SECURING SMARTPHONE HANDWRITTEN PIN CODES WITH RECURRENT NEURAL NETWORKS

Gaël Le Lan *Vincent Frey*

Orange Labs, France

{gael.lelan, vincent.frey}@orange.com

ABSTRACT

This paper investigates the use of recurrent neural networks to secure PIN code based authentication on smartphones, in a scenario where the user is invited to draw digits on the touchscreen. From the sequence of successive positions of the users finger on the touchscreen, a bidirectional recurrent neural network computes a discriminative embedding in terms of writer traits, carrying the contextual information of the written digit. This allows to reject impostors who would have knowledge of the PIN code. The neural network is trained to recognize both users and digits of a training dataset. Evaluations are run on two datasets of 43 and 33 users, respectively, absent from the training dataset. Results show that when enrolling the users on 4 examples of each digit, the Equal Error Rate reaches 4.9% for a 4-digit PIN code. Including digit value prediction during training is key to achieve good performances.

***Index Terms*—** behavioral biometrics, recurrent neural network, writer verification, authentication.

1. INTRODUCTION

Smartphones are nowadays very popular devices. As such, many companies offer services or applications requiring a secure access. Authentication usually relies on a knowledge factor, a secret, such as a password or PIN (*Personal Identification Numbers*) code. However, an impostor with the knowledge of the secret can access the protected service [1]. Some accesses may be protected by a biometric sensor; however, sensors do not replace secrets: an impostor who cannot authenticate through a biometric sensor will usually be prompted for a secret after a few authentication failures.

To enhance security, behavioral biometrics is considered as a new mean of authentication [2, 3, 4, 5, 6]. Recently, behavioral biometrics was used to enforce security challenges such as password prompts, as a second authentication factor [7, 8, 9]. In [7], the authors proposed to enforce the secret path security with an analysis of the drawing style. Using a method based on Pearson correlation coefficients and Dynamic Time Warping (DTW), an Equal Error Rate (EER) of 17% was obtained on dataset of 15 users. [9] proposed a One-Time-Password (OTP) scenario where users are invited to perform finger-drawn PIN codes on a touchscreen. In addition to the validity of the code, the user is authenticated through the way he/she interacts with the touchscreen. The method was evaluated on 43 users, achieving 9.3% EER for a 4 distinct digits PIN code, using feature selection and DTW.

In the field of behavioral biometrics, recurrent neural networks (RNNs) were explored in the last years [10, 11, 12]. RNNs are well adapted to model temporal sequences and have been used with success for English and Chinese writer identification [11], with 99% accuracy among 150 users, using a bidirectional RNN based on Long Short Term Memory (LSTM) Cells [13]. In another domain, RNNs were used to authenticate handwritten signatures [12]. In their work, the authors proposed a siamese architecture based on RNNs to compare pairs of signatures. On an evaluation dataset of 100 users, verification performances reach 5.5% EER, the system being trained on 300 distinct users.

In this article, we propose to use RNNs for the task of writer verification, in a finger-drawn PIN code authentication scenario. The proposed approach is related to that of [11]. However, in our work, writer verification is performed on single handwritten digits, acquired through a smartphone or tablet touchscreen, instead of being done on paragraphs of sentences written with a connected pen. Experiments are run on the *eBioDigit* [9] dataset, completed by an *internal* dataset. This article is structured as following. In a first section, we present the authentication system and the use case. In the second, we describe the proposed method, based on RNNs. The third section is dedicated to the experiments, followed by a discussion about the results.

2. AUTHENTICATION SYSTEM

The proposed authentication system is a smartphone application where the user is asked to draw digits, one after the other. To be able to authenticate himself/herself, the user has been previously enrolled on the device on the 10 digits. The enrollment phase consists in performing and storing 4 examples of each handwritten digit in a local template database. During the authentication phase, the user is asked to enter a PIN code of 4 successive distinct digits. Authentication is performed on the device itself, on the validity of the PIN code and on

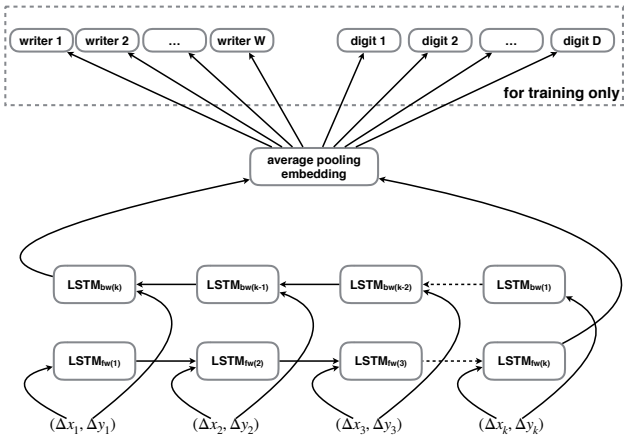

Fig. 1. Bidirectional recurrent neural network architecture.

the way the user writes the digits: if the writer traits are not recognized, the user is rejected even if the code is valid.

To decide whether the user is genuine or an impostor, each handwritten digit of the code is compared to those stored in the template and a confidence score is computed. For a 4 digits PIN code, the 4 scores are averaged and compared to a threshold. In this article, we focus on the writer verification task: we only consider the case where all impostors have knowledge of the PIN code of the genuine user, and where the only way to reject them is to analyse their writer traits.

3. ARCHITECTURE

To model writer traits, we propose to use a representation extracted by a RNN [14]. RNNs are widely used for speech or handwriting recognition, because they are designed to model variable length sequences in a fixed size representation (a vector). Thus, sequences modeled by a RNN can be easily compared through that representation.

When a user writes a digit, the smartphone collects a sequence of vectors $\boldsymbol{\tau}_i = (x_i, y_i)_{i\in[1..N]}$, describing the position of the finger on the touchscreen. x_i and y_i are the coordinates of the finger on the touchscreen plan. The number of collected points N is different for each sequence: its total duration is variable and the sampling frequency of the touchscreen is not fixed. In an approach similar to that of [11], we transform that input data in a sequence $\boldsymbol{S}$ of strokes $\boldsymbol{s}_i = (\Delta x_i, \Delta y_i)_{i\in[1..N-1]}$, where $\Delta x_i = x_{i+1} - x_i$ and $\Delta y_i = y_{i+1} - y_i$. This makes the data independent of the absolute position of the drawing on the screen, while being suitable to represent its size and dynamics.

Here, we propose to estimate, through a recurrent neural network, a non linear function f to extract a vector $f(\boldsymbol{S})$, which embeds the value of the handwritten digit and the writer's identity. In the neural network architecture, we force that vector to the unit sphere, so that different input sequences can be compared with cosine similarity. For example, the similarity between two stroke sequences $\boldsymbol{S}$ and $\boldsymbol{Q}$ is described in equation 1.

Table 1. Datasets composition.

name	$eBioDigit$ [9]		$internal$	
device	Samsung Galaxy Note 10.1		Samsung Galaxy A8	
type	*train*	*eval*	*train*	*eval*
users	50	43	29	33
sessions/user	2		1	2
digit/session	10×4		10×5	

$$sim(\boldsymbol{S}, \boldsymbol{Q}) = \frac{f(\boldsymbol{S})f(\boldsymbol{Q})}{\|f(\boldsymbol{S})\|\|f(\boldsymbol{S})\|} \tag{1}$$

The network architecture is presented in figure 1. It is a bidirectional RRN [15] composed of LSTM cells [13]. The bidirectional structure is designed for the model to consider both past and future elements of the input sequence. In the forward (resp. backward) layer, the sequence is presented in the chronological (resp. anti-chronological) order. The outputs of the forward and backward layers are averaged to form the embedding.

For training only, the embedding layer is fully connected to two distincts softmax layers: one for writer prediction, the other for digit value prediction. The RNN is trained to recognize the value of the digit and the identity of each input stroke sequence's writer, by minimizing a cross entropy loss function with gradient back-propagation [16]. The cost function is that of equation 2, where W represents the ensemble of writers $(w)_{w\in[1..W]}$ and D the ensemble of digits $(d)_{d\in[1..D]}$. The operator $\mathbb{1}_{[\boldsymbol{S}\in w]}$ equals 1 when $\boldsymbol{S}_k$ was written by w, 0 otherwise. $\mathbb{1}_{[\boldsymbol{S}\in d]}$ works likewise for digit values. In this article, D represents the digits from 0 to 9, but could correspond to any form of symbol or graphical representation.

$$\mathcal{L}(\mathcal{T}) = -\frac{1}{M}\sum_{k=1}^{M}\left[\sum_{w=1}^{W}\mathbb{1}_{[\boldsymbol{S}_k\in w]}\log(p_{\boldsymbol{S}_k\in w}) + \sum_{d=1}^{D}\mathbb{1}_{[\boldsymbol{S}_k\in d]}\log(p_{\boldsymbol{S}_k\in d})\right] \tag{2}$$

Optimal parameters of the neural network are estimated through exhaustive search and early stopping. The chosen embedding dimension is of 1024, while the chosen optimizer is *Adam* [17] and the RNN is implemented with *Keras* [18] and *Tensorflow* [19]. Once the RNN has been trained, both softmax layers are removed and the resulting network is only used to extract handwritten digit embeddings. Embedding extraction is performed: offline for evaluation, online in the considered authentication application (i.e. on the device itself).

Table 2. System performances on various *train/eval* combinations.

#	*system*	*train*	*train sessions count*	*eval*	*EER*			
					RNN trained with digit prediction		*without*	
					1 digit	*4 digits*	*1 digit*	*4 digits*
1	[9]	$eBioDigit_{train}$	4000	$eBioDigit_{eval}$	18.6	9.3	-	-
2	*proposed*				15.1	6.5	20.9	11.3
3				$internal_{eval}$	18.5	8.2	23.1	13.2
4		$eBioDigit_{train} + internal_{train+eval}$	8 750	$eBioDigit_{eval}$	**12.5**	**4.9**	**18.0**	**9.9**
5		$eBioDigit_{train+eval} + internal_{train}$	8 890	$internal_{eval}$	15.8	6.3	22.7	11.8

4. EXPERIMENTAL SETUP

4.1. Datasets

To evaluate authentication based on writer traits, we use a *train* dataset to train the RNN. Afterwards, the RNN is used as an embedding extractor and its performances are evaluated on an *eval* dataset. *Train* and *eval* subsets are built from two handwritten digits *corpora*: *eBioDigit* [9] and an *internal corpus*. Those datasets contain drawings of handwritten digits, with writer identity and digit value annotations. A drawing consists in sequences of positions (x, y) of the users' finger on the touchscreen. Digits were either collected on a tablet (*eBioDigit*) or a smartphone (*internal*). For a same user, there can be 2 sessions of collection, which were at least 2 weeks apart. A session contains 4 or 5 examples of the 10 digits, depending on the dataset.

Each dataset is split into a *train* and an *eval* subset, which contain distinct users. Zero mean and unit variance normalization is applied to the ensemble of stroke sequences of each dataset independently, so that data from both datasets can be jointly used for training. Depending on the considered experiment, when evaluation is performed on one of the two *eval* subset (ie. *eBioDigit* of *internal*), the other one can be used as additional data to train the RNN.

4.2. Evaluation protocol

For each user of the *eval* dataset, the first session of collected digits is used to build an enrollment template, while the second is used to simulate trials. Each sequence of strokes is passed through the RNN and the template is composed of 4 examples per digit (ie. 4 embeddings per digit). Trials are scored using a 1-nearest neighbor classifier. For a given trial, its embedding is compared to those of the corresponding digit in the template and the trial-template similarity score is the cosine similarity between the trial embedding and the closest template embedding. To evaluate the system, we distinguish genuine and impostor tests. A genuine (resp. impostor) test consists in scoring a trial-template pair representing the same (resp. a different) user. The system is evaluated with the EER (the lower the better), computed over all possible trial-template tests of the *eval* dataset.

We consider two authentication scenarios. The first (resp. second) one consists in authenticating the user on one single digit (resp. a 4 distinct digit PIN code). In the chosen authentication scenarios, impostors do not have any *a priori* knowledge of the genuine writer traits. They just know what digits they have to enter to authenticate, as if they had stolen the PIN code. A contrastive experiment consists in removing the digit prediction layer during RNN training, to quantify how this extra contextual information helps.

5. RESULTS

Table 2 shows EERs obtained for various experiments. Multiple dataset combinations are used to train the neural network and evaluate the system. Whatever the combination, no user is present in both *train* and *eval* datasets. For example, in experiment #3, the neural network was trained on a combination of $eBioDigit_{train}$, $eBioDigit_{eval}$ and $internal_{train}$ datasets and evaluated on $internal_{eval}$. The table also shows the number of training sessions per experiment, and presents the results of the contrastive RNN training configuration, where it is trained without the digit prediction layer. Finally, we report the results of [9] (experiment #1 of the table), in comparable experimental conditions.

5.1. Comparison with previous state of the art

In the experimental conditions of [9], we note that using RNNs (exp. #2) outperforms the feature selection/DTW-based method (exp. #1). The EER on a single digit is reduced from 18.6% to 15.1%, while it reaches 6.5% for a 4 digits PIN code, instead of 9.3%.

Our initial RNN design was similar to that of [11] and was only trained to recognize writers, whatever the digit value. The loss expression was reduced to the first term of equation 2. We wanted the RNN to model writer traits without providing any contextual information. The system's results using that RNN configuration are reported in the last two columns of table 2. They show that without additional training data, the RNN-based method only reaches an EER of 11.3% on 4 digits (exp. #2), 2% worse than the *baseline* of 9.3%. Including digit value prediction during RNN training proves to be effective for all tested *train/test* configurations and gives a 5.8% EER improvement for experiment #2 (from 11.3% to 6.5%). This shows that including contextual data helps the RNN to model writer traits on short sequences of strokes.

In the work of [11], results showed that the RNN needed to be trained on word or sentence level sequences to effectively recognize writers. In our work, we can only train the system character by character, thus we cannot model writer traits based on between-character or between-word strokes. However, using contextual information such as the digit value helps the RNN to work with short sequences, and this addition lowers the EER for the 4 *train/eval* tested configurations.

5.2. Volume of training data

For a given *eval* dataset, results show that the RNN benefit from additional training data. For example, the EER on $eBioDigit_{eval}$ is lowered when $internal_{train+eval}$ is additionally used for training, doubling the number of training sessions (exp. # 4). On $eBioDigit_{eval}$, the EER on a 4 digits PIN code (resp. 1 digit) is reduced from 6.5% to 4.9% (resp. from 15.1% to 12.5%).

When evaluating the system on $internal_{eval}$, we note that the system does not perform as efficiently as on the $eBioDigit_{eval}$ dataset. Using $eBioDigit_{train}$ only for training, the EER on 4 digits (resp. 1 digit) is 1.7% (resp. 3.4%) higher (exp. #3). Performances obtained on $internal_{eval}$ also benefit from additional training data, since the EER is lowered to 6.3% when doubling the number of training sessions (exp. #5). This gap of performances is probably due to a mismatch between *train* and *eval* datasets. In experiment #5, most of the *train* data is from ***eBioDigit*** (around 84% of the *train* sessions, not shown in the table). Since both datasets were collected on different devices, touchscreen sampling or resolution can vary, even if we applied normalization.

Table 3. Performances per digit, whether examples of the digit are included in the *train* dataset or not.

	EER	
digit	$\in train$	$\notin train$
0	15.3	17.0
1	14.5	15.1
2	17.4	17.0
3	15.2	18.0
4	13.4	12.2
5	12.5	12.8
6	16.9	18.6
7	12.5	12.8
8	12.5	13.8
9	13.0	13.4
avg	14.3	15.1

5.3. Expanding to other symbols

The proposed system is designed to authenticate users based on handwritten digits. Building it relies on collecting annotated data to train the RNN. When we collected our *internal* data, it was acceptable for people to participate because it was fast: performing 50 handwritten digits only takes a few minutes. When we were asked if the system would work for any kind of characters or symbols, it became far less acceptable to collect such data. To do so, it would require to collect new symbols, while still being able to authenticate users on those.

To evaluate that aspect, we propose to evaluate the system on a new contrastive experiment: the RNN is trained on 9 digits only. We exclude the one we want to evaluate the system on, as if it was a new symbol. The RNN will be used to extract embeddings for digits it has never seen during training. The system is trained on $eBioDigit_{train}$ and evaluated on $eBioDigit_{eval}$. Results are presented in table 3. In the first column, the RNN is trained in the same configuration of that of experiment #2 of table 2, but with 10% removed sessions. In the second one, we remove all examples of a specified digit from the training dataset. In both cases, the total number of training sessions per writer is identical.

Results show that the system usually performs better when the evaluated digit is included in the *train* dataset, except for digits 2 and 4. In average, we note a 0.8% EER gap between the two conditions: the system is still able to perform correctly on unseen symbols, the EER being only degraded of 2.8% in the worst case (digit 3). This shows that authenticating users on unseen digits or symbols is possible: the RNN is able to model writer traits that are common between symbols.

6. CONCLUSIONS

In this article, we proposed a novel method to help securing handwritten PIN codes on smartphones. We designed a LSTM-cell based bidirectional RNN to encode any handwritten digit in a fixed compact representation, discriminative in terms of writer traits and carrying the contextual information of the written digit. The RNN is trained on an annotated dataset of smartphone users, whose handwritten digits were collected in two sessions to account for the within-writer/between-session variability. Once trained, The trained network is then used to simulate an authentication scenario: each handwritten digit can be compared to any other through cosine similarity between corresponding embeddings.

Experiments were run on two datasets from two distinct collection campaigns, and show that the proposed approach achieves 4.9% EER on the ***eBioDigit*** dataset. It outperforms state of the art, which was of 9.3% with a DTW-based approach. This gain is due both to the method itself (2.8%) and the use of additional data for RNN training (additional 1.6%). Including digit value prediction during training is key to achieve good performances, while having shown that the system is still able to discriminate writers on unseen symbols.

The authentication scenario simulated in our experiments considers that the impostors do not have any *a priori* knowledge of the way users write on their phone. Further work could include an *over the shoulder* attack scenario, where an impostor would try to mimic the way the user writes its digits. It could also be beneficial to use additional data sources such as finger pressure, accelerometer or gyroscope.

7. REFERENCES

[1] Vishal M Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[2] Roman V Yampolskiy and Venu Govindaraju, "Behavioural biometrics: a survey and classification," *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[3] Zdenka Sitova, Jaroslav Sedenka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran Balagani, "Hmog: A new biometric modality for continuous authentication of smartphone users," *arXiv preprint arXiv*, vol. 1501, 2015.

[4] Marcos Martinez-Diaz, Julian Fierrez, and Javier Galbally, "Graphical password-based user authentication with free-form doodles," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 607–614, 2016.

[5] Upal Mahbub, Sayantan Sarkar, Vishal M Patel, and Rama Chellappa, "Active user authentication for smartphones: A challenge data set and benchmark results," in *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE, 2016, pp. 1–8.

[6] Attaullah Buriro, *Behavioral biometrics for smartphone user authentication*, Ph.D. thesis, University of Trento, 2017.

[7] Michael Beton, Vincent Marie, and Christophe Rosenberger, "Biometric secret path for mobile user authentication: A preliminary study," in *Computer and Information Technology (WCCIT), 2013 World Congress on*. IEEE, 2013, pp. 1–6.

[8] Patrick Lacharme and Christophe Rosenberger, "Synchronous one time biometrics with pattern based authentication," in *Availability, Reliability and Security (ARES), 2016 11th International Conference on*. IEEE, 2016, pp. 260–265.

[9] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, and Javier Ortega-Garcia, "Incorporating touch biometrics to mobile one-time passwords: Exploration of digits," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 471–478.

[10] Natalia Neverova, Christian Wolf, Griffin Lacey, Lex Fridman, Deepak Chandra, Brandon Barbello, and Graham Taylor, "Learning human identity from motion patterns," *IEEE Access*, vol. 4, pp. 1810–1820, 2016.

[11] Xu-Yao Zhang, Guo-Sen Xie, Cheng-Lin Liu, and Yoshua Bengio, "End-to-end online writer identification with recurrent neural network," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 2, pp. 285–292, 2017.

[12] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, and Javier Ortega-Garcia, "Exploring recurrent neural networks for on-line handwritten signature biometrics," *IEEE Access*, vol. 6, no. 5128-5138, pp. 1–7, 2018.

[13] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[15] Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[16] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533, 1986.

[17] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] François Chollet, "Keras," 2017.

[19] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.