I}_{x}	Initial acceleration on x axis
b_1	Bandwidth of peak 1	acc^{I}_{y}	Initial acceleration on y axis
b_2	Bandwidth of peak 2	acc^{I}_{z}	Initial acceleration on z axis
μ_1	Mean of peak 1	acc^{v}_{x}	Value of valley
μ_2	Mean of peak 2	b_v	Bandwidth of valley
σ_1	Standard deviation of peak 1	μ_v	Mean of valley
σ_2	Standard deviation of peak 2	σ_v	Standard deviation of valley

To train an efficient classifier, we need to extract useful feature. By analysing the collected data in Section III, we choose to use statistical features relevant to ρ_1, ρ_2 and υ, 3 features relevant to the initial position, and the duration of the gesture. We list 16 features we find useful to classify different volunteers. In particular, we provide a description of the features. (1)Acceleration Value: We extract the acceleration of ρ_1, ρ_2 and υ which we denote as $acc^{v}_{x,1}$, $acc^{v}_{x,2}$ and acc^{v}_{x}. (2)Bandwidth: For ρ_1 and ρ_2, we define the Bandwidth as:

$$\min_{0\le t\le T_\upsilon} acc_x^t < acc_x^\rho \cdot \alpha - \max_{T_\upsilon \le t\le T_N} acc_x^t < acc_x^\rho \cdot \alpha \tag{1}$$

Intuitively, we search from the peak to both sides until the value drops below a threshold. It reveals how fast a user performs the gesture. This threshold is relevant to the value of the peak. And we empirically select $\alpha = 0.5$ in our work. Moreover, the valuence of acc_x varies from volunteer to volunteer. Thus we define the left side of the Band of υ as:

$$\min_{T_\upsilon \le t\le T_{\rho 2}} acc_x^t < acc_x^\upsilon + \beta - \max_{T_{\rho 1}\le t\le T_\upsilon} acc_x^t < acc_x^\upsilon + \beta \tag{2}$$

It's similar to Bandwidth of a peak, but both positive and negative value of acc_x can be dealt with. Here we choose $\beta = 2$ empirically. (3)Mean, Standard deviation: We calculate the mean value and standard deviation within the bandwidth of ρ_1, ρ_2 and υ, which we use as statistical features. (4)Duration: The duration of the gesture is extracted using the segmentation method we describe at Section III. (5)Initial acceleration: We define the acceleration value along 3 axes before the user performs the gesture as the initial acceleration, e.g. the readings when user's hand holds still. The time user starts the gesture is T_0, thus we calculate the initial acceleration by average:

$$acc_x^I = \sum_{t=0}^{T_0} acc_x^t / T_0 \tag{3}$$

And acc^I_y , acc^I_z are obtained using the same method. We plot the distribution of all volunteers in 2-dimensional feature space in Fig. 7. It can be seen that we can possibly separate each volunteer from others efficiently utilizing a simple feature tuple.

4.3 Classifier training

In practical scenario, we will only obtain the training data from the legitimate user. Therefore we utilize one-class SVM to train the model. One-class SVM learns a decision function for novelty detection. First we feed the extracted training dataset from the legitimate owner to the algorithm. Here the samples of this dataset are considered as normal observations. The SVM learns a decision boundary that enclose as much as normal observations while excludes outliers. After training process, we can use this model to classify a new data. If a new data is identified as a similar one to the training set, e.g. lies within the closure we learned, we recognize the user as the authorized user. Otherwise we recognize the data as an illegal user's, and reject the user from accessing the device.

We choose the one-class SVM with the Radial Basis Function (RBF) kernel implemented in scikit-learn python package to train the model. In order to find the optimal parameter ν and γ for the model, we perform a grid search on the ranges $2\text{-}18 \leqslant \nu \leqslant 20$, $2\text{-}18 \leqslant \gamma \leqslant 20$. And we do a 10-cross validation on the training dataset. Since the training dataset are all from the legitimate user, we can select the parameters which provide the highest true positive rate (TPR). Even through high TPR is usually accompanied by high false positive rate (FPR). We show in the evaluation section that FPR is totally acceptable in our system.

5 Evaluation

In this section, we evaluate the performance of our system. We ask 20 volunteers to wear a smartwatch on the right hand, and the experiment is conducted for 20 days. The experiment is divided into two sessions. In the first session during the first 10 days, each volunteer is asked to perform 10 gestures for each day. And we train a particular classifier for each volunteer V_i using the data from V_i. Then we test the classifier on the data that are not used in training process. For example, after the classifier for V_i is obtained, we test it on the data of$\{ V_j \mid 1 \leqslant j \leqslant 20 \text{ and } j \neq i\}$. In the second session during the last 10 days, first we ask one volunteer to perform the gesture for 5 times and we record the volunteer's action. Afterwards, we show the video to 5 of other volunteers and ask them to imitate the

gesture of the volunteer in the video for 10 times. In this scenario other volunteers are attackers. We repeat this process for each volunteer at each day. Together we obtain 13000 observations.

Table II. Total accuracy

Volunteer	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Accuracy(%)	99.43	99.13	95.45	97.72	98.24	98.77	96.52	98.35	99.11	99.17
Volunteer	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Accuracy(%)	99.32	95.83	97.65	98.73	97.18	98.53	99.62	97.22	97.19	98.04

5.1 Metrics

We define the metrics as follows to evaluate the performance of our system:

- *Accuracy*: The probability that the classification of an observation A is actually the label of A.
- *Precision*: The probability that an observation A accepted by the classifier is actually an observation of the legitimate owner.
- *Recall*: The probability that an observation A of the legitimate owner is accepted by the classifier.
- *False positive rate(FPR)*: The probability that an observation A of the illegal owner being accepted by the classifier.
- *Equal error rate(EER)*:The value of FPR such that 1-TPR equals FPR.

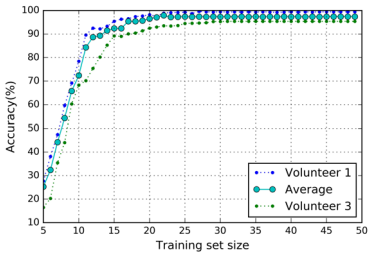


Figure 8. Accuracy under different size of training set.

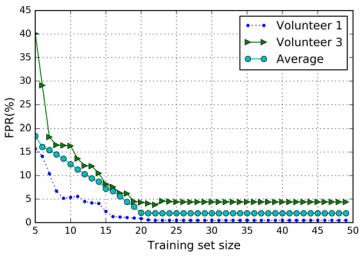


Figure 9. FPR under different size of training set.

5.2 Attacking without imitation

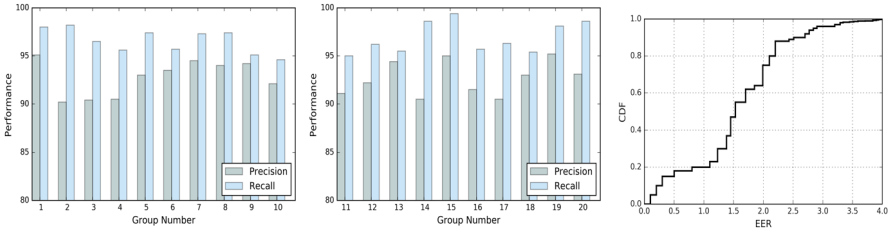
First we evaluate our system on the dataset collected from the first session. We report the result in TABLE II. It can be seen that the lowest accuracy is 95.45% for volunteer 3. The total accuracy for all volunteer is above 95%. Despite that we only use the observations from the legitimate owner in training phase, the one-class SVM can guarantee the exactness of the system. Although more observations from the legitimate owner gives better result in our system, it's practical to use as less as observation to achieve the appropriate accuracy. In this way, the training process can finish quickly and the legitimate owner can use our system in real world. We can feed more observations from the legitimate owner to the SVM for better performance later. Therefore, we show the performance of our system when using different size of training set in training process in Fig. 8. We notice that the classifier of Volunteer 1 can achieve 99% accuracy using only 25 observations. Moreover, Volunteer 3 can achieve 94% accuracy using at most 30 observations. On average, each volunteer can obtain the highest accuracy using at most 25 observations.

However, since we choose the parameters to obtain highest TPR in training phase, more observations is needed to decrease the FPR of the system. Fig. 9 shows that although the FPR can be beyond acceptable when we own few observations from the legitimate owner, it will drop to the lowest value using a small number of observations. In worst case scenario, i.e. for Volunteer 3, FPR drop below 5% using at most 25 observations. This indicates that even we miss the negative observations from the

illegal owners, a well-trained classifier for the legitimate owner can separate illegal owners from the authorized one effectively.

5.3 Attacking with imitation

In this scenario, we have 20 group of 6 volunteers. Within each group there are one legitimate owner and 5 attackers. We exploit the well trained classifier for each legitimate owner from the first session and test it on the data collected from the second session. We present the results for each group in Fig. 10. We observe that the recall is above 95% for each group, indicating that the identity of the legitimate owner



(a) Group 1 to 10. **(b)** Group 11 to 20. **(c)** CDF of EER.

Figure 10. Performance

can be effectively recognized. And the precision is slightly lower since the attackers are now able to observe the legitimate owner closely for imitation. However the attackers can rarely hack into the device by imitation since the precision is above 90%. Therefore the device is safeguarded.

To better demonstrate the usability as well as security of our system, we calculate the EER for each group. Since lower TPR results in lower FPR, we preserve some classifiers with lower TPR during grid search. We test those classifiers on the collected data to calculate FPR. Afterwards we have several TPR-FPR tuples. Using ROC curve we can calculate the EER [13]. We plot the cumulative distribution function(CDF) of the EER in Fig. 10(c). It shows that our system can achieve an EER lower than 4% which is good enough for practical use.

Conclusion

This paper presents a novel authentication system using single punch gesture. The intuition that each person performs this gesture in a unique way leads to this work. We first collect observations of volunteer performing this gesture and recognize the characteristic. Then we build a classifier for each volunteer based on one-class SVM to recognize the legitimate owner. We evaluate the performance of our system on 20 volunteers in two sessions. We collect 13000 observations in total and the results show that our system can achieve an accuracy of at least 95% using at most 25 observations from the legitimate owner. When the attacker access the system by imitation, our system can achieve an EER lower than 4%.

References

1. Apple watch. http://www.apple.com/watch/
2. Huawei watch. http://consumer.huawei.com/en/wearables/ huaweiwatch/index.htm
3. I. Ahmed, Y. Y, S. Bhattacharya, N. Asokan, G. Jacucci, P. Nurmi, and S. Tarkoma, Proc. UbiComp 15, 391-401(2015)
4. W. Wang, L. Yang, Q. Zhang, Proc. UbiComp 16, 670-681(2016)
5. J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. M. Makela, and H. J. Ailisto, Proc. ICASSP 05, 973-976(2005)
6. A. H. Johnston, G. M. Weiss, Proc. BTAS 15, 1-6(2015)

7. D. Gafurov, K. Helkala, T. Sondrol, J. Computers **1**, 51-59(2006)
8. M. Derawi, P. Bours, K. Holien, Proc. IIHMSP 10, 312-317(2010)
9. R. Liu, J. Zhou, M. Liu, X. Hou, Proc. ICIEA 07, 2654-2659(2007)
10. T. liu, Y. Inoue, K. Shibata, Measurement **42**, 978-988(2009)
11. J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, M. Gruteser, Proc. MobiCom 15, 142-154(2015)
12. F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto, A. Koike, Proc. ISPACS 06, 45-50(2006)
13. K. Murphy, Machine Learning: A Probabilistic Perspective. MIT(2012)
14. K. Matsuo, F. Okumura, M. Hashimoto, S. Sakazawa, Y. Hatori, ICB 07, 211-221(2007)