Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2018 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

6-26-2018

Keystroke Biometrics for Freely Typed Text Based on CNN model

Kun Lv
School of Economic Information Engineering, Southwestern University of Finance and Economics, 1172977438@qq.com

Jiafen Liu
School of Economic Information Engineering, Southwestern University of Finance and Economics, jfliu@swufe.edu.cn

Ping Tang
School of Economic Information Engineering, Southwestern University of Finance and Economics, 740758547@qq.com

Qing Li
School of Economic Information Engineering, Southwestern University of Finance and Economics, liq_t@swufe.edu.cn

Keystroke Biometrics for Freely Typed Text Based on CNN model

Completed Research Paper

Kun Lv
School of Economic Information Engineering, Southwestern University of Finance and Economics
555 Liutai Ave., Chengdu, 611130, China
lkklein@163.com

Jiafen Liu(✉)
The Key Laboratory of Financial Intelligence and Financial Engineering of Sichuan Province
School of Economic Information Engineering, Southwestern University of Finance and Economics
555 Liutai Ave., Chengdu, 611130, China
jfliu@swufe.edu.cn

Ping Tang
School of Economic Information Engineering, Southwestern University of Finance and Economics
555 Liutai Ave., Chengdu, 611130, China
740758547@qq.com

Qing Li
The Key Laboratory of Financial Intelligence and Financial Engineering of Sichuan Province
School of Economic Information Engineering, Southwestern University of Finance and Economics
555 Liutai Ave., Chengdu, 611130, China
Liq_t@swufe.edu.cn

Abstract

Keystroke biometrics, as an authentication method with advantages of no extra hardware cost, easy-to-integrate and high-security, has attracted much attention in user authentication. However, a mass of researches on keystroke biometrics have focused on the fixed-text analysis, while only a few took free-text analysis into consideration. And in the field of free-text analysis, most researchers usually devote their efforts to extracting the most appropriate keystroke features on their own experience. These methods were inevitably questionable due to their strong subjectivity. In this paper we proposed a multi-user keystroke authentication scheme based on CNN model, which can automatically figure out the appropriate features for the model, adjust and optimize the model constantly to further enhance the performance of model. In the experiment on a small sample set, the performance is improved more than 10% compared with the benchmark. Our model achieves an average recognition accuracy of 92.58%, with FAR of 0.24% and FRR of 7.34%.

Keywords: CNN, keystroke biometrics, feature extraction, feature matrix, free-text analysis

Introduction

With the rapid development of information technology, we have entered a hyper-connected era. In this era, we connect and communicate with each other constantly via the Internet. The instant messaging softwares, such as QQ and WeChat, are further narrowing the distance between people. However, with the increase in connectivity, the information security such as personal privacy and commercial data has been challenged. Although people have invested heavily in security, those network systems are still vulnerable to security breaches, such as the famous WannaCryn ransomware attack (Ehrenfeld 2017) in 2017 that has inflicted heavy losses on the assets of individuals and enterprise.

User authentication is important for the system security, which is regarded as the first gateway of a network system. As we know, username/password scheme is the most commonly used method for user authentication. But it is not safe enough because the password can be easily forgotten or stolen by others. To overcome the inborn defect of username/password scheme, various authentication methods have been developed. Among them, biometric authentication is noticeable for its ideal recognition performance. Biometrics consists of physical characteristics and behavioral characteristics. The former refers to a person's physical attribute, such as fingerprint, iris, DNA, while the latter refers to the way people do things, including gait and signatures. In recent years, biometric authentication has been applied more and more widely, such as fingerprint unlock, face payment and so on. However, most of these physical features require extra equipment to extract, which increases the system's cost and difficulty of operation.

Contrary to physical characteristics, the keystroke dynamics in behavioral characteristics break through those limitations, which authenticate users by measuring and assessing user's typing rhythm on digital devices such as mobile phone or computer keyboard. The typing features can be directly accessed when users input their personal password without extra equipment, making it extremely easy to obtain and can be integrated into the traditional password-only authentication system to strengthen its security. Nowadays, keystroke dynamics has attracted many researchers to research in this field, owing to the advantages including low-cost, easy-to-integrate and high-security (Vizer et al. 2009).

As early as 1975, scientists noticed that every typist had the unique tapping rhythm (Spillane 1975), and the studies of keystroke authentication had been carried out in the 1980s. At present, the keystroke recognition research can be classified to fixed-text analysis and free-text analysis. The former refers to the analysis of keystroke data of fixed-length strings, such as user IDs, passwords, or pre-defined strings. Prior literatures suggest that most researchers tended to study the fixed-text analysis, resulting in producing the most abundant results (Karnan et al. 2011). Hence, there are lots of research methods in this area, such as statistical methods (Magalh ães et al. 2005), neural networks (Cho et al. 2000), support vector machines (Yu & Cho 2003), and fuzzy logic (Mandujano & Soto 2004). Much progress has been achieved and some models even meet the commercial requirements. However, it is a pity that authentication on fixed-text analysis has an inherent shortcoming: when the user accomplish the login phrase, the system will not detect whether the current user is a valid user constantly. Hence, keystroke authentication method based on freely typed text analysis has been developed. However, the literatures on free-text analysis are quite few and mainly focus on the extraction of suitable features and the establishment of an effective authentication model (Alsultan & Warwick 2013). It is more difficult to analyze the keystroke features of free-text than that of fixed-text. In terms of modeling, it's easy to convert the keystroke of short and pre-defined text to a fixed-length feature vector, whereas long and freely typed text is relatively more difficult because we have no idea what keys the user entered or even how many keys are entered before the user finishes typing. Due to the difficulty of feature extraction in free-text analysis, it is hard to apply machine learning models to free-text analysis.

In this paper, in order to reduce the difficulty and subjectivity in feature extraction for free-text analysis, we propose a multi-user keystroke authentication method based on convolutional neural network (CNN), inspired by the idea of representation learning and face recognition. Distinguish from previous studies, our method does not require much work on feature extraction, and it can get good performance by just focusing on the model. Our method transforms the conventional monographs and four types of digraphs into two types of feature matrices, and act as the inputs of CNN model, similar to the architecture of the AlexNet (Krizhevsky et al. 2012), which uses group convolution to operate the features of monograph and digraph respectively. This will not only accelerates the training process, but also extracts the deep features of monograph and digraph. Finally, the results show that the average FAR and FRR reduce to 0.24% and 7.34% respectively. Meanwhile, with the highest recognition accuracy of 96.09% and an average recognition accuracy of 92.58%.

The remainder of the paper is organized as follows. Section 2 reviews the related works for freely typed text. And Section 3 introduces the architecture of the entire keystroke authentication method, including the establishment of feature matrix and the architecture of CNN. Section 4 presents the data and experimental design, and Section 5 shows our experimental results and analysis. We summarizes our approach and the experimental results, as well as the outline of future work in Section 6.

Related Works

Most previous works focus on fixed-text analysis to strengthen authentication systems, while a few concern about the long and freely typed text. Although it is roughly difficult to study, the free-text analysis still has more obvious advantages than fixed-text analysis in security authentication. As we mentioned above, these a few literatures related to free-text analysis can be divided into two groups, namely the extraction of suitable features and the establishment of an effective authentication model. In this section, we will introduce some representative works related to each group. For more general reviews on keystroke authentication systems, ranging from feature extraction to authentication models, please refer to Alsultan and Warwick (2013).

Feature Extraction for Freely Typed Text

Almost all keystroke features are mainly base on the duration time from pressing/releasing one key (start key) to pressing/releasing another key (end key). According to the number of keys depressed in the duration, these time features are called digraph, tri-graph, four-graph, etc. These are collectively referred to as n-graph. In particular, if the start key and end key are the same one, this feature is called monograph. In the previous free-text analysis, the feature is mainly formed by the average of this duration, and some creative processing is performed. To the best of our knowledge, Monrose and Rubin (1997) are the earliest researchers setting foot in this field, they selected the average press-press time of the top N most frequent digraph and tri-graph as a feature of each user, which is also the most commonly used method. However, the most classic approach comes from Gunetti and Picardi (2005), who extracted features from the average typing speed of the relative (R-measure) and absolute (A-measure) for all the common n-graphs in any two keystroke sequences. In this way, deeper keystroke information are captured. After that, other researchers also tried to improve authentication performance by using features derived from the basic R+A method (Hu et al. 2008; Messerman et al. 2011). Regrettably, R+A method only considers common syllables between two keystroke sets, losing too much information about the other keystrokes. To make up for this defect, Filho and Freire (2006) proposed a method by using press-press time of all keystrokes to construct a histogram, which was used by authentication model to calculate the overlap of the two keystroke datasets. The more the overlap, the greater the probability of that those two datasets were typed by the same person. There is no doubt that they would lose the information of keystroke sequence, but they kept all the keystrokes and got more information. In addition, there are some researchers who introduce keyboard grouping technique and extract the average keystroke duration from one block to another block. Park et al (2010) divided all keyboard into four disjoint attributes: left hand side keys, right hand side keys, spacebar and backspace bar; then, creating 16 digraph using these attribute combinations. Later, Singh and Arya (2011) improved this technique by dividing all keyboard into 8 sections: two left and right halves and then each half divided into 4 lines representing the rows of the keyboard. For example ‘WM’ is represented as Left 2- Right 4. This kind of process not only reduces the variety of keys, but also ensures that the user's overall keystroke features are captured. Beyond above, some researchers also extracted some statistical features (Li & Liu 2016; Alsultan et al. 2017) such as the keystroke speed of Shift, usage probabilities of spacebar, average pause time, and so on.

Authentication Models for Freely Typed Text

The authentication models can be roughly divided into statistical model, the conventional machine learning model and deep learning model. Statistical model is relatively the most widely studied, including but not restricted to euclidean distance (Villani et al. 2006), manhattan distance (Bours 2012) and hypothesis testing (Park et al. 2010). These methods generally achieve good performance, and the process of authentication is similar to the method of fixed-text analysis except in the field of extracting feature vectors. Nevertheless, the machine learning model is relatively less used. For instance, Gunetti and Ruffo (1999) employed decision tree as their base classification algorithm for user authentication and eventually achieved a recognition accuracy of 90% on 10 human samples by applying the digraph latency and executed commands as features. Moreover, Hu et al. (2008) extracted features based on the R+A method and constructed authentication model by K-nearest neighbor algorithm, which not only reduced the computational complexity but also obtained 0% FRR and 0.045% FAR. In the research of Kang (2015) and Kim (2018), they both employed multiple novelty detection algorithms, targeted legitimate users for single classification modeling to simulate realistic keystroke scenarios, resulting in EERs of 5.64% and 0.44%, respectively. Further, Alsultan et al. (2017) employed decision tree and SVM as their base classification algorithms for user authentication, and regarded legitimate users as positive samples and other users as negative samples to construct a binary classification model, eventually they got 1.1% FAR and 3.75% FRR.

Though deep learning methods have been popular, there are few applications in keystroke authentication during the past years, especially on free-text analysis, due to the difficulty of fitting deep learning model with numerous parameters to a small dataset for keystrokes authentication. However, the appearance of transfer learning has solved the problem of the small datasets. Çeker and Upadhyaya (2017) applied transfer learning to conventional SVM model, by pre-training and then fine-tuning on a few new samples, which had solved the impact of keystroke authentication under different environmental conditions. Extending this stream of research, Hellström (2018) and Yunbin (2015) employed the same approach except the base models were replaced by neural network and RBM respectively, but both of them only aimed at the fixed-text analysis such as password. In addition, Hellström (2018) also reduced the dimensionality of keystroke features by using the Auto-Encoder, and got better performance based on the transfer learning. Further, Sun et al. (2017) proposed a multi-view multi-class framework for user identification on mobile devices. They divided freely keystroke datasets into three parts: alphabet, special characters and accelerometers, which exactly as the three parts of input to the neural network model with the hidden units of GRU-BRNN respectively, and the authentication model is formulated as multi-class classification. Ultimately they got a recognition accuracy of 93.5%, while the time of applying the training model to identify is only 0.657ms, achieving satisfactory performance. No wonder that deep learning can still achieve considerable recognition performance in the field of keystroke authentication.

In conclusion, prior works always need much effort in feature extraction and modeling such as adjusting the features and modifying the models constantly in attempt to get better performance. However, with the emergence of deep learning, we can fuse the feature extraction into our model, and let the model learn better representation of features by itself, thus enhance the performance of the model. With this idea, we develop a multi-user authentication method based on deep learning for free-text analysis in our study, which can automatically does feature learning.

The Architecture

Feature Matrix

As we know, feature vectors like monograph and digraph are widely used in traditional keystroke dynamics, where more complex models are rarely used (Karnan et al. 2011). The former, monograph refers to the continuous press time of a single key, and digraph refers to the time feature of two adjacent keys, which specifically contains four basic features, as shown in Figure 1 and Table 1 below. In free-text keystroke dynamics, using monograph and digraph of the key directly cannot form fixed-length feature vectors because of the unrestricted input text. Therefore, in previous studies, mostly utilized the mean and standard deviations of all time features of the same keys, the model for particular language will choose the appropriate features of monograph and digraph based on linguistic characteristic (Vizer et al. 2009; Li & Liu 2016).

In this paper, we employ all features of monograph and digraph directly, ignoring the characteristics of different languages, and all features need to be converted into a one-dimensional sequence vector or two-dimensional matrix as the input of the CNN model. However, we just convert all features of monograph and digraph into two-dimensional matrix, due to the reasons that CNN is more suitable for the classification of two-dimensional matrix such as images. Besides, the two-dimensional matrix is more capable of mining the implicit relationship among keys.

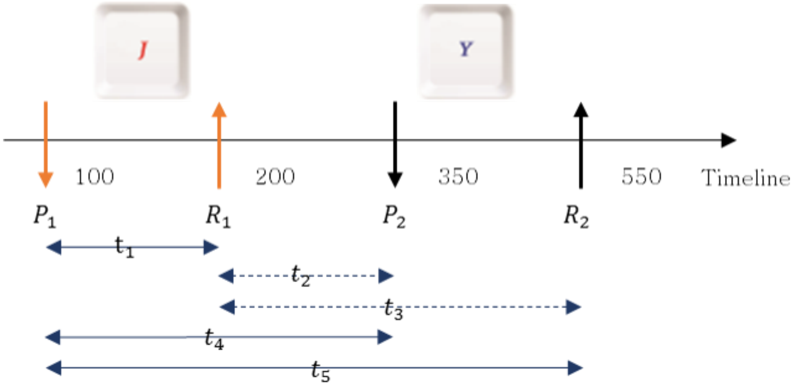
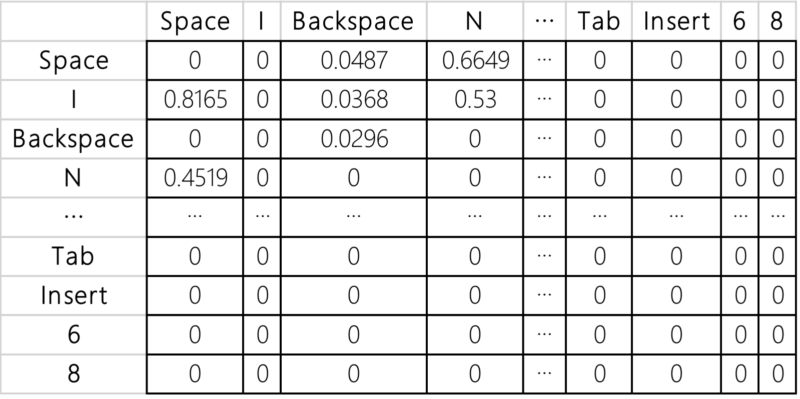


Figure 1. Different Keystroke Events of Two Characters “*J*” And “*Y*”

Table 1. Five Basic Features from Figure 1.

symbol	time from action	time to action	expression
t_1	Press J	Release J	$D = R_1 - P_1$
t_2	Release J	Press Y	$F_{type1} = P_2 - R_1$
t_3	Release J	Release Y	$F_{type2} = R_2 - R_1$
t_4	Press J	Press Y	$F_{type3} = P_2 - P_1$
t_5	Press J	Release Y	$F_{type4} = R_2 - P_1$

In particular, as shown in Table 1, we select each action of all characters in the column 2 (‘time from action’) as the row for each matrix, and action in column 3 (‘time to action’) as the column. Next, we use average duration time between two actions as the value of each matrix, in which the non-existent keystroke combination is marked as 0 in the matrix. Then we sort the whole keystroke actions by a certain order in attempt to format the feature matrix. Finally, we transform the five time features, including t_1, t_2, t_3, t_4 and t_5, into five matrices respectively. For each feature matrix, all the values are normalized so that they fall in the [0, 1] range. The normalization is performed using the MINMAX algorithm. As shown in Figure 2, it is a sample of feature matrix formed by t_2, in which a row index of a matrix represents releasing a corresponding key, a column index represents pressing a corresponding key, and the corresponding intersection exactly represents average duration time between the two actions above. For instance, the value in row 1 and column 3 is 0.0487, which is the normalized time from releasing the Spacebar to pressing the Backspace.



	Space	I	Backspace	N	···	Tab	Insert	6	8
Space	0	0	0.0487	0.6649	···	0	0	0	0
I	0.8165	0	0.0368	0.53	···	0	0	0	0
Backspace	0	0	0.0296	0	···	0	0	0	0
N	0.4519	0	0	0	···	0	0	0	0
···	···	···	···	···	···	···	···	···	···
Tab	0	0	0	0	···	0	0	0	0
Insert	0	0	0	0	···	0	0	0	0
6	0	0	0	0	···	0	0	0	0
8	0	0	0	0	···	0	0	0	0

Figure 2. The Sample of Feature Matrix Formed By t_2

Moreover, we regard the feature matrix formed by t_1 as a single channel of the input, because the matrix is formed by the monograph. Otherwise, since the feature matrices formed by t_2, t_3, t_4 and t_5

are constructed by the digraph features, we concatenate them together to format as four channels of the input. Ultimately, these two types of input matrices are input to CNN.

The Architecture

Convolution neural network (CNN) was first proposed by LeCun et al. (1998), dedicated to solving the problems such as image classification and so on. It was not widely applied until Krizhevsky et al. (2012) use AlexNet, one kind of CNN, to win the ImageNet contest in 2012. After that, CNN greatly promoted the development of deep learning. Moreover, the salient features of CNN such as weight sharing, local receptive field and subsample, are the key points to the success of CNN.

Specifically, CNN contains three structures, namely convolutional layer, pooling layer and full-connection layer. The first two types of layers are used for automatically feature extraction, and the last one is just a normal neural network.

The operation of convolution refers to the inner product of the filter and different windows of the input image data separately. In the convolutional layer, the filter convolves the local input data. After calculating the local data in a data window, the window continuously translates and slides until all the data is calculated. More importantly, although the data in the window is constantly changing, the weight of the same filter remains the same, which greatly reduces the amount of parameters and makes CNN easier to train. In addition, each filter can get a few features due to weight sharing, so there are typically multiple filters in this layer to capture more features.

On the other hand, the pooling layer is a method of data compression in CNN, which reduces the size of data by subsampling. Taking the max pooling as an example, the model preserves the largest value in each data window, which not only ensures the generalization ability of the model but also reduces the size of the output data. Besides, in the keystroke feature matrix, there are some discriminative keystroke features under different windows, such as the longest or shortest keystroke time. In fact, the operations of convolution and pooling as mentioned above can highlight these values and optimize the parameters of the corresponding filter, thereby enhance the classification performance of the model. In order to obtain more and deeper features in the CNN, there are usually many layers of convolution and pooling, eventually utilizing the normal neural network model to train and predict.

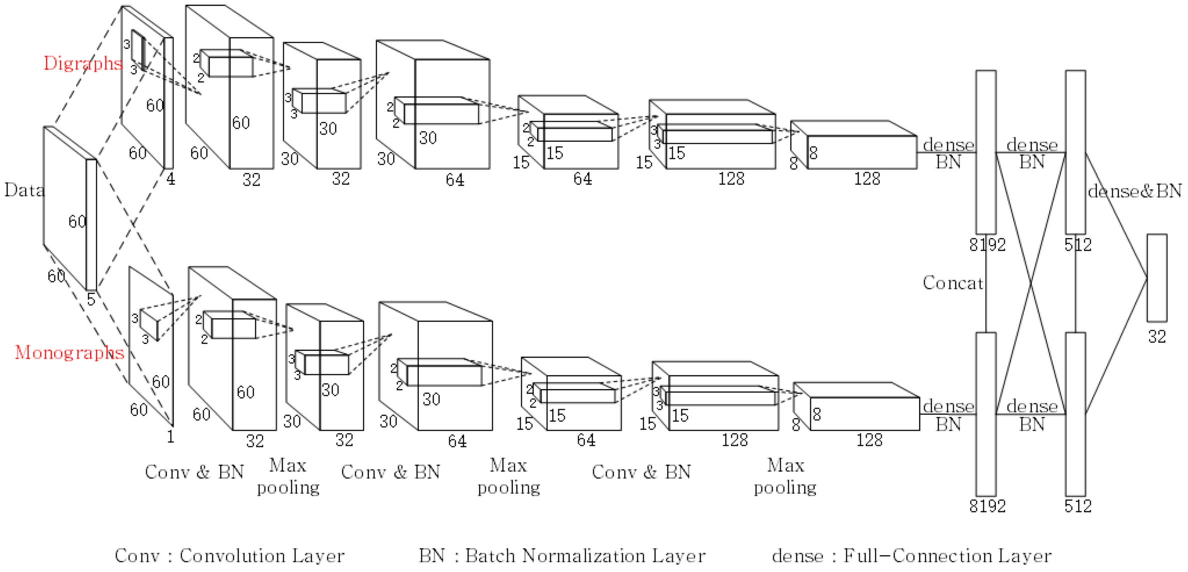


Figure 3. The Specific Architecture of Our CNN Model

In this paper, we use an architecture similar to AlexNet, due to the reasons that our input including the feature matrices of both monograph and digraph. The specific architecture consists of feature learning and authentication model. In terms of feature learning, the two sets of inputs perform two sets of convolution operations respectively to learn better features. As for the authentication model, new features will be put into the full connection layer, and the output layer outputs final classification results. The detail of architecture is shown in Figure 3 and some technical points will be explained as follows.

Group Convolution

Group convolution first appeared in AlexNet (Krizhevsky et al. 2012). However, convolution cannot be handled on only one GPU while training AlexNet, due to the limited hardware resources at that time. Therefore, Krizhevsky et al. used multiple GPUs to process multiple feature maps separately. And they eventually concatenated the results of multiple GPUs.

Being inspired by the idea of group convolution, we divide feature matrix into two categories, like monograph and digraph. These matrices will be separately operated through three-layer which are convolution and pooling, using multiple GPU parallel processing, and finally concatenate those two parts of the results as a long vector which is learned by CNN and exactly regarded as the final features. This processing not only extracts vast implicitly features of monograph and digraph in several, but also achieves parallel GPU, accelerating training speed.

Batch Normalization

It is well known that data distribution of training set and test set should be consistent with original data in machine learning. When using mini-batch for training, the data distribution of each batch should be basically the same, and that of each layer in CNN should be consistent, too. However, the distribution of data in all layers of the network is constantly changing, due to the reasons such as the change of each batch data and constant adjustment of parameters in the network. In order to adapt new data distribution, the network will adjust timely and constantly, resulting in reduction of training speed and difficulty of fitting model. This phenomenon is called internal covariate shift by Ioffe and Szegedy (2015).

Fortunately, they proposed a method named ‘Batch Normalization’ that exactly solve the defect of this data distribution inconsistency. By normalizing the input of each layer, this method can ensure that the input data distribution of each layer is stable and consistence, so as to converge quickly. In our architecture, adding Batch Normalization to each convolutional and full connection layers, we obtained an average accuracy of 80% for only in 1000 batches of training, ultimately with an increase to 90% or more.

Dropout

Dropout proposed by Hinton et al. (2012) means that during training process of deep learning network, the neurons will be temporarily discarded from the network according to a certain probability so as to solve problems including large amount of calculation and over-fitting in the network. In this paper, because the size of keystroke dataset we use is small, if the network we set is too large, it will lead to serious over-fitting which in turn affect the performance of the model. So, we add the dropout in the full connection layer, thus ensuring the network's generalization.

Multi-Class

As we know that CNN is quite sensitive to imbalanced data (Hensman & Masko 2015), but the imbalance of positive and negative samples is significant in keystroke authentication. No matter single classifier or binary classifier are used, it will lead to a decline in performance. Fortunately, the multi-classifier overcome this problem because each user inputs text freely for the same times in the data collection, so the sample is quite balanced. Meanwhile, the number of the output neurons from neural network namely indicates the number of categories, thus we get a multi-classifier. When a new input comes, we can compute the probability of this sample belonging to each class separately and the highest one indicates the most possible user. If the highest probability exceeds our pre-defined threshold, we infer that the new input comes from some person in our database, and we can tell whether he/she is a valid user. Otherwise, if the highest probability is smaller than the threshold, we tend to consider the new input is typed by an imposter.

Data and Experiment Design

Data

The data we use comes from Li and Liu (2016), who developed a Java applet based on HOOK to collect Chinese input keystroke data. There are 32 subjects entered freely by different users in the dataset. Each subject consists of nine texts with different topics, and each text contains at least 400 Chinese characters. That is to say, everyone will type nine times with approximate 400 Chinese characters corresponding to 9 topics. Finally, the keystroke data contains a total of 288 records, 546,414 keystrokes, and the average of each record contains 1897 keystrokes, the minimum is 1367 times, and the maximum is 3117 times.

Experimental Design

It was discovered by statistics that the keystrokes of each subject in our experiment contains 85 characters, in which the maximum number of keystrokes is the spacebar, with 66,811 keystrokes; and the minimum number of keystrokes is the keypad number 6 and F12 keys and so on, which only have 1 keystroke. In order to ensure the generalization of the model and discard part of the noise data, we select the top 70% keys, namely the first 60 keys, of which each key contains no less than 40 keystrokes, and then construct a 60×60 feature matrix based on these keys.

In the study of long and freely typed text, Curtin et al. (2006) pointed out that if the number of keystrokes is over 300 times, then results could likely accomplish higher accuracy. Moreover, it is believed that the more data in deep learning, the better performance could be in training process. In our experiment, to make full use of the existing keystroke data, we divide each sample into three subsamples or four subsamples. We regard the former as the Cutting Type I, the latter as the Cutting Type II, just like the experiment of Li and Liu (2016). At such, after the keystroke data is divided into three subsamples, each subject has 27 sample data, where its total to 864, and when divided into four subsamples, each subject has 36 sample data, where its total to 1152.

On the other hand, we suspect that different character arrangement orders in our feature matrices may affect the final performance of CNN model. Just like an image, we can get a totally different image by rearranging the pixels. Therefore, different character arrangement orders may result in completely different feature matrices, which may form input windows different from each other, just as mentioned before. In our study, we conduct experiments using two different character arrangement orders, one is in ascending order of ASCII values (Sorting Type I) and the other is in descending order based on the total number of keystrokes (Sorting Type II). Then, we cross these two Cutting Types with two Sorting Types, carrying out four experiments respectively as shown below:

- Experiment 1: Cutting Type I and Sorting Type I.
- Experiment 2: Cutting Type I and Sorting Type II.
- Experiment 3: Cutting Type II and Sorting Type I.
- Experiment 4: Cutting Type II and Sorting Type II.

After what we have done above, for each experiment, we use 10% of the data for the test set, while 10% of the remaining part for the validation set and the rest for the training set. We train the model with a 10 folds cross-validation method on a Linux computer with two Tesla M40 GPUs and 128G memory. Finally, we evaluate the experimental results with three criterions, including the average accuracy, average FAR and average FRR.

Results

The Analysis of Model Results

In our experiment, the default experimental setting is Cutting Type I and Sorting Type II, and the input of CNN model is the adjusted feature matrix. We roughly adjust the model parameters in attempt to make the model work well as much as possible and obtain a better prediction result.

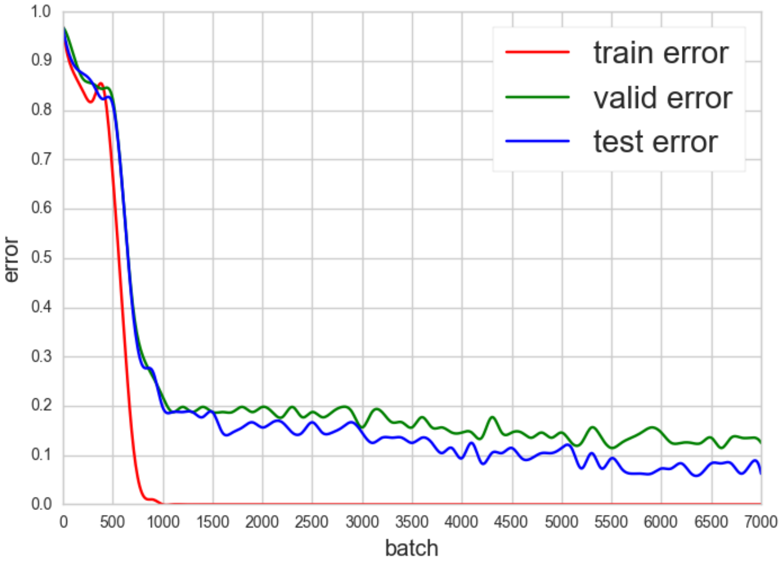


Figure 4. Different Errors in the Model Training Process

Figure 4 shows the variation of each error in the training process. It can be seen that our model fits keystroke data well, reducing training error to almost zero and testing error to around 0.2 in only one thousand batches of training, and gradually the latter error even close to 0.0625 at the end. However, it seems like a problem that the training error in the training process is perfectly fitted. But the validation error and test error of the model are still declining, which means that the model has been continuously optimized. Therefore, the perfect state of training error does not affect the overall training process of the model. The reason for this phenomenon comes from that the large training parameters in deep learning require a lot of data to be tuned, but the number of our keystroke data is only about one thousand or so, thus the model cannot fit parameters completely in time. In fact, instead of optimizing the training accuracy, the model just gradually optimize the output probability, and then gradually adjust parameters to improve the testing accuracy.

Experimental results

We input four datasets above related to the four experiment settings separately into not only the CNN model for training, but also the SVM model for doing a controlled trial to compare the performance of our CNN model. Specifically, we concatenate multiple feature matrices together and reshape into a vector, and acts as the input of the SVM. In addition, we adjust some parameters of two models to get better feedback. The results are shown as follow in Table 2.

Table 2. Model Performance in Different Experimental Settings

No. of Settings	Model	Average accuracy	Average FAR	Average FRR
Experiment 1	CNN	92.58%	0.24%	7.34%
	SVM	79.86%	0.79%	19.90%
Experiment 2	CNN	90.99%	0.31%	9.06%
	SVM	79.86%	0.79%	19.90%
Experiment 3	CNN	91.05%	0.30%	8.98%
	SVM	82.49%	0.68%	17.76%
Experiment 4	CNN	90.15%	0.32%	9.86%
	SVM	82.49%	0.68%	17.76%

The performance of two models on different datasets are summarized in Table 2, our CNN model achieves the average accuracy above 90%, and the highest accuracy achieved under the settings of Experiment 1 is 92.58%, which is 12.72% higher than that of SVM under the same experiment settings. At the same time, the average FAR for CNN is 0.24% in Experiment 1, which means that the model can reject imposers at a very high level. And the average FRR for CNN is 7.34%, which is a little high, but it's much less than our benchmark results. Besides, our CNN model can achieve the highest recognition accuracy of 96.09% in one of the 10-folds under the setting of Experiment 3. It can be seen that CNN performs better for mining implicit information in large-scale features.

Comparing Experiment 1 with Experiment 2, we find that the change of character arrangement orders from Sorting Type I to Sorting Type II does not affect the result of SVM, but it reduces the accuracy of CNN by 1.57%. Thus we believe that the change of character arrangement orders will slightly affect the training performance of CNN, and we get the same conclusion from the contrast between experiment 3 and experiment 4. From this, we suppose that there may be some kind of orders that will let CNN get a better classification result, however, we have not verified yet.

In addition, the two models have exactly the opposite performance when we just change the Cutting Type from I to II. It is obviously that the CNN model has a higher accuracy under condition of Cutting Type I, while the SVM performs better under condition of Cutting Type II. In fact, the amount of data has increased by only 288 when the condition changed as mentioned before, showing that CNN is not sensitive to a slight increase in data volume. Instead, if the length of each sample is too short, it might occur the loss of feature information, which affects the performance of CNN. On the contrary, it is more suitable for SVM. In our experiment, the dimensions of the input features are up to 18000, generating the dimensions of the SVM separating hyper plane are also high, which lead to that the small amount of data cannot find the optimal hyper plane, but increasing the amount of data can optimize this separating hyper plane to get a better solution.

The Performance of Feature Learning

In this part, we conduct another experiment using the experiment settings of Experiment 1, which produced the best performance (Cutting Type I and Sorting Type I). As for the feature, we use the long vector which is well-trained by CNN model and mentioned in Group Convolution (Section 3), then put it into SVM model to accomplish the multi-class classification in our paper. Besides that, we conduct a controlled trial by putting original time features into the same SVM model under the same condition, in order to compare performances of these two results.

Table 3. Performance of Different Features in Same Condition

Features	Model	Average accuracy	Average FAR	Average FRR
Well-trained	SVM-1	92.31%	0.26%	7.81%
Original	SVM-2	79.86%	0.79%	19.90%

As shown in Table 3，the performance of keystroke authentication by using the well-trained features can produce a 12.45% higher average accuracy than the original time feature. At the same time, the average FAR and FRR are also higher. It can be seen that the well-trained features have exactly had a good distinguishing ability, so the model with these features can recognize users at a very high level. In fact, it also shows that deep learning can indeed strengthen the representation ability of original features, automatically learn better features, then enhance the recognition performance of the model.

Conclusions and Future Work

In this paper, we have proposed a multi-user authentication method based on CNN model for free-text analysis to lighten the workload of feature extraction. To validate this method, we used the Chinese free-text keystroke data of Li and Liu (2016), which contains a total of 546,414 keystrokes from 32 participants. Then four experiments were conducted on the data in which the experimental results are described as follows: (1) The CNN model achieves the highest accuracy of 96.09% and the average accuracy of 92.58%, which is more than 10% higher than the SVM. And the average FAR and FRR reduce to 0.24% and 7.34% respectively, which means that it can really protect the system from invading in a continuous session, after the user has logged in. (2) The order of keys in the feature matrix will slightly affect the authentication performance of CNN. (3) The CNN model is not sensitive to a slight increase in keystroke data volume. Otherwise, if user can type more keystrokes in one session, the model can learn more features and perform better with an appropriate amount of data. (4)Our model accomplishes feature learning automatically, and gets satisfactory performance of keystroke authentication.

Generally speaking, it's just a primary study of keystroke authentication with CNN model for us in this paper. Surely, the proposed method has obtained satisfactory results, but some limitations still remain. First, this is not the best results of CNN model for the free-text analysis, and there are some ways to enhance the recognition performance or speed up the training speed such as increasing the amount of input data, changing the character orderings or adjusting the network structures. Second, when constructing feature matrix, we only employed the average time of five basic features, thus losing the sequence information and stability characteristics of user keystrokes. Therefore, it's considered that constructing the feature matrix using other organizing ways in the future. Third, in practical application scenarios, if a new user has been registered, it will cost much time to re-train a new CNN authentication model. Instead, we will do fine-tune on a well pre-trained CNN model with transfer learning in the future, so that the new model can converge quickly to meet the needs of practical applications. In summary, CNN model has many new possibilities in the area of keystroke recognition, and we will continue to improve and study in this field.

Acknowledgements

This work was supported by National Natural Science Foundation of China [91218301]; and the Key Laboratory of Financial Intelligence and Financial Engineering of Sichuan Province.

References

Alsultan, A., and Warwick, K. 2013. "Keystroke dynamics authentication: a survey of free-text methods," *International Journal of Computer Science Issues* (10.4), pp.1–10.

Alsultan, A., Warwick, K., and Wei, H. 2017. "Non-conventional keystroke dynamics for user authentication," *Pattern Recognition Letters* (89), pp.53-59.

Bours, P. 2012. "Continuous keystroke dynamics: A different perspective towards biometric evaluation," Information Security Technical Report (17.1-2), pp. 36-43.

Çeker, H., and Upadhyaya, S. 2017."Transfer learning in long-text keystroke dynamics," in Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on, IEEE, pp.1-6.

Curtin, M., Tappert, C., Villani, M., Ngo, G., Simone, J., Fort, H.S., and Cha, S. 2006. "Keystroke biometric recognition on long-text input: A feasibility study," Proc. Int. MultiConf. Engineers & Computer Scientists (IMECS).

Deng, Y., and Zhong, Y. 2015. "Keystroke dynamics advances for mobile devices using deep neural network," *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics* (2), pp. 59-70.

Ehrenfeld, J. M. 2017."Wannacry, cybersecurity and health information technology: A time to act," *Journal of medical systems* (41.7), pp. 104.

Gunetti, D., and Picardi, C.2005. "Keystroke analysis of free text." *ACM Transactions on Information and System Security (TISSEC)* (8.3), pp. 312-347.

Gunetti, D., and Ruffo, G. 1999. "Intrusion detection through behavioral data," in *International Symposium on Intelligent Data Analysis*, Springer, Berlin, Heidelberg, pp.383-394.

Hellström, E. 2018. "Feature learning with deep neural networks for keystroke biometrics: A study of supervised pre-training and autoencoders," pp.68.

Hensman, P., and Masko, D. 2015."The impact of imbalanced training data for convolutional neural networks," *Degree Project in Computer Science*, KTH Royal Institute of Technology.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. 2012. "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, 3(4), pp. 212-223.

Hu, J., Gingrich, D., and Sentosa, A. 2008. "A k-Nearest Neighbor Approach for User Authentication through Biometric Keystroke Dynamics," in *Communications, 2008. ICC'08. IEEE International Conference on*, IEEE, pp. 1556-1560.

Ioffe, S., and Szegedy, C. 2015. "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International conference on machine learning*.

Kang, P., and Cho, S. 2015. "Keystroke dynamics-based user authentication using long and free text strings from various input devices," *Information Sciences* (308), pp. 72-93.

Karnan, M., Akila, M., and Krishnaraj, N. 2011. "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing* (11.2), pp.1565-1573.

Kim, J., Kim, H., and Kang, P. 2018. "Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection," *Applied Soft Computing* (62), pp. 1077-1087.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp.1097-1105.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* (86.11), pp. 2278-2324.

Li, X., and Liu, J. 2016. "Keystroke Biometric Recognition on Chinese Long Text Input," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Cham, pp.260-271.

Magalh ães, P. S. T., and Santos, H. D. D. 2005. "An improved statistical keystroke dynamics algorithm," in *Iadis Virtual Multi Conference on Computer Science and Information Systems*, pp.223-227.

Mandujano, S., and Soto, R. 2004. "Deterring password sharing: User authentication via fuzzy c-means clustering applied to keystroke biometric data," in *Computer Science, 2004. Proceedings of the Fifth Mexican International Conference in*, IEEE, pp.181-187.

Messerman, A., Mustafic, T., Camtepe, S. A., and Albayrak, S. 2011. "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics," in *Biometrics (IJCB), International Joint Conference on*, IEEE, pp.1-8.

Monrose, F., and Rubin, A. 1997. "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM conference on Computer and communications security, ACM*, pp.48-56.

Montalv ão Filho, J. R., and Freire, E. O. 2006. "On the equalization of keystroke timing histograms," *Pattern Recognition Letters* (27.13), pp.1440-1446.

Park, S., Park, J., and Cho, S. 2010. "User authentication based on keystroke analysis of long free texts with a reduced number of features," in *Communication Systems, Networks and Applications (ICCSNA), Second International Conference on*, IEEE, Vol. 1, pp.433-435.

Sim, T., and Janakiraman, R. 2007. "Are digraphs good for free-text keystroke dynamics?" in *Computer Vision and Pattern Recognition, CVPR'07. IEEE Conference on*, IEEE, pp.1-6.

Singh, S., Arya, K. V. 2011. "Key classification: a new approach in free text keystroke authentication system, " *Circuits, Communications and System (PACCS), 2011 Third Pacific-Asia Conference on*, IEEE, pp. 1-5.

Spillane, R. 1975. "Keyboard apparatus for personal identification," *IBM Technical Disclosure Bulletin* (17.3346), pp. 3346.

Sun, L., Wang, Y., Cao, B., Yu, P. S., Srisa-An, W., and Leow, A. D. 2017. "Sequential Keystroke Behavioral Biometrics for Mobile User Identification via Multi-view Deep Learning," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Cham.

SungzoonCho, ChigeunHan, and Heehan, D. 2000. "Web-based keystroke dynamics identity verification using neural network," *Journal of organizational computing and electronic commerce* (10.4), pp. 295-307.

Villani, M., Tappert, C., Ngo, G., Simone, J., Fort, H. S., and Cha, S. H. 2006. "Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions," in *Computer Vision and Pattern Recognition Workshop, 2006, CVPRW'06, Conference on*, IEEE, pp.39.

Vizer, L. M., Zhou, L. and Sears, A.2009. "Automated stress detection using keystroke and linguistic features: An exploratory study," *International Journal of Human-Computer Studies* (67.10), pp. 870-886.

Yu, E., and Cho, S. 2003. "GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification," in *Neural Networks, 2003, Proceedings of the International Joint Conference on*, IEEE, Vol. 3, pp.2253-2257.