HandiText: Handwriting Recognition Based on Dynamic Characteristics with Incremental LSTM

25

LIMING FANG, HONGWEI ZHU, and BOQING LV, Nanjing University of Aeronautics and Astronautics, China

ZHE LIU, Nanjing University of Aeronautics and Astronautics and State Key Laboratory of Cryptology, China

WEIZHI MENG, Technical University of Denmark, Denmark

YU YU, Nanjing University of Aeronautics and Astronautics, China

SHOULING JI, Zhejiang University, China and Georgia Institute of Technology, USA

ZEHONG CAO, University of Tasmania, Australia

The Internet of Things (IoT) is a new manifestation of data science. To ensure the credibility of data about IoT devices, authentication has gradually become an important research topic in the IoT ecosystem. However, traditional graphical passwords and text passwords can cause user's serious memory burdens. Therefore, a convenient method for determining user identity is needed. In this article, we propose a handwriting recognition authentication scheme named HandiText based on behavior and biometrics features. When people write a word by hand, HandiText captures their static biological features and dynamic behavior features during the writing process (writing speed, pressure, etc.). The features are related to habits, which make it difficult for attackers to imitate. We also carry out algorithms comparisons and experiments evaluation to prove the reliability of our scheme. The experiment results show that the Long Short-Term Memory has the best classification accuracy, reaching 99% while keeping relatively low false-positive rate and false-negative rate. We also test other datasets, the average accuracy of HandiText reach 98%, with strong generalization ability. Besides, the 324 users we investigated indicated that they are willing to use this scheme on IoT devices.

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61872181, 61802180, 61772466, U1936215 and U1836202; the Natural Science Foundation of Jiangsu Province under Grant No. BK20180421; the National Cryptography Development Fund under Grant No. MMJJ20180105; the Fundamental Research Funds for the Central Universities under Grant No. NE2018106; the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant No. LR19F020003; the Provincial Key Research and Development Program of Zhejiang under Grant No. 2019C01055; the Ant Financial Research Funding, and the Alibaba-ZJU Joint Research Institute of Frontier Technologies.

Authors' addresses: L. Fang, H. Zhu, B. Lv, and Y. Yu, Nanjing University of Aeronautics and Astronautics, 29 Jiangjun Avenue, Jiangning District, Nanjing City, Jiangsu Province, China; emails: {fangliming, zhuhongwei}@nuaa.edu.cn, lbqvn@gmail.com, 18010787771@163.com; Z. Liu (corresponding author), Nanjing University of Aeronautics and Astronautics, 29 Jiangjun Avenue, Jiangning District, Nanjing City, Jiangsu Province, China and State Key Laboratory of Cryptology, P. O. Box 5159, 100878, Beijing, China; email: liuzhenuaa@163.com; W. Meng, Technical University of Denmark, Richard Petersens Plads, 2800 Kgs. Lyngby, Denmark; email: weme@dtu.dk; S. Ji, Zhejiang University, No. 38, Zheda Road, Xihu District, Hangzhou City, Zhejiang Province, China and Georgia Institute of Technology, USA; email: shoulingji@163.com; Z. Cao, University of Tasmania, Churchill Avenue, Hobart TAS 7005, Australia; email: zhcaoctu@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2577-3224/2020/11-ART25 \$15.00

<https://doi.org/10.1145/3385189>

CCS Concepts: • **Security and privacy** → **Biometrics**;

Additional Key Words and Phrases: Data Science, Internet of Things, Authentication, Handwriting, LSTM

ACM Reference format:

Liming Fang, Hongwei Zhu, Boqing Lv, Zhe Liu, Weizhi Meng, Yu Yu, Shouling Ji, and Zehong Cao. 2020. HandiText: Handwriting Recognition Based on Dynamic Characteristics with Incremental LSTM. *ACM/IMS Trans. Data Sci.* 1, 4, Article 25 (November 2020), 18 pages.
<https://doi.org/10.1145/3385189>

1 INTRODUCTION

Because the Internet of Things (IoT) applications about data science are characterized by different devices, networking standards, and data management strategies, authentication becomes a very important issue in the IoT ecosystem, such as ensuring the trustworthiness of data about IoT devices. Different IoT devices or applications on the IoT devices store users' privacy information to different degrees, and, according to Gartner, the number of IoT devices will reach 26 billion by 2020 [32], so how to protect the privacy of these devices is also getting more and more attention.

Today's smart locks can be opened by entering a password, entering a fingerprint, or recognizing a face. Although entering the password is convenient, it is easy for the attacker to peep to get the password [30]. Fingerprints and faces ensure security to a certain extent; however, with the development of technology, the cost of identifying devices is not only high, but there are still some difficulties in the liveness detection [24]. Experiments show that both fingerprint recognition and face recognition can be cracked by three-dimensional (3D) printing technology [14]. Moreover, most of the current IoT devices can be controlled by mobile phone applications, and most of the unlocking methods on the mobile phone include entering a password, entering a fingerprint, or recognizing a face, so people need a safe and convenient authentication method.

For mobile terminal devices, such as mobile phones, authentication schemes can be varied. For the traditional text-based password login authentication scheme, not only is their password reuse degree is high [29, 34], but also they are not resistant to many kinds of attacks, such as brute-force attack, random guessing attack [8], dictionary attack [33], shoulder-surfing attack [18, 20], smudge attack [17, 28], recording screen attacks, and so on. Therefore, the security of this kind of password is low, and the user's memory burden is increased at the same time. To reduce users' memory burden, image-based passwords have been gradually developed. However, the password method is not very resistant to shoulder-surfing attack, smudge attack, and recording screen attacks, and so on, so the login authentication schemes based on biological features have become popular.

The login authentication schemes based on biological features use a biometric of human beings that cannot change to carry out identity authentication. It is difficult for attackers to imitate these biometrics, such as iris [22, 35], fingerprint [11], face [23], and so on. Some schemes (such as fingerprint and iris authentication methods) adopt static biometric authentication, which provides attacker with opportunities to log in illegally, thus threatening people's privacy and property security.

So for static biometric features like irises and fingerprints, it is easy to crack, and attackers can fake it with 3D printing [1, 27]. By early 2019, NVIDIA cracked facial recognition with a 95% success rate using Generative Adversarial Networks (GAN) [14]. Therefore, the existing login authentication schemes based on biological characteristics are no longer safe. People need new, convenient, and secure login authentication schemes.

In this article, based on the static and dynamic features of biology, we propose a login authentication scheme named HandiText, which is safe and convenient. We collect pressure, speed, and other features of people in the writing process, use a machine learning algorithm to train the data,

and classify users' login authentication data to complete the identity. By comparing Long Short-Term Memory (LSTM) [9], Supporting Vector Machine (SVM) [26], K-Nearest Neighbor (KNN) [36], Logistics Regression (LR) [7], and other algorithms, we found that LSTM can reach the best accuracy of 99% with a lower false-positive rate and false-negative rate, and the training time of the model is short as well, which makes it easier to use. We also limit the time of collecting user writing data, which makes it difficult for attackers to conduct an impersonation login operation, which is safer than traditional solutions. HandiText is also tested on other datasets, such as MNIST [6], IAM [3, 21], and CASIA [19, 25], the average accuracy is over 98%, and it has a low false-positive rate and false-negative rate; it is proven that our scheme has a strong generalization ability.

1.1 Contributions

Our main contributions are summarized as follows:

- We establish a Chinese dynamic handwritten dataset, with a total of 2,490 handwritten data, including dynamic behavior features such as pressure and speed. It makes up for the deficiency of Chinese handwritten dataset and adds online features compared with the offline database, making the data features more abundant.
- Based on the biological static and dynamic characteristics, we propose a login authentication scheme named HandiText, which is safer than traditional schemes. Compared with traditional offline handwriting recognition, it has more advantages because of its online and dynamic features. At the same time, because of its dynamic characteristics, it is difficult for attackers to imitate, so HandiText can resist shoulder-surfing attacks, smudge attacks, and screen recording attacks.
- By comparing with LSTM, SVM, KNN, LR, and other algorithms, we find that LSTM has a lower false-positive rate and false-negative rate with a high accuracy of 99%, and the identity time is similar to TouchID and FaceID.
- HandiText is also tested on other datasets, such as MNIST [6], IAM [3, 21], and CASIA [19] [25], and we find that the average accuracy is over 98%, which proves the availability of our scheme.

1.2 Outline

The organizational structure of this article is as follows: Section 2 introduces biometry-based existing schemes. Section 3 introduces a login authentication scheme based on biological static and dynamic characteristics called HandiText. Section 4 carries out comparison experiments to select the best algorithm by comparing LSTM, SVM, KNN, LR, RNN, GRU, and HMM and shows the results of being tested on other datasets to verify the security and generalization ability of HandiText. Section 5 collects user preferences to verify the usability of HandiText. Section 6 describes our future work and summarizes the article.

2 RELATED WORK

For the traditional text-based password login authentication scheme, their password reuse degree is not only high [29, 34], but also does not easily resist many kinds of attacks, such as brute-force attacks, random guessing attacks [8], dictionary attacks [33], shoulder-surfing attacks, smudge attacks, recording screen attacks, and so on. Therefore, the security of this kind of password is low, and the user's memory burden is increased at the same time. To reduce users' memory burden, image-based passwords have been gradually developed. However, this password method is also difficult to resist shoulder-surfing attacks [18, 20], smudge attacks [17, 28], recording screen attacks, and so on, so the login authentication schemes based on biological features have become popular.

This section mainly shows the login authentication schemes based on biological features, including iris recognition, fingerprint recognition, face recognition, handwriting recognition, and so on, and analyzes the advantages and disadvantages of each scheme.

2.1 Iris Recognition

Iris recognition is based on the iris in the eye for identification [35]. The iris contains a lot of inter-lacing spots, filaments, coronae, stripes, recesses, and other details, which determine the uniqueness of the iris. Iris recognition is typically used in highly secure locations such as banks, vaults, and so on. Some IoT devices with high security requirements also have iris recognition, such as smart locks.

Iris recognition and verification is generally performed by shining infrared light on the eyes through LED lights on the device and then capturing iris images by infrared camera, which then uses algorithms for positioning and preprocessing. The last step, like fingerprint identity, is normalization, key point extraction, and comparison with database information.

Since most hardware uses iris recognition [1], there is no vivisection at all, and some can be bypassed, so people can log in by impersonating others. Because most hardware uses iris recognition [1], there is no liveness detection at all, and some can be bypassed, so people can authenticate by impersonating others. Therefore, iris recognition based on biometric-based login authentication is no longer safe [12], and because cameras that recognize the iris are expensive, the method has been gradually abandoned.

2.2 Fingerprint Recognition

Fingerprint recognition applications are more widely used, especially in recent years, applied to mobile device unlocking [11], such as mobile phone screen unlock, mobile payment, computer screen unlock, and so on. Because most IoT devices are currently controlled by mobile phone applications, the security of fingerprint recognition is not to be underestimated. Although fingerprint identification is convenient and easy to use, it is the most easily cracked form of authentication due to its non-deformation and static characteristics. For example, fingerprint identification can be easily cracked through 3D printing and fingerprint film, thus posing a threat to the user's property [14]. Therefore, fingerprint recognition is no longer safe.

2.3 Face Recognition

Face recognition is also becoming more widely used, such as mobile phones [23], smart locks, and so on. The difference between face recognition and other biometric features lies in the fact that face recognition mainly uses liveness detection, but it can also be successfully recognized by using people's static photos combine with certain image processing technology and dynamic synthesis technology.

By early 2019, NVIDIA cracked facial recognition with a 95% success rate using GAN [14]. At the Black Hat 2019, Tencent invented a pair of glasses that can crack the face recognition by destroying the liveness detection in face recognition [2]. Therefore, the existing login authentication schemes based on biological characteristics are no longer safe. People need new, convenient, and secure login authentication schemes.

2.4 Handwriting Identity

Because the existing login authentication schemes based on biometrics are easy to crack, people began to pay attention to the new login authentication schemes based on handwriting recognition.

Most of the existing handwriting recognition and authentication systems are mainly in English. According to the research by Han et al. [4], Chinese and English and other language structures

are very different, for example, there are more strokes used to write Chinese characters, but the features system of handwriting features is similar, that is, the features of the language writing structure have little effect on the classification. Jia's research shows that although Chinese handwriting is more complicated than English due to the complexity of Chinese itself, there is no essential difference between Chinese and English handwriting identification [16]. Therefore, in this article, our research object is Chinese.

2.4.1 Multi-touch. Multi-touch [37] is based on the mixture of biometric features and behavioral features. It collects users' hand geometry, including size of hand and space between fingers, and habits of swipe fingers, including velocity, pressure, and angle. Combining biometric and behavioral features, using machine learning algorithms, the accuracy of recognition is very high. However, the method is not easy to perform on a small phone screen. Some screens are not big enough to contain four fingers. So it is unfriendly to users' behavior. To solve the problem, users have to use three or fewer fingers to swipe across the screen. In this case, the error ratio will increase from 2% to 7.7%, because the number of biometric features is reduced.

2.4.2 Humm's Scheme. Humm et al. put forward a new strategy [13] that collects both online notes and voice at the same time. It only needs users to indicate what they are writing. The advantage of this method is that it can make the authentication results more accurate without increasing the access time. However, in the process of using microphones, there are certain requirements for hardware equipment, which may cause inconvenience to users.

2.4.3 TextCaps. Because handwritten datasets are generally difficult to collect [15], figuring out a way to use a small amount of data to complete accurate analysis has become a research difficulty. In this article, the author proposes that only 200 data quantities can be used to generate more data by algorithmic transformation. These data have certain fuzziness, but because of the increase of data quantities and training set, the accuracy of classification can be increased. This is very enlightening.

Therefore, in general, the existing login authentication scheme dataset for handwriting recognition is too small, and the authentication scheme does not consider the online problem of time sequence or requires other additional hardware devices. Therefore, we propose HandiText, an online dynamic handwritten recognition scheme that takes time sequence into account and does not need other additional hardware devices.

In summary, people need a secure and convenient login authentication solution and do not need additional hardware equipment, which reduces costs.

3 OUR SCHEME: HANDITEXT

Because existing handwritten recognition schemes have problems, such as excessive data collection burden, false-positive rate, or a too-high false-negative rate, we need a handwritten recognition scheme that only needs to collect handwritten data of users and ensures high accuracy but also has low false-positive rate or false-negative rate.

In this section, we will introduce our handwriting recognition scheme, HandiText. First, we will start from the design goals of the scheme and then introduce our scheme for HandiText.

3.1 Design Goals

Because our goal is to design an authentication scheme with high speed, high security, and good usability, the scheme needs to meet the following requirements.

3.1.1 High Speed. When new data appear, the algorithm should not take too long to train new data to ensure that users do not need to wait too long for the process of authentication.

Fig. 1. The composition and workflow of our scheme.

3.1.2 High Security. Places requiring authentication generally protect the privacy or property of users, so high security is required to ensure that the secret keys of authentication are not easily cracked by attackers. In this scheme, the algorithm is required to have high accuracy, and the false-positive rate or false-negative rate should also be kept at a low value.

3.1.3 Good Usability. For traditional authentication, users usually need to remember high-strength text passwords made up of numbers, letters, and symbols, which is also a challenge for users' memory. And with the increase of password reuse times [5, 34], its security is also difficult to guarantee, so we need an authentication scheme that users can easily operate without requiring too much memory.

As shown in Figure 1, to meet the above requirements, we designed a handwritten authentication scheme by capturing users' behavior and biometrics. The left part of the figure is the visual front-end of the user, and the right part of the figure is the back-end for user data collection and training. The implementation will be introduced in Section 3.3 and Section 3.2.

3.2 Implementation

This subsection mainly introduces the scheme composition, algorithm, and features used in the algorithm. The interface between the input and output of the data and the algorithm is established, and the algorithm is encapsulated independently, which is convenient for the improvement of the algorithm later.

3.2.1 Modules. As shown in the right part of Figure 1, the algorithm module is mainly divided into training module and classification module, which will be introduced in detail below.

- **Training Module.** As shown in the upper right part of Figure 1, when the user registers, the data will be stored in the database and the training module can get the data from the database. The first step is to filter the data, normalizing the data while removing the dirty data. After data filtering, the data will be trained, and the training model is the output of the training module.

To enhance the classification accuracy, we adopt the incremental algorithm. Each time the user completes the authentication, new data will be generated and stored in the database.

Fig. 2. The description of pressure.

When the new data reach a certain amount, we will retrain the new data to improve the accuracy of the algorithm.

- **Classification Module.** As shown in the lower right part of Figure 1, when the user authenticates, he or she needs to enter the user's secret key, namely the handwritten data of the user. This data are taken as the input of the classification module and will be classified according to the model trained by the training module, and the classification result will be sent back to the front-end by the back-end. Because this is a single classification, the back-end result only needs to determine whether the data are the user's correct secret key.

3.2.2 Data and Features. Traditional writing is presented offline [10], and an attacker can simulate a user for authentication as long as the user's handwriting is sufficiently similar. Therefore, from the perspective of time, we adopt the online mechanism to collect the dynamic changes of behavior characteristics in the writing process, so as to ensure that it is difficult for attackers to imitate the writing process of ordinary users, so that they cannot pretend to be ordinary users for authentication, greatly improving the security of authentication by handwriting.

- **Time.** Because handwriting is a continuous process, the state at one moment is highly relative to that at previous and next moment, it is natural to collect time-series data as our dataset. Within 3 s, we capture a fixed time spot every 20 ms and record the information of these time spots to characterize the whole handwriting process approximately. If the data are beyond 3 s, then they will be deleted, and if the data are deficient in 3 s, then they will be completed with the last valid data. When the last valid data completion occurs, it proves that the user's signature time is less than 3 s. Because the same user writes the same word, the time it takes should be similar, and the user's real handwriting time can be used as a feature.
- **Relative Position.** We define the first recorded touch spot as the origin of coordinates, so all the following spots' positions can be considered as the relative position of the first spot. We use X and Y coordinates to represent each point's position. Knowing the coordinates and interval, we can figure out the average linear velocity between any two spots. As shown in Figure 2, we recorded the position coordinates of points A , B , and C . We can get the vector $\vec{BC} = \vec{AC} - \vec{AB}$. At the same time, the interval between points B and C is 20 ms. We can get the linear velocity v and angular velocity ω from point B to point C .
- **Pressure Value.** We capture each point's pressure value, for example, we recorded the pressure values at points B and C , instead of the mean pressure values from points B to C . The pressure that the user used in handwriting process were recorded and transferred into

Table 1. User Handwritten Data Storage Format

No.	X	Y	Pressure	Description
1	0.0	0.0	0.301961	Point A in Figure 1
2	3.320862	−2.0	0.294118	
3	20.62878	0.9912712	0.286275	
.....				
43	−62.48285	445.135	0.200002	Point B in Figure 1
44	−2.0	493.0	0.137255	Point C in Figure 1
.....				
104	−87.0	925.0	0.184314	Time is 2.08 s
.....				
149	−87.0	925.0	0.184314	Time remains the same as No. 104.
150	−87.0	925.0	0.184314	Time remains the same as No. 104.

numerical values between 0 and 1. If the user's phone is not pressure-support, then the pressure values are all 0. During data processing, we can obtain the average pressure value, the maximum pressure value, the minimum pressure value in the entire writing process from the records, and the pressure value of each recorded point can be obtained by combining with the time point.

As mentioned, each spot includes three numerical values: XY coordinates and pressure value, and the sequential form of data also implicate the time and speed information, which do not manifest directly.

Altogether, in one handwriting, we have 3 features in each time spot, and the number of spots is 150. That means one handwriting has 450 features in total. Like normal sequential data, each spot is relative to the adjacent spots, so the data series should be taken as whole, shown as Table 1.

3.2.3 Algorithm. We examine the data arrangement and chose *LSTM* for our learning algorithm. There are two main reasons to choose this algorithm. First, the data we have are sequential (the current data are relative to the data input before and next), so it is natural to come up with the idea to use Recurrent Neural Network. Second, a datum is much more relative to its neighbor than a datum that is far from it, so to forget the older data, it is better to use LSTM with a forget gate than normal RNN.

Normal RNN concatenates the current input x_i and the previous output h_{t-1} to a new vector and puts it into the network, so the information can be transmitted to the next step. But the training step is complicated. The complexity of network is determined by timestep; if a timestep is very large, then it is highly possible to cause gradient vanish and gradient explode, which may slow down learning speed.

LSTM is a special type of RNN that can learn long-term information dependence. LSTM avoids gradient vanish and gradient explode by a deliberate design called a "gate." Gates are a way to optionally let information through. An LSTM has three of these gates, an input gate, an output gate, and a forget gate, to protect and control the cell state.

Because our data collection system continuously collects new data according to the daily usage of users, we combine LSTM with incremental learning, which can improve the classification accuracy of the data while shortening the model training time in a later period.

Let A be the initial training sample set, B be the increment set, A_T be the output of classifier T_1 , B_{err} be the test error set, and B_{corr} be the test correct set. We first train classifier T_1 with the

Fig. 3. Interface of HandiText Design.

initial training samples set A and, according to classifier T_1 to verify B , divide B into test error set B_{err} and test correct set B_{corr} . If $B_{err} \neq \emptyset$, then take the union of A_T and B_{err} as the new training set and obtain T_2 . For A , the union set of the residual samples of A_{sv} and B_{corr} is removed as the incremental sample set B' of the T_2 classifier. Let $B = B'$, and repeat previous steps. If $B_{err} = \emptyset$, then the repetition ends. The output of this algorithm is the final trained classifier.

To improve the training time of the data and the classification accuracy of the algorithm, the combination of LSTM with incremental learning showed good results in the experiments in Section 4.

3.3 Using Process

In this subsection, we will explain the user using process. As shown in the left part of Figure 1, when the user registers and authenticates, he or she needs to input his or her user name and mobile phone number as the userID, which can be uniquely identified, and then enter the phase of registration and authentication as required.

3.3.1 Registration Phase. As shown in Figure 3(a), after the user inputs his or her user name and mobile phone number, he or she will click “Register” to enter the registration phase. In the registration phase, the user needs to write a certain word several times to complete the registration, as shown in Figure 3(b).

3.3.2 Authentication Phase. As shown in Figure 3(a), after entering his or her user name and mobile phone number, the user clicks “Authenticate” to enter the authentication phase. In the authentication phase, the user needs to write the word in the registration phase by hand, and the system will return the result to prompt the user to authenticate successfully or fail, as shown in Figure 3(c).

4 ALGORITHM COMPARISONS

To find the optimal algorithm that can accurately classify whether it is handwritten by the user, we design some experiments, mainly comparing the classification accuracy, false-positives rate, and false-negative rate of different machine learning algorithms for a handwriting dataset. Therefore, we express our algorithms' comparisons from the perspective of experiment preparation, experiment implementation, and experiment results.

4.1 Experiment Preparation

What we do is a handwritten authentication system based on user behavior and biological features, which can be applied in different places, such as IoT devices. Because most of the current IoT devices control and manage devices through mobile apps, to facilitate the detection of the accuracy of our system, we will conduct experiments on a mobile phone.

4.1.1 Market Investigation. In Section 3.2.2, the features we need to collect include the pressure of users at every moment in the handwriting process, so we need to investigate whether mobile phones in the market can collect the pressure. According to the results of the survey, since the Android ecosystem does not support pressure-sensitive screen very much, there are still some phones that do not support pressure collection. Even if Android supports the function of Force Touch, this function is built on the basis of Android's pressure-sensitive screen [31]. Fortunately, phones produced after 2016 can support pressure-sensitive screen, such as the HUAWEI P20 and iPhone environment is fairly commons, with all models since the iPhone 6S supporting pressure collection, which is beneficial to our research.

4.1.2 Experiment Setting. Because we encapsulate the algorithm embedding module when implementing the scheme, we could experiment with different algorithms and compare their accuracy, false-positive rate, false-negative rate, and other dimensions. Our algorithms are all tested in the same environment to ensure the most accurate security comparison under the same environment.

4.1.3 Dataset. To facilitate the collection of training data, we design an application based on Android and iOS. The application requires the user to enter the user name and phone number and then convert his or her handwriting into a ".csv" file vertically based on the features in Section 3.2.2 and transfer it into numbers to the database. One user can write 150 pieces of handwritten data at a time, and all of one user's data is stored in a ".csv" file. In the process of data collection, we collected 2,490 data for model training, and the data are still increasing now. This dataset can be expanded with the use of users, which helps us to incrementally train the model, improve the accuracy of system recognition, and make generalization performance better, because the data involve user privacy issues, so we do not openly deal with it.

4.1.4 Participants. We randomly select 500 people from Nanjing University of Aeronautics and Astronautics (NUAA) and Nanjing Zhongyi Union Internet of Things Company as our training dataset sources. Before their data collection, we train them. We ask them to download our data collection application, and each person completes at least 10 complete word writing procedures, so at least 5,000 pieces of data were generated. However, because some mobile phones did not support pressure-sensitive screens and other reasons, the final valid data were 2,490. HandiText is trained for each user separately. Therefore, when performing binary classification training, the data of this user is marked as 1, and the data of other users are marked as 0.

4.1.5 Experiment Assessment. To better evaluate our security experiments, we mainly use the following indicators:

Table 2. The Results of Each Algorithms

No.	Algorithm	F-score	Accuracy	PRE	RR	FPR	FNR
1	LSTM	0.9266	99.06	93.18	92.14	0.46	7.86
2	SVM	0.5854	94.25	69.77	50.42	1.92	49.58
3	KNN	0.7583	96.45	75.83	73.39	1.66	26.61
4	LR	0.3375	91.32	33.75	24.77	2.16	75.23
5	RNN	0.8000	97.89	72.73	88.89	1.67	11.11
6	GRU	0.8889	98.94	88.89	88.89	0.56	11.11
7	HMM	0.0015	7.63	0.75	1.00	92.43	0

Table 3. Comparison Before and After Incremental Learning

	ACC	RR	TNR	FNR	FPR	PRE	F-score	Time
Before Incremental Learning	99.06	92.14	99.54	7.86	4.64	93.18	0.9266	73s
After Incremental Learning	99.37	98.87	99.38	1.12	0.62	91.67	0.9513	54s

- Accuracy (ACC) = $(TP + TN) / (TP + TN + FP + FN)$
- True-Positive Rate (TPR) = $TP / (TP + FN)$ = Recall Rate (RR)
- True-Negative Rate (TNR) = $TN / (FP + TN)$ = $1 - FPR$
- False-Positive Rate (FPR) = $FP / (FP + TN)$
- False-Negative Rate (FNR) = $FN / (TP + FN)$ = $1 - TPR$
- Precision (PRE) = $TP / (TP + FP)$
- F-score = $2 \cdot PRE \cdot RR / (PRE + RR)$

4.2 Experimental Implementation and Results Analysis

First, we checked the dataset, finding that the values of the XY coordinates are much higher than the values of pressure, and in a few of writings, the 150 pressure values are identical. By examining the source code of APP, we found that some smart phones are not pressure supporting. To make the classification more accurate, we deleted the data with identical pressure values.

Second, when more and more people are used and more and more data records are generated, it means that more and more data are marked as 0, so that there is an imbalance between the positive and negative samples. To balance the samples, we define the ratio of positive and negative samples as 1 : 4, which is also the best ratio obtained through our experiments.

To better compare the effects of different algorithms, we select the sample of the same user A as the positive sample and randomly select the samples of the remaining users as the negative sample. After data cleaning and regularization, we select 151 samples, where 30 of them are written by user A, which are considered as the positive sample and labeled as 1, and the rest are written by other users, considered as the negative sample and labeled as 0. Then, we randomly split the dataset into a training dataset, containing 113 samples, and a test dataset, containing 38 samples. Then, we used KNN, SVM, LR, LSTM, RNN, GRU, and HMM and fed the training dataset to these models, respectively. We used the test dataset to test their performance. The test results are shown in Table 2 and Figure 4. This is only the data used in the case of the comparison algorithms, not the sample data actually used.

To improve the training time of the data and the classification accuracy of the algorithm, the combination of LSTM with incremental learning showed good results in the experiments. As shown in Table 3, after performing incremental LSTM learning, comprehensive evaluations such as accuracy and F-score show that incremental learning can indeed shorten training time and

Fig. 4. The ROC curves of each algorithms.

Table 4. LSTM Experiment Results on Our Dataset

No.	Learning rate	Number of neurons	Number of iterations	Loss	Accuracy	Training Time
1	0.01	100	50	0.06	99.06	—
2	0.01	200	50	0.01	98.43	—
3	0.01	50	50	0.05	99.01	—
4	0.01	100	200	—	97.62	3 min 1 s
5	0.008	200	50	0.007	96.55	3 min 14 s
6	0.008	100	50	0.017	99.06	2 min 15 s
7	0.015	200	50	0.003	99.06	1 min 13 s
8	0.015	100	50	0.02	99.06	1 min 13 s
9	0.015	50	50	0.02	99.06	23 s
10	0.008	50	50	0.06	97.79	37 s

improve classification accuracy. The accuracy of HandiText was improved from 99.06% to 99.37%, and the time to train the model was reduced from 73s to 54s. When incremental learning is continued, the accuracy will still improve, and the training time will be reduced, which will help improve the performance of HandiText and enhance its generalization ability.

Table 4 is a detailed test information of the LSTM algorithm, including the learning rate, the number of neurons, the number of iterations, and the loss function for each experiment. It can also be seen that under the premise of ensuring the accuracy, each training does not take too long; that is, after each registration, the user does not have to wait too long to perform the authentication operation.

Average accuracy for each algorithm is as follows: 99.06% for LSTM, 96.45% for KNN, 91.32% for Logistic Regressive, 94.25% for SVM, 97.89% for RNN, 98.94% for GRU, and 97.85% for HMM. Unlike SVM, KNN, and other algorithms, GRU and HMM are also time sequence algorithms. From Table 2, we can get that although GRU has achieved good results and HMM experimental results are not very good, the performance of LSTM is still superior. And from the perspective of FPR and FNR, the performance of LSTM is better than other algorithms, so we adopt LSTM as our final algorithm.

Table 5. Test Results on Different Datasets

Dataset	F-score	Accuracy	PRE	RR	FPR	FNR
Our Chinese Dataset	0.9266	99.06	93.18	92.14	0.46	7.86
Our English Dataset	0.9163	98.75	86.92	96.88	1.10	3.13
MNIST [6]	0.8571	98.29	80.90	91.14	1.28	8.86
IAM [3, 21]	0.8831	98.72	82.93	94.44	1.05	5.56
CASIA [19, 25]	0.8877	98.51	87.37	90.22	0.91	9.78

4.3 Generalization Ability Test

After completing the training on our handwritten dataset, we need to test the generalization ability of HandiText on other datasets. We test it with the LSTM algorithm on the MNIST [6], IAM [3, 21], and CASIA [19, 25] datasets, and the test results are shown in Table 5.

In the three datasets, the accuracy of HandiText is greater than 98%, and it also guarantees low false-positive rates and false-negative rates, so HandiText has a good generalization performance.

In addition, to test whether the complexity of the language structure has an impact on the performance of the HandiText system, we used English and collected the same features for experiments. The experimental results are shown in Table 5. Even if the same features are adopted in English and the LSTM algorithm is used for training, the accuracy of the HandiText system is also high, achieving 98.75%. This proves that HandiText can be applied to other languages, and the complexity of language structure has less influence on the dynamic behavior of people when handwriting.

5 PERFORMANCE EVALUATION OF HANDITEXT

Because most IoT devices can be accessed and controlled through mobile apps, mobile authentication is also critical. In this section, we experimentally compare the following metrics to assess the practical performance of a handwritten authentication scheme (HandiText) based on user behavior and biometrics:

- All participants are needed to register and authentication, and we count the final authentication success rate.
- The time token for new users and experienced users to register and authenticate to the system.
- The usability preferences of users in different scenarios.
- One person imitates another person's handwriting for authentication and counts its success rate.

5.1 Experimental Preparation

To conduct a usability experiment, we improve the implementation of our scheme. In addition, after the experiment, we set up a questionnaire to collect users' feedback on our experiment.

5.1.1 Experimental Setting. We used the same application implementation as the security experiment, but the algorithm fixes the LSTM. At the same time, we add the time interface required for user registration and authentication and the specific date and time point of each registration and authentication.

Table 6. Details of Participants

	No.	Percentage
Gender	324	100%
Male	157	48.4568%
Female	167	51.5432%
Age	324	100%
20–29	269	83.0247%
30–39	36	11.1111%
40–49	12	3.7037%
50–59	7	2.1605%
Profession	324	100%
Student	232	71.6049%
Teacher	22	6.7901%
Engineer	61	18.8272%
Business	9	2.7778%
Phone Brand	324	100%
iPhone	82	25.3086%
HUAWEI	117	36.1111%
OPPO	27	8.3333%
VIVO	23	7.0988%
XIAO MI	59	18.2099%
Samsung	3	0.9259%
Others	13	4.0123%
Support Pressure Detection	324	100%
Yes	249	76.8519%
No	75	23.1481%
Authentication Situation	324	100%
Success	324	100.0000%
Failure	0	0.0000%
Imitation Situation	20	100%
Success	0	0.0000%
Failure	20	100.0000%

5.1.2 Participants. We recruit 324 participants for the experiment, and each participant conduct 10 times of authentication on his or her own account. Each participant will register and authenticate according to the user manual. The details of the participants are shown in Table 6.

5.1.3 Experimental Investigation. We design a questionnaire and distribute it to each participant to collect data that could not be easily measured by the software, such as users' preferences. The main questions are as follows:

- Does your phone support pressure sensing?
- Would you be willing to use handwriting identification on the phone?
- Would you be willing to use handwriting identification on the computer?
- Would you be willing to use handwriting identification on smart locks?
- Would you be willing to use handwriting identification at a pocket bank?

Fig. 5. Performance evaluation of HandiText.

5.2 Experimental Results and Analysis

We mainly analyze the practical performance of our scheme from the successful authentication rate and time of user registration and authentication, devices used by user, the usage preference, and impersonation authentication success rate.

5.2.1 Success Authentication Rate. In Section 4, we theoretically proved that the incremental algorithm can improve the accuracy of the result. Actually, we also recruited 324 participants to calculate their probability of being successfully authenticated in the actual situation.

Three hundred twenty-four participants were asked to register and authenticate. Each person would be required five authentication operations, and we counted the number of people who were ultimately successful or failed. If the authentication operation fails once in 5 times, then the participant is deemed to have failed the authentication. As shown in the Table 6, 324 participants have a total of $324 \cdot 5 = 1,620$ authentications, and the successful authentication rate is 100%.

5.2.2 Time. We record the time cost of participants during the registration and authentication phases. As shown in Figure 5(a), there is not much difference between the registration time of new users and experienced users and the authentication time. The average registration time for the first four times is 33.4309s, the average registration time for the fifth and after the fifth is 27.8102s, the average authentication time for the first four times is 4.8779s, and the average authentication time for the fifth and after the fifth is 1.4094 s. Subsequent registration and authentication times remain stable, as shown in Figure 5(a).

5.2.3 Devices. Because whether the screen of the device used by the user contains pressure test may affect our experiment result, we did not deliberately stipulate the use of the user's mobile phone brand and model. After the experiment, we conduct an investigation on whether users' mobile phones support pressure sensing, and the result is shown in Figure 5(b).

Of the participants' mobile phones, 76.8519% support pressure collection, but 23.1481% do not support pressure collection. After further investigation, it is found that these participants' mobile phone models were all released before 2016. We believe that this situation can be improved over time. In addition, according to the feedback results of users, the influence of whether the mobile phone supports pressure collection can be ignored.

5.2.4 Preference. We also conduct an investigation on users' usage preferences. We set up several different scenarios to investigate whether users will be willing to apply our behavior-based and biometric handwriting authentication scheme to these scenarios. The result is shown in Figure 5(b).

For personal digital products such as mobile phones and computers, more than half of the participants chose this scheme. For smart locks, some of the participants have a smart home, which can protect the property in the home. Therefore, more than half of the participants choose this. For

Table 7. Compared with Other Schemes

Scheme	Detail	Accuracy	Usability
TFST [37]	Four fingers writing	98%	×
TFST [37]	Three fingers writing	92.3%	×
Humm et al.'s scheme [13]	Spoken Signatures	98.9%	×
Humm et al.'s scheme [13]	Spoken Handwritings	99.7%	×
TexCaps [15]	—	90.46%	✓
Our Scheme: HandiText	—	99.06%	✓

pocket bank security, most of participants choose to use this scheme. Therefore, for places with high security requirements, users have a strong sense of security, which also proves the practical performance of our handwritten authentication scheme HandiText.

5.2.5 Impersonation Successful Authentication Rate. We selected 20 participants from the 324 participants recruited to imitate one person's handwriting and attempted to authenticate, and the results are shown in Table 6.

Each imitator can arbitrarily select the handwriting of the remaining 5 participants for imitation, and each time the authentication can be performed 5 times, so 20 imitators will generate a total of $20 \times 5 \times 5 = 500$ imitation records. As shown in Table 6, 20 imitators have failed to imitate, and the imitation authentication success rate is 0%, which proves that HandiText has strong security, and it is difficult to imitate others' handwriting.

5.3 Compared with Other Schemes

We compared our scheme HandiText with several other handwritten authentication schemes, as shown in Table 7.

Compared with the TFST scheme [37], because the TFST requires four fingers to operate on the screen at the same time, the number of features acquired is large, and the accuracy of the scheme is 98%. When the screen is small enough to accommodate four fingers, the accuracy of the program will decrease, and the accuracy with three fingers will be 92.3%. Therefore, under the premise of one-finger operation, the scheme can achieve accuracy of 99.06%, and the security is better than TFST [37].

Humm et al.'s scheme [13] requires the collection of handwriting and voice, so an additional hardware device microphone is needed to collect the voice, which is not only costly but also less usable. HandiText only needs to collect the user's handwriting, which greatly enhances the usability.

TexCaps [15] can use a small amount of data to produce large amounts of ambiguous data and then use that generated dataset for training. Because the generated data are ambiguous, its accuracy is 90.46%. TexCaps's idea of generating large amounts of data through small amounts of data is worth learning, but the accuracy still needs to be improved compared to our 99.06%.

6 FUTURE WORK AND CONCLUSION

6.1 Future Work

This article is based on the dynamic characteristics people's handwriting to design a handwriting recognition system to protect people's data and privacy security. The user's dynamic characteristics and time constraints make it difficult for attackers to imitate. In practical applications, HandiText can not only be used for login authentication but also has many other uses, such as being applied to electronic signatures, which can identify whether the signer is the user him- or herself by capturing the dynamic characteristics. Because in HandiText users mostly write the

same word, in the future we can expand the system to allow users to write different words to expand the application scenarios of HandiText.

In the future, we will focus on improving the security of HandiText. Even though the accuracy of HandiText is now very high, achieving 99.06%, we can start with the complexity of the features and use more features to make HandiText more secure, such as the gestures when users raise their hands. While ensuring the security of HandiText, we will also improve the performance of the system, such as reducing the model training time. We can improve the algorithm to reduce the time for model training, increase the accuracy, and improve the generalization at the same time.

6.2 Conclusion

In this article, we propose a handwriting recognition authentication scheme named HandiText based on behavior and biometrics features. When people write a word by hand, HandiText captures their static biological features and dynamic behavior features during the writing process, such as speed and pressure. Even if an attacker obtains the specific word written by the user through a shoulder-surfing attack, smudge attack, or screen recording attack, due to the unique dynamic features of the user and the limitation of the writing time, such as the speed and pressure of different writing points within 3 s, the handwriting is more more difficult for the attacker to imitate. Therefore, HandiText can resist shoulder-surfing attack, smudge attack, and screen recording attack. The experimental results also show that HandiText is safer than traditional solutions and has a good performance of usability. So it can better guarantee the security of data in IoT devices.

REFERENCES

- [1] almosthuman2017. 2018. AI Era No Absolute Security: Baidu Mystery Lab Minutes to Crack Iris Recognition and Vein Recognition Hardware. Retrieved from <https://36kr.com/p/5116996>.
- [2] Amusi. 2019. Biometric Authentication Under Threat: Liveness Detection Hacking. Retrieved from <https://cloud.tencent.com/developer/article/1484902/>.
- [3] D. Bertolini, L. Oliveira, E. Justino, and R. Sabourin. 2008. Ensemble of classifiers for off-line signature verification. In *Proceedings of the IEEE International Conference on Systems*.
- [4] Zhen Li Danyan Han, Jingwen Wang and Hao Li. 2007. Comparative studies on handwriting features of Chinese and English scripts. *Crim. Techn.* 4 (2007), 16–18. DOI : [10.16467/1008-3650.2007.04.006](https://doi.org/10.16467/1008-3650.2007.04.006)
- [5] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The tangled web of password reuse. In *Proceedings of the Network and Distributed System Security Conference (NDSS'14)*, Vol. 14. 23–26.
- [6] Li Deng. 2012. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Sign. Process. Mag.* 29, 6 (2012), 141–142.
- [7] S. Dreiseitl and L. Ohnomachado. 2002. Logistic regression and artificial neural network classification models: A methodology review. *J. Biomed. Inf.* 35, 5 (2002), 352–359.
- [8] Nitin Garg, Raghav Kukreja, and Pitambar Sharma. 2013. Revisiting defences against large scale online password guessing attacks. *Int. J. Sci. Res. Publ.* 3, 4 (2013).
- [9] Felix Gers and Douglas Eck. 2001. Applying LSTM to time series predictable through time-window approaches. In *Proceedings of the International Conference on Artificial Neural Networks*.
- [10] Golnaz Ghiasi and Reza Safabakhsh. 2013. Offline text-independent writer identification using codebook and efficient code extraction methods. *Image Vis. Comput.* 31, 5 (2013), 379–391.
- [11] Youn Hee Gil, Yongwha Chung, Dosung Ahn, Jihyun Moon, and Hakil Kim. 2001. Performance analysis of smart card-based fingerprint recognition for secure user authentication. In *Proceedings of the IFIP Conference on Towards the E-society: E-commerce*.
- [12] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. 2016. Smart locks: Lessons for securing commodity internet of things devices. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 461–472.
- [13] Hennebert Jean Ingold Rolf Humm, Andreas. 2008. Combined handwriting and speech modalities for user authentication. *IEEE Trans. Syst.* 39, 1 (2008), 25–35. DOI : [10.1109/TSMCA.2008.2007978](https://doi.org/10.1109/TSMCA.2008.2007978)
- [14] IT168. 2018. iPhone Has Taken the Fight to 3D Printing by Easily Cracking Android's Face Recognition Feature. Retrieved from <https://baijiahao.baidu.com/s?id=1620243794505991223&wfr=spider&for=pc>.

- [15] Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Jathushan Rajasegaran, Suranga Seneviratne, and Ranga Rodrigo. 2019. TextCaps: Handwritten character recognition with very small datasets. In *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV'19)*.
- [16] Xiaoguang Jia. 2006. A comparative study on the identification of chinese and english signatures. *J. Chin. People's Publ. Secur. Univ. Sci. Technol.* 12, 3 (2006). DOI : [10.3969/j.issn.1007-1784.2006.03.004](https://doi.org/10.3969/j.issn.1007-1784.2006.03.004)
- [17] Taekyoung Kwon and Sarang Na. 2014. TinyLock: Affordable defense against smudge attacks on smartphone pattern lock systems. *Comput. Secur.* 42, 4 (2014), 137–150.
- [18] Taekyoung Kwon, Sooyeon Shin, and Sarang Na. 2017. Covert attentional shoulder surfing: Human adversaries are more powerful than expected. *IEEE Trans. Syst. Man Cybernet. Syst.* 44, 6 (2017), 716–727.
- [19] Cheng Lin Liu, Fei Yin, Da Han Wang, and Qiu Feng Wang. 2011. CASIA online and offline chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition*, Vol. 1. 37–41. DOI : [10.1109/ICDAR.2011.17](https://doi.org/10.1109/ICDAR.2011.17)
- [20] Johnny Long. 2011. *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*. Syngress.
- [21] U. V. Marti and H. Bunke. 2002. The IAM-database: An english sentence database for offline handwriting recognition. *Int. J. Doc. Anal. Recogn.* 5, 1 (2002), 39–46.
- [22] Kenrick Mock, Bogdan Hoanca, Justin Weaver, and Mikal Milton. 2012. Real-time continuous iris recognition for authentication using an eye tracker. In *Proceedings of the ACM Conference on Computer & Communications Security*.
- [23] Hyeonjoon Moon. 2004. Biometrics person authentication using projection-based face recognition system in verification scenario. In *Proceedings of the 1st International Conference on Biometric Authentication (ICBA'04)*.
- [24] Gang Pan, Zhaohui Wu, and Lin Sun. 2008. Liveness detection for face recognition. In *Recent Advances in Face Recognition*. IntechOpen.
- [25] P. J. Phillips, K. W. Bowyer, and P. J. Flynn. 2007. Comments on the CASIA version 1.0 iris data set. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 10 (2007), 1869.
- [26] Alain Rakotomamonjy. 2003. Variable selection using svm based criteria. *J. Mach. Learn. Res.* 3, 7–8 (2003), 1357–1370.
- [27] Runye. 2018. Fingerprint Unlock Can Also Be Broken. Retrieved from <http://baijiahao.baidu.com/s?id=1601780706643320115&wfr=spider&for=pc>.
- [28] Stefan Schneegass, Frank Steimle, Andreas Bulling, Florian Alt, and Albrecht Schmidt. 2014. Smudgesafe: Geometric image transformations for smudge-resistant user authentication. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 775–786.
- [29] Sandeep K. Sood, Anil K. Sarje, and Kuldip Singh. 2009. Cryptanalysis of password authentication schemes: Current status and key issues. In *Proceedings of the 2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS'09)*. IEEE, 1–7.
- [30] Sandeep K. Sood, Anil K. Sarje, and Kuldip Singh. 2010. Cryptanalysis of password authentication schemes: Current status and key issues. In *Proceeding of the International Conference on Methods & Models in Computer Science*.
- [31] JC Torres. 2019. Android Q Gets “3D Touch” Pressure-Sensitivity Support. Retrieved from <https://www.slashgear.com/android-q-gets-3d-touch-pressure-sensitivity-support-08572355/>.
- [32] Liam Tung. 2017. IoT Devices Will Outnumber the World's Population This Year for the First Time. Retrieved from <https://www.zdnet.com/article/iot-devices-will-outnumber-the-worlds-population-this-year-for-the-first-time/>.
- [33] Ding Wang and Ping Wang. 2015. Offline dictionary attack on password authentication schemes using smart cards. In *Information Security*. Springer, 221–237.
- [34] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. 2016. Understanding password choices: How frequently entered passwords are re-used across websites. In *Proceedings of the 12th Symposium on Usable Privacy and Security (SOUPS'16)*. 175–188.
- [35] G. O. Williams. 2002. Iris recognition technology. *IEEE Aerosp. Electr. Syst. Mag.* 12, 4 (2002), 23–29.
- [36] Ying Xin. 2018. Research and implementation of handwriting recognition system based on kNN. *Electronic Design Engineering* 26, 7 (2018), 27–30. DOI : [10.14022/j.cnki.dzsjgc.2018.07.007](https://doi.org/10.14022/j.cnki.dzsjgc.2018.07.007)
- [37] Y. Song, Z. Cai, and Z. Zhang. 2017. Multi-touch authentication using hand geometry and behavioral information. In *2017 IEEE Symposium on Security and Privacy (SP'17), San Jose, CA, 2017*. 357–372. DOI : [10.1109/SP.2017.54](https://doi.org/10.1109/SP.2017.54)

Received August 2019; revised January 2020; accepted February 2020