Smartphone User Authentication Based on Holding Position and Touch-Typing Biometrics

Yu Sun[1, 2, *], Qiyuan Gao[3], Xiaofan Du[3] and Zhao Gu[3]

Abstract: In this advanced age, when smart phones are the norm, people utilize social networking, online shopping, and even private information storage through smart phones. As a result, identity authentication has become the most critical security activity in this period of the intelligent craze. By analyzing the shortcomings of the existing authentication methods, this paper proposes an identity authentication method based on the behavior of smartphone users. Firstly, the sensor data and touch-screen data of the smart phone users are collected through android programming. Secondly, the eigenvalues of this data are extracted and sent to the server. Thirdly, the Support Vector Machine (SVM) and Recurrent Neural Network (RNN) are introduced to train the collected data on the server end, and the results are finally yielded by the weighted average. The results show that the method this paper proposes has great FRR (False Reject Rate) and FAR (False Accept Rate).

Keywords: Smartphone, user authentication, support vector machine, recurrent neural networks.

1 Introduction

With the development of cloud computing and mobile computing technologies [Xu, Jiang, Wang et al. (2018); Xu, Wei, Zhang et al. (2018)], increasing amounts of private and sensitive information is stored in our smartphones. About 93% of smartphone users store private information in their smartphones [Sucasas, Mantas, Oliveira et al. (2018); Ghaffari, Ghadiri, Manshaei et al. (2017)]. Therefore, the smartphones that store personal and valuable information have become very attractive targets for attackers. Identity authentication is an essential way to prevent the breech of privacy from attackers on the smartphone [Song, Cai and Zhang (2017)].

Traditional mobile phone authentication methods are mostly code, Nine palace map, fingerprint [Wang and Hu (2011)], and face recognition [Ge, Zhao, Li et al. (2019)]. However, these authentication methods can only provide a one-time protection component. For instance, once the first line of defense is invaded, the smartphone will lose its ability to resist completely.

[1] College of Business Administration, Shenyang University, Shenyang, 110044, China.

[2] College of Business Administration, Northeastern University, Shenyang, 110169, China.

[3] Software College, Northeastern University, Shenyang, 110169, China.

[*] Corresponding Author: Yu Sun. Email: sunyu@syu.edu.cn.

CMC. doi:10.32604/cmc.2019.06294

Fortunately, we all have unique “living passwords”, such as fingerprints, facial characteristics, voice tones, optic components, and even behavioral attributes. They are the unique and distinctive characteristics of individuals popularly known as “biometric signatures”. User behavior is just one way of reflecting a person’s identity. It determines whether the current user is a legitimate user by collecting and training characteristic data in order to realize identity authentication. It mainly has the following advantages:

1) It is a completely back-end program, meaning the entire process of collecting data and verification does not require the user to do anything, which sounds very friendly to the user.

2) It does not require complex hardware support, needing only a smartphone with normal sensors and a touch screen. Even low-end devices are capable of running the program.

3) It does not conflict with other identity authentication methods.

4) It transforms security certification from one-off to continuous process. The password is considered a legitimate user once it is confirmed, but behavior-based identity authentication is a long-term ongoing certification process.

5) It is hard to steal unless the intruder can perfectly mimic the owner’s behavior, however this is almost impossible.

In 2007, a continuous authentication method was proposed by Ahmaed and Traore who collected data in the daily work of the users, extracted the eigenvalues of the data, and used the artificial neural network method to do the comparative analysis. In 2012, Nickel et al. [Nickel, Wirtl and Busch (2012)] collected users’ walking posture data through an accelerator, and used the knn algorithm to discriminate the user, obtaining 3.97% FAR and 22.22% FRR. In 2015, Yang et al. [Yang, Guo, Ding et al. (2015)] analyzed the accelerators of mobile phones and obtained 15% FAR and 10% FRR results after collecting samples of 200 users. In 2017, Lee et al. [Lee and Lee (2017)] proposed a smartphone user authentication scheme with using sensors and contextual machine learning. The current accuracy rate based on the identity authentication of smartphone users cannot achieve the high recognition rate such as the combination of multiple devices. Therefore, this paper extracts the data of mobile phones by using many different aspects while defining them separately using SVM and RNN algorithms to carry out training and discrimination. Finally, the research obtains the final identity authentication results through weighted average. What’s more, the experimental analysis of the method presented in this paper shows that the method has better FRR and FAR.

2 Preliminaries

2.1 Support vector machine

The Support Vector Machine (SVM) method is based on the VC dimension theory and the structural risk minimization principle of the statistical learning theory. The algorithm for training SVM comes down to solving the constrained convex quadratic optimization problem. For small-scale secondary optimization problems, Newton’s method and interior point method can be used. When the data size is large enough, fast algorithms and sequential minimum optimization algorithms can be referred to.

Linearly separable and linearly inseparable cases may occur for different data sets. A linear SVM is a classifier that distinguishes different categories of data by finding a hyperplane with the largest edge. The decision boundary of a linear classifier can be written as follows:

$$w \cdot x + b = 0 \tag{1}$$

where w, b are the parameters of the model, and each sample is represented as a binary group (x_i, y_i), where $x_i=(x_{i1}, x_{i2}, \dots x_{id})^T$ can predict any test sample Z in the following manner. The class number is as follows:

$$y = \begin{cases} 1 \ if \ w \cdot z + b > 0 \\ -1 \ if \ w \cdot z + b < 0 \end{cases} \tag{2}$$

What we care about is how to calculate w and b. Here we train the user feature template. The kernel function we selected is the radial basis (RBF) kernel, as shown in Eq. (3).

$$K(x_i, x_j) = \exp\{-\frac{|x_i - x_j|^2}{s^2}\} \tag{3}$$

The work after the selection is to adjust the parameters, that is, the parameters of Eq. (3), obtain a hyperplane through training, and separate the samples of different users to achieve the purpose of identity authentication.

2.2 RNN neural network

RNN is an improved model [Lai, Jin and Yang (2017)] of traditional neural network DNN [Prasad (2017)], by connecting several neurons in a series, using the last neuron outputs as a part of the input of the next neuron, which realizes the whole model which can process sequential data and make judgments and classifications according to the data context. For the training of the neural network, we use BP algorithm. The RNN is more suitable for the sequence data than the basic neural network and the Statistical learning method because of its structure. In this paper, we use the classic model LSTM [Lu, Shi, Jia et al. (2016)].

3 Authentication model and system framework

The identity authentication model based on smartphone user behavior mainly includes three parts: data acquisition part, behavior analysis part, and behavior modeling matching part. It is specifically shown in Fig. 1.

The data acquisition model includes data acquisition and data preprocessing. Data acquisition mainly collects data from a mobile phone linear acceleration sensor, an angular acceleration sensor, and from user touch screen data, which needs to get ROOT permission. The data preprocessing mainly extracts the features of the sensors' original data in the time domain and the frequency domain using the statistical method, and decodes the touch screen data.

The behavior analysis part obtains the feature vector by extracting the feature values from the preprocessed data, in order to train the classification model as a data input for user identity authentication.

The behavior modeling matching part constructs the model by the sensor data and the touch screen data through the machine learning SVM model and the neural network RNN model respectively, who obtains the final authentication result by weighting and summing them up.

On the mobile client, data is collected, preprocessed, and sent to the server for identity training and verification, where the results will then be sent to the mobile client for appropriate operations. The background monitor continuously captures user behavior data, and it is completely transparent to the smartphone users, who do not need any additional operations. This whole system is very friendly to users.

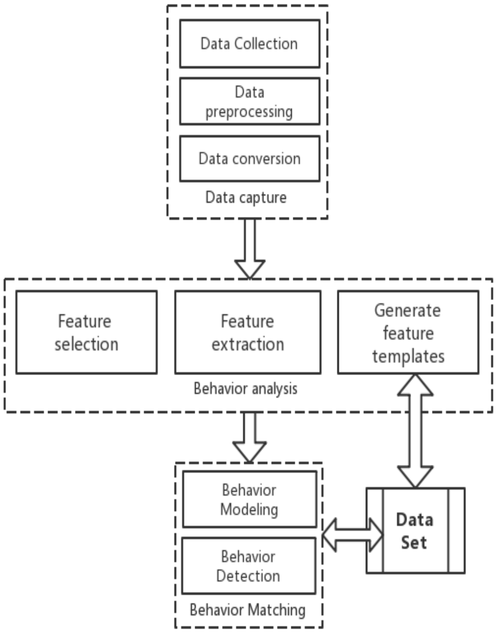

Figure 1: Identity authentication model framework based on smartphone user behavior

4 User behavior definitions and processing

4.1 User behavior data capture

In order to effectively describe the user's behavior and extract valid features, we divide the user's behavior data into two parts, corresponding to sensor data and touch screen data. In order to make the capture program stay in memory for a long time and not be cleaned up by anti-virus software, we use dual Service keep-alive technology to start two different priority services, to capture sensor data and touch screen data respectively, and to monitor whether each is alive at the same time.

4.1.1 User holding posture behavior

Android provides a sensor framework for sensor interaction and sensor data acquisition. By using SensorManger to encapsulate the HAL layer for different sensor implementations, the management of different sensors can also be decoupled. SensorManger can proxy the interface of the sensor and send the data to your registered sensor event listener by making a request to SensorManger.

We set the sensor sampling frequency to 50 Hz. Because the user's behavior is relatively small in the case of holding the handshakes and the sampling rate is too low to lose information. Therefore, we choose to collect the information at the sampling frequency of 50 Hz. The buffering time is our time window size.

By synthesizing the vectors of the three orthogonal directions of the sensor, the angle and the mode length between the final vector and the horizontal plane of the mobile phone can be calculated.

4.1.2 User touch screen operation

Root permission is required to obtain the system touch screen data, the screen interaction data can be obtained through "getevent-l", and all the types of raw data can be captured as follows:

- EV_KEY BTN_TOUCH DOWM/UP
    - Press or lift the physical button of the mobile phone. As this paper mainly examines the user's touch screen gestures and holding posture this type of data is ignored.
- EV_ABS ABS_MT_POSITION_X
    - EB_ABS ABS_MT_POSITION_Y
    - The absolute value of the horizontal and vertical coordinates of the user's touch screen.
- EV_ABS ABS_MT_PRESSURE
    - It mainly records the amount of pressure when the user touches the screen.

While collecting touch screen data, we found that all touch screen behaviors can be differentiated into a sequence of basic touch screen events. Below is a description of the basic touch screen events.

Touch screen movement event (m): The touch screen moves from one point to another to trigger the event.

Touch screen press event (down): Touch the screen to trigger the event.

Touch screen bounce event (up): Trigger the end of the screen to trigger the event.

Moving event threshold t_{mm}: The maximum time interval between two moving events to form a moving sequence.

Touch screen sliding behavior (SDD): Refers to the start of the touch screen, after a period of movement, the end of the event with a touch screen.

$SDD =< down_{t_1}, [m_{t_2}, m_{t_3}, \cdots, m_{t_{n-1}}], up_{t_n} >$

Touch screen movement sequence (SMS): Refers to an entire sequence of touch screen movement events, and the time interval between two chronological touch-screen sliding behaviors is less than t_{mm}.

$$SMS_{t_1}^{t_n} =< m_{t_1}, m_{t_2}, \cdots, m_{t_n} \mid \forall k: 1 \le k \le n-1, t_{k+1} - t_k \le t_{mm} >$$

Through the definition, we can extract touch events completely and independently.

4.2 User behavior feature and its calculation method

4.2.1 User holding posture behavior

After obtaining the raw data of the sensor, the statistical analysis is used to extract the features of the time domain and the frequency domain. And the time domain features include mean, variance, maximum, minimum, median, quartile, number of intersections with 0, root mean square, kurtosis, skewness, etc.

The analysis of frequency domain features is generally based on the fast algorithm of discrete Fourier transform, which includes frequency, signal energy, entropy, etc.

After analyzing and comparing the differences of multiple features, we chose the maximum, minimum, average, variance, frequency domain peak, second highest peak, and peak frequency band of the time domain as our feature vector, as shown in Eq. (4).

$$F = \{Mean, \mathrm{var}, Max, Min, Ran, Peak, Peak\_f, Peak2, Peak2\_f\} \tag{4}$$

The feature vector is described as in Tab. 1.

Table 1: Sensor feature vector meaning

Feature names	Feature meanings
Mean	Average of sensor flow data
Var	Sensor variance
Max	Maximum sensor data
Min	Minimum value of sensor data
Ran	Range of sensor data
Peak	Main frequency amplitude
Peak_f	Main frequency of data stream
Peak2	Offset amplitude
Peak2_f	The size of the offset

4.2.2 Feature extraction of touch screen data

After getting the native data and decoding operations, we used mathematical calculations to get the statistic of users' single actions. Merge the single data which is less than t_I in the time cell, roughly estimate each action's time cell of the user, and also calculate the average pressure. Take the first operation's coordinate in the time cell as the start coordinate and the last coordinate as the end coordinate, so as to get the feature of the touch screen.

In Fig. 2, there are some dispersed points between (X_1, Y_1) and (X_1, Y_1), For these dispersed points, we define the two points whose time difference is less than t_{mm} as one action, and so on when the time difference of two points is larger than t_{mm}, we take the second point as the start of the next action. The t_{mm} of different tasks are different.

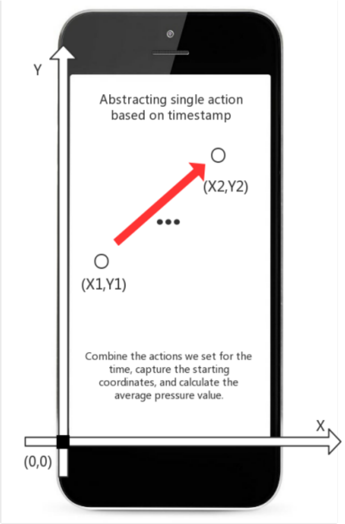
Y
Abstracting single action based on timestamp
(X2,Y2)
•••
(X1,Y1)
Combine the actions we set for the time, capture the starting coordinates, and calculate the average pressure value.
X
(0,0)

Figure 2: Extract touch screen features

After recording the start coordinate, we also count the number of the points in the time cell. It is noteworthy to mention that we merge all the points in unit time as one action. In other words, one action may have several unit times. In summary, the feature vector of touch screen data is as shown below in the Eq. (5).

$$F = \{x_{average}, y_{average}, s, \theta, c, P_{average}\} \tag{5}$$

The feature vector meaning of touch screen is described in Tab. 2.

Table 2: Touch screen feature vector meaning

Feature name	Feature meaning
$x_{average}$, $y_{average}$	Coordinate average
S	Moving distance from the start point to the end
θ	Moving angle: $\theta_i = \arctan^*(\frac{\delta y_1}{\delta x_1}) + \sum_{j=1}^{i} \delta\theta_i, \delta\theta_i = \delta \arctan^*(\frac{\delta y_i}{\delta x_i})$
c	Curvature $c = \delta\theta / \delta s$
$P_{average}$	Average pressure throughout the process

5 Training and authentication

Arrange the touch screen data in chronological order, put every Timestep data as an input vector into LSTM for training. We hope the model not only learns the features of users' single actions, but also combines the users' action context in order to make the classification more accurate. The specific process is shown below.

I) The data size of input to server model is Timesteps*6, Timesteps is the time cell, the model uses Timestep's actions to judge whether the user is legal, 6 is the dimension of the vector in Eq. (2). Then it trains the LSTM model.

II) Put the input data into LSTM to calculate by turn, every neuron will transmit a vector h whose dimension is 6 to itself, the h is also the output vector. By the feedback, LSTM neurons update its legacy information C, then, the output vector goes to full connection layer.

III) After calculating Timestep input vectors, the LSTM transmit a vector whose size is Timesteps*6 to the full connection layer, after the calculation of the full connection layer, the vector will be extended to Timesteps*6*2. The dropout layer disconnects the connection between two full connection layers into certain probabilities, in order to increase the complexity of the model in case of the overfitting.

IV) After the dropout layer, the vector goes into the next full connection layer, in this layer; vector is converted into two values by the sigmoid function, which represents the result of classification. We transform the question into a two-point problem in order to achieve the goal of identity authentication.

6 Experiment analyses

The data collection of smartphone users is carried out in the daily life of users. Each user installs client software that can continuously monitor the booting and self-starting on their respective mobile phones, and automatically sends the data to the server at specific time intervals.

This paper collected data on the behavioral habits of 10 smartphone users, all using a mobile phone with a screen resolution of 1920*1080 (FHD) and an operating system of Android. The collected data is a series of user actions, system times (accurate to milliseconds), current active process information, and so on.

For a mobile phone user-based behavior authentication method, the most important performance parameter is the false rejection rate (FRR), which treats legitimate users as illegal users and rejects their access to the mobile phone, and the error acceptance rate. And FAR (False Accept Rate) is a resource that an illegal user is regarded as a legitimate user and authorizes an illegal user to access the mobile phone. This paper analyzes the proposed identity authentication method by using these two performance parameters and Accuracy (average of FRR and FAR).

We need to determine the data size, when we are training the data; here we specify the window size as 6 seconds. Another important parameter in the system is the volume of the data set, which is quite important for model training and authentication, because a large data set will provide too much content to the model. The relationship between data set size and identity authentication accuracy is shown in Fig. 3 below:

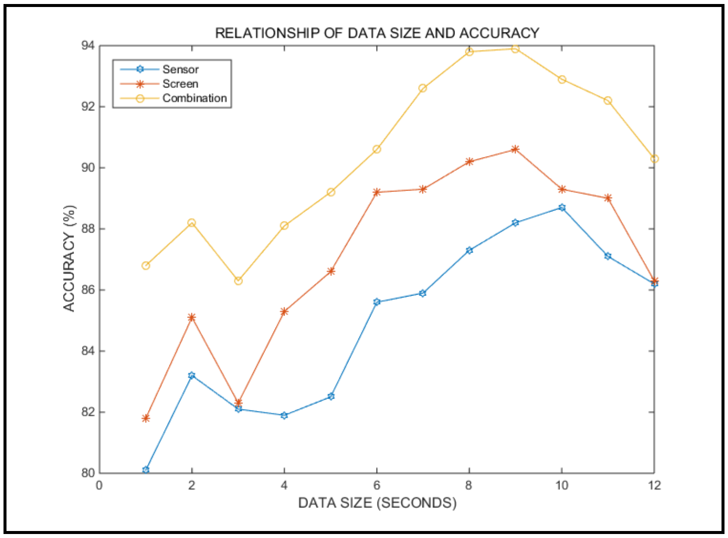

Figure 3: Training data set size and identity authentication accuracy

The statistical result of Combination is obtained by combining Sensor and Screen, and the probability of access restriction is set by weighted summation. From the above figure, we can clearly see that the data set size reaches 900 at the peak of Accuracy, and the accuracy will decrease when the data size continues to increase. This is because an oversized data set will provide over-fitting to our model. The algorithm causes the results of the model to appear with more errors than we expected, so we choose 900 for our dataset size.

At the same time, we get these statistics for each indicator when the data set size is 900 seconds and the window time is 6 seconds, shown in Tab. 3.

Table 3: The value of each indicator when the data set size is 900

Device	FRR	FAR	Accuracy
Sensor	13.4%	15.4%	85.6%
Screen	10.6%	9.2%	90.1%
Combination	5.2%	7%	93.9%

From the experimental results, we can see that the value of FAR is 7% now. For the length of the authentication response time window, we set the response time to n window lengths. Then the probability that the attacker wants to invade during this period p^n, p is the value of FAR.

The FAR value of the current model is 7%. Here we set n to 4, that is, the response time is 24 seconds. So then, the probability of a successful attack by an attacker is 0.002%, which virtually ensures it will not be invaded.

7 Conclusions

This paper proposes an identity authentication method based on mobile phone user behavior. The entire authentication process is transparent to users and the experiment results prove to be sound. Firstly, we give the definition of user behavior and divide it into sensor data and touch screen data, and give the eigenvalues and their calculation methods. In the data authentication phase, we authenticate users by combining the weighted results of the SVM algorithm and the RNN neural network algorithm. Finally, the experimental analysis of the proposed method shows that the method has better FRR and FAR.

Acknowledgement: This work is supported, in part, by the Natural Science Foundation of Liaoning Province of China under grant (No. 20180550629), in part, by the Postdoctoral Research Initiation Fund of Shenyang University.

References

Ahmed, A. A. E.; Traore, I. (2005): Detecting computer intrusions using behavioral biometrics. *3rd Annual Conference on Privacy, Security and Trust*, pp. 91-98.

Ge, S. M.; Zhao, S. W.; Li, C. Y.; Li, J. (2019): Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 2051-2062.

Ghaffari, M.; Ghadiri, N.; Manshaei, M. H.; Lahijani, M. S. (2017): P^4QS: a peer-to-peer privacy preserving query service for location-based mobile applications. *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9458-9469.

Lai, S. X.; Jin, L. W.; Yang, W. X. (2017): Online signature verification using recurrent neural network and length-normalized path signature descriptor. *International Conference on Document Analysis and Recognition*, pp. 1-9.

Lee, W. H.; Lee, R. B. (2017): Implicit smartphone user authentication with sensors and contextual machine learning. *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 297-308.

Lu, Y. J.; Shi, Y. H.; Jia, G. M.; Yang, J. F. (2016): A new method for semantic consistency verification of aviation radiotelephony communication based on LSTM-RNN. *IEEE International Conference on Digital Signal Processing*, pp. 422-426.

Nickel, C.; Wirtl, T.; Busch, C. (2012): Authentication of smartphone users based on the way they walk using KNN algorithm. *8th International Conference on Intelligent Information Hiding and Multimedia Signal*, pp. 16-20.

Prasad, R. B. (2017): Protection of privacy in big data using sdd framework with DNN. *2nd International Conference for Convergence in Technology*, pp. 998-1001.

Song, Y. P.; Cai, Z. M.; Zhang, Z. L. (2017): Multi-touch authentication using hand geometry and behavioral information. *IEEE Symposium on Security and Privacy*, pp. 357-372.

Sucasas, V.; Mantas, G.; Althunibat, S.; Oliveira, L.; Antonopoulos, A. et al. (2018): A privacy-enhanced oauth 2.0 based protocol for smart city mobile applications.

Computers & Security, vol. 74, pp. 258-274.

Wang, Y.; Hu, J. K. (2011): Global ridge orientation modeling for partial fingerprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 77-87.

Xu, J.; Jiang, Z. H.; Wang, A. D.; Wang, C.; Zhou, F. C. (2018): Dynamic proofs of retrievability based on partitioning-based square root oblivious ram. *Computers, Materials & Continua*, vol. 57, no. 3, pp. 589-602.

Xu, J.; Wei, L. W.; Zhang, Y.; Wang, A. D.; Zhou, F. C. et al. (2018): Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures. *Journal of Network and Computer Applications*, vol. 107, pp. 113-124.

Yang, L.; Guo, Y.; Ding, X.; Han, J.; Liu, Y. et al. (2015): Unlocking smart phone through handwaving biometrics. *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1044-1055.