Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 147 (2019) 314–318

Procedia Computer Science

www.elsevier.com/locate/procedia

2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018

Continuous authentication by free-text keystroke based on CNN plus RNN

Lu Xiaofeng[a], Zhang Shengfei[a], Yi Shengwei[b]

[a]*Beijing University of Post and Telecommunications, No.10, Xi Tucheng Road, Beijing100876, China*
[b]*Information Technology Security Evaluation Center, No. 8, Shangdi West Road, Beijing100085, China*

Abstract

Personal keystroke mode is difficult to imitate and can therefore be used for identity authentication. According to the keystroke data when a person inputs free text, the keystroke habit of the person can be learned. Detecting a user's keystroke habits as the user enters text can continuously verify the user's identity without affecting user input. This paper proposes to divide the user keystroke data into a fixed-length keystroke sequence, and convert the keystroke sequence into a keystroke vector sequence according to the time feature of the keystroke. A model of a recursive neural network plus a convolutional neural network is used to learn a sequence of individual keystroke vectors to obtain individual keystroke features for identity authentication. The model was tested using an open data set and the best False Rejection Rate(FRR) was 1.95%, False Acceptance Rate (FAR) was 4.12% and Equal Error Rate(EER) was 3.04%.

Keywords: Authentication;Keystroke dynamics;Free-text;CNN;RNN

1. Introduction

The user authentication for computer systems can be seen everywhere in daily life. Most systems utilize traditional one-time authentication method, such as passwords, fingerprint and facial recognition based on biometrics. However, these methods have some shortcomings. For example, passwords may be cracked or forgotten by users. Fingerprint and facial recognition require specific hardware devices. The keystroke dynamics recognition is a biological behavior feature recognition, which can achieve the purpose of authentication by analyzing individual

10.1016/j.procs.2019.01.270

typing rhythm on the keyboard [1]. The typing rhythm strongly depends on the individual's typing habits and the experience of keyboard operation. It has a certain degree of stability and will not change rapidly in a short term [2]. According to the temporal characteristics of individual keystroke, the individual typing rhythm can be abstracted. Detecting a user's keystroke habits as the user enters text can continuously verify the user's identity without affecting user input.

Most of the studies on keystroke dynamics authentication are a one-time authentication mechanism based on fixed texts (such as login passwords), focusing on the processing of time characteristics and the selection and improvement of algorithms [3]-[5]. one-time authentication mechanism obtains a partial input mode by inputting a fixed text. It is difficult to represent individual keystroke rhythm and can't be applied to a new fixed text. At present, some researchers begin to study the continuous authentication mechanism based on free-text. Comprehensive personal keystroke behavior patterns can be obtained by learning a large amount of personal keystrokes. When a user uses a keyboard to input words, the background software can authenticate the user continuously [6].

This paper uses a model of a Convolutional Neural Network (CNN) plus a Recurrent Neural Network (RNN) to learn free-text keystroke data. The model systematically studies the free-text dataset and obtains a personally unique keystroke input mode. The neural network can automatically extract features for learning. We design the corresponding CNN+RNN model according to the characteristics of keystroking data. The trained model can fully reflect the individual keystroke behavior pattern.

2. Related work

Yan Sun et al [7] collected the Buffalo keystroke dataset, and used the Gaussian mixture model clustering method. The EER of their experiment was 0.01%. Esra Vural et al. obtained a new dataset [8] and used the Gunetti & Picardi algorithm [9] to determine whether the two samples belong to the same person. The certification testing on the dataset got an optimum FRR of 3.93% and a FAR of 0.75%. Murphy et al. collected one of the largest free-text keystroke datasets [10], which recorded the data of all keyboard operations, mouse operations, and software programs during 2.5 years. They used the Gunetti & Picardi algorithm on the dataset and obtained the best EER of 10.36%.

Jiaju Huang et al. proposed a kernel density estimation(KDE) algorithm[11] to calculate the distance between the training samples and the test samples through the probability density between the training dataset and the test dataset, and then to determine the identity authenticity of the person. The algorithm was tested on various published datasets and received 1.95% EER. Tomer Shimshon et al. [12] studied the effect of the window length for continuous verification and received 3.47% FAR. Ahmed and Issa Traore[13] established a multi-layer perceptron (MLP) to automatically learn the feature combinations of keystrokes and learn individual keystroke behavior models. In their experiment, EER reached 2.46%.

3. Continuous authentication method based on free-text

3.1. Continuous authentication

Personal identity authentication is actually a binary classification. When a keystroke sequence successfully matches a user's input pattern, it indicates that the user is real, otherwise he/she is an intruder or counterfeiter. Continuous authentication checks user's behavior over a period of time without disturbing the user. It has a sliding window is shown in Fig. 1.

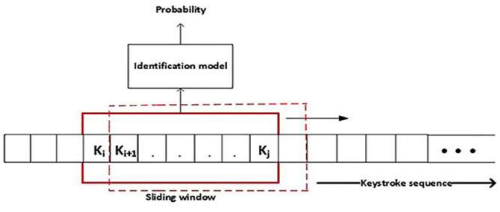


Fig. 1. sliding window authentication.

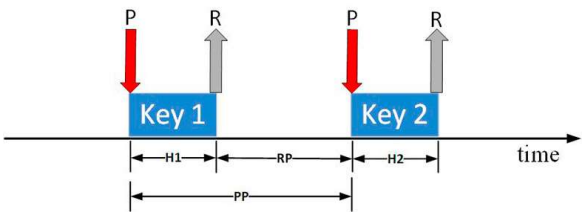


Fig. 2. keystroke time characteristics.

K_i represents single keystroke information in a keystroke sequence. The processed keystroke data in the sliding window is input into the trained authentication. The model outputs the probability of being recognized as a real user. When the user newly enters a key, the sliding window moves backward by one step and the model produces a new output. As the window slides with the input words, the authentication system can keep checking, achieving the purpose of continuous identity authentication. Hence, it is necessary to establish an independent neural network for each user and obtain their unique keystroke modes. The personal authentication model needs to be trained in advance, and the model is directly used for identification when authenticating the user. In our experiment, each authentication model occupies 301KB, and a model prediction takes 0.4ms.

3.2. Record keystroke information and indicate keystroke feature

The original data is formed by those recorded time of tapping or releasing keys. The time characteristics of the key in current studies are shown in Fig. 2. ***H***(the hold time of the key) indicates the delay time between the keys; ***RP*** indicates the interval between the release of the first key and the tap of the second key; ***PP*** indicates the interval between the press of the first key and the second key. ***P*** is the timestamp of a tapped key, and ***R*** is the timestamp of a released key.

3.3. Learning personal keystroke model by CNN+RNN

The traditional authentication model calculates means, variance and other mathematical characteristics of ***RP*** between various keystrokes. Taking those results as individual keystroke characteristics, the system learns personal keystroke input patterns with conventional machine learning algorithm. However, it is difficult to verify a person's complex and varied keystroke behavior by only using the statistical characteristics of the keystroke ***RP***. When the system contains a large amount of personal keystroke data, it can only characterize partial keystroke behavior, which leads to low accuracy. The reliability and scalability of model established by above method are not sufficient.

The neural network RNN is a excellent structure that automatically learns time series. Therefore, this article applies RNN to learn the keystroke feature because the keystroke sequence is also a chronological sequence. In order to characterize the keystroke sequence better, the CNN convolution process is performed before the keystroke sequence before inputting into the RNN network. Each convolution kernel yields a higher-level keystroke signature sequence after the convolution operation. Finally, the processed feature sequences are input into the RNN network and be trained to obtain the individual keystroke behavior patterns. The characteristics of the individual's input behavior will change from time to time. The RNN network can learn the current input characteristics and retain the previous input characteristics.

3.4. Vectored keystroke sequence

This model divides the overall keystroke data into a fixed-length keystroke sequences, and convert the keystroke time data to keystroke vectors according to the time characteristics of the keystroke. Fig. 3 shows the process of converting a keystroke text sequence to a keystroke signature sequence, ***P***[*] represents the timestamp of the key tap; ***R***[*] represents the timestamp of key release, * represents any key.

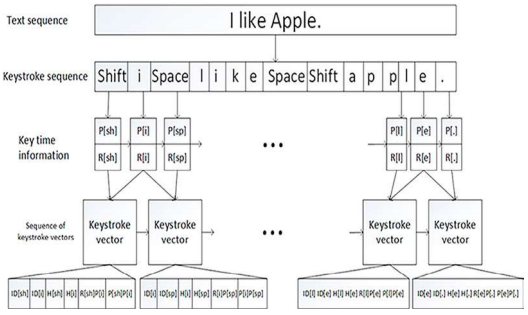


Fig. 3. keystroke sequence vectorization process.

The specific data format of the keystroke vector is as shown in Fig. 4, in which "1" represents the first key and "2" represents the second key. ***ID*** represents key label; ***H*** represents duration of the key; ***R***[1]***P***[2] represents the **RP** characteristics of the two keys; ***P***[1]***P***[2] represents the ***PP*** characteristics of the two keys,

ID[1]	ID[2]	H[1]	H[2]	R[1]P[2]	P[1]P[2]

Fig. 4. keystroke vector specific data format

3.5. Design of CNN+RNN model

The GRU network is a particular RNN that effectively solves the problem of "long-term dependence." Fig. 5 indicates our CNN+GRU model. The grid in the same color in Fig. 5 is obtained with the same convolution kernel. Different convolution kernels extract different feature sequences and then perform Pooling operation to obtain a wider range of correlation features. In the end, the pooled feature sequence is input into a double-layer GRU network for training. Besides, a dropout layer adds between each layer in the model in order to reduce overfitting risk.

4. Experiment and result

As the model's dropout = 0.5, the user's keystroke data is divided into fixed-length keystroke sequences in order to facilitate network processing. In this paper, the length of L=(10,30,50,70,100) is used to study the most optimizable length of the model. At the same time, it sets a comparison experiment: a model in which data has not been processed by CNN before inputting GRU network.

4.1. Dataset

The Buffalo dataset is collected by researchers of SUNY Buffalo. This dataset contains 157 participants' long fixed text and free text keystroke data, and all those participants can use the keyboard skillfully. With the input of fixed text and open-ended questions, participants' keystroke data has been collected. The participants completed inputting through 3 sessions, and each participant has an average of 5,700 keystrokes in each session. The average of total 3 sessions have exceeded 17,000 keystrokes.

4.2. Experimental Results

Performing experiments on all the 75 users, keystroke sequences of different lengths are input into the model for training and testing. Test results are shown in Table 1. When the sequence length is 10, the keystroke sequence contains too little information, with which the model cannot learn the user's input pattern completely and leads to high error rate and poor recognition. The sequence length continues to increase, the noise data in the sequence will increase as well as FAR , FRR and EER. When the fixed keystroke sequence length obtained by the experimental results in this paper is 30, the FRR and EER are the lowest in the experiment.

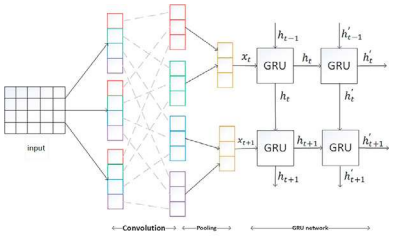
Table 1. Different sequence lengths.

sequence length	FRR(%)	FAR(%)	EER(%)
10	16.02	3.48	9.75
30	1.95	4.12	3.04
50	2.26	5.64	3.95
70	1.9	5.91	3.95
100	2.67	7.57	5.12

Table 2. Different feature combinations.

Keystroke features	FRR(%)	FAR(%)	EER(%)
H	12.39	5.96	9.17
RP	4.20	7.68	5.94
RP+PP	2.05	4.55	3.30
H+RP+PP	1.95	4.12	3.04

Fig. 5 Framework of CNN+GRU model.



If the model is trained by keystroke vectors composed of different combinations of keystroke features and set

fixed sequence length to 30, the test results are shown in Table 2. The model's recognition effect is poor if only using the duration feature H. If the interval time feature RP is used the error rate of the model decreases. If the interval time features RP and PP are combined to form a keystroke vector, the error rate is further reduced. Finally, the error rates of the model reach the lowest level when three keystroke features: H, RP, and PP were used together.

This paper designs a model comparison experiment in which CNN is used or not in keystroke sequence. The model without CNN yielded FRR = 4.05%, FAR = 6.01%, and EER = 5.03%. The experimental with CNN has a convolution kernel length of 2, a number of 32, and a pooling length of 2. The RNN+CNN model yielded FRR=1.95%, FAR=4.12%, and EER=3.04%. It can be seen that using CNN processing will lower the error rate of model. So adding CNN related layers will improve the model's overall effect.

5. Conclusion

This paper employs the CNN+RNN model to learn the keystroke data of free texts. The model obtains a more complete personal keystroke input mode to carry on continuous authentication. The keystroke data is vectorized according to keystroke time feature combinations and then divided into fixed-length keystroke feature sequences in order to enable keystroke sequences to input into the RNN networks. In order to improve the performance of the network model, the keystroke sequence is firstly processed by CNN to extract a higher-level keystroke feature sequence. Then the keystroke feature sequence inputs into the RNN. After training, the model obtained the best experimental results: FRR = 1.95%, FAR = 4.12%, EER = 3.04%. It can achieve the best identity recognition effect with a sequence length of 30, and it achieves good practicality.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61472046) and the Information Network Security Key Laboratory of the Ministry of Public Security Open Project (C17607).

References

[1] F.Bergadano, D.Gunetti, and C.Picardi. (2002) " User authentication through keystroke dynamics." *ACM Trans. Security* **5(4)**: 367–397.

[2] Rybnik, and Mariusz. (2013) "An exploration of keystroke dynamics authentication using non-fixed text of various length." *Biometrics and Kansei Engineering (ICBAKE)*: 245-250.

[3] Alshanketi, Faisal, Issa Traore, and Ahmed Awad Ahmed. (2016) "Improving performance and usability in mobile keystroke dynamic biometric authentication. " *2016 IEEE Security and Privacy Workshops (SPW)*: 66-73.

[4] Ali, Md Liakat, and Tappert C C. (2016) "Keystroke Biometric User Verification Using Hidden Markov Model." *The 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*: 204-209.

[5] Yadav, Jatin, and Gupta S. (2017) "Keystroke dynamics based authentication using fuzzy logic." *The tenth International Conference on Contemporary Computing (IC3)*: 1-6.

[6] P.Bours. (2012) "Continuous keystroke dynamics: A different perspective towards biometric evaluation." *Information Security Technical Report***17(12)**: 36–43.

[7] Sun, Yan, Hayreddin Ceker, and Shambhu Upadhyaya. (2016) "Shared keystroke dataset for continuous authentication." *IEEE International Workshop on information Forensics and Security (WIFS)*: 1-6.

[8] Vural, Esra, and Hou D. (2014) "Shared research dataset to support development of keystroke authentication." *IEEE International Joint Conference on Biometrics (IJCB)*: 1-8.

[9] D. Gunetti, C. Picardi. (2005) "Keystroke analysis of free text." *ACM Trans. Inf. Syst. Secur***8(3)**: 312–347.

[10] Murphy, Christopher, and Hou D. (2017) "Shared dataset on natural human-computer interaction to support continuous authentication research." *IEEE International Joint Conference on Biometrics (IJCB)*: 525-530.

[11] Huang, Jiaju, and Schuckers. (2017) "Benchmarking keystroke authentication algorithms." *Information Forensics and Security (WIFS)*: 1-6.

[12] Shimshon, Tomer, and Rokach L. (2010) "Continuous verification using keystroke dynamics." *Computational Intelligence and Security (CIS):* 411-415.

[13] A. A. Ahmed and I. Traore. (2014) "Biometric recognition based on free-text keystroke dynamics." *IEEE Transactions on Cybernetics***44(4)**: 458–472.

[14] Çeker, Hayreddin, and Shambhu Upadhyaya. (2016) "User authentication with keystroke dynamics in long-text data." *The 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*: 1-6.