Keystroke Mobile Authentication: Performance of Long-Term Approaches and Fusion with Behavioral Profiling

Alejandro Acien(✉), Aythami Morales, Ruben Vera-Rodriguez, and Julian Fierrez

BiDA Lab, School of Engineering, Universidad Autonoma de Madrid,
C/ Francisco Tomas y Valiente 11, 28049 Madrid, Spain
{alejandro.acien, aythami.morales, ruben.vera, julian.fierrez}@uam.es

Abstract. In this paper we evaluate the performance of mobile keystroke authentication according to: (1) data availability to model the user; and (2) combination with behavioral-based profiling techniques. We have developed an ensemble of three behavioral based-profile authentication techniques (WiFi, GPS Location, and App usage) and a Keystroke state-of-the-art recognition approach. Algorithms based on template update are employed for profiling systems meanwhile bidirectional recurrent neuronal networks with a Siamese training setup is used for the keystroke system. Our experiments are conducted on the semi-uncontrolled UMDAA-02 database. This database comprises smartphone sensor signals acquired during natural human-mobile interaction. Our results show that it is necessary 6 days of usage data stored to achieve the best performance in average. The template update allows to improve the equal error rate of keystroke by a relative 20%–30% performance.

Keywords: Mobile authentication · Biometric recognition · Behavioral pattern · Behavioral-based profiling · Keystroke dynamics

1 Introduction

In the last decade smartphones have become a vital gadget for an important percentage of the world population. Recent reports reveal that mobile lines exceeded the world population in 2018 [1]. Moreover, more than 90% of citizens decline to go out without their smartphones due to the need for contact with their friends or work responsibilities among others reasons [2]. During our daily routines, smartphones become a sort of data hubs storing a wide variety of sensitive information: from personal information (e.g. photos, videos, chats messages) stored by ourselves, behavioral traits (e.g. touch gestures, GPS location, WiFi connections, keystroke patterns) stored by the smartphones during the user interaction, up to critical information (e.g. bank transactions, account's passwords, contacts list). Due to this capacity of storing sensitive information, according to [3] more than a half of population would be willing to pay 500$ and the 30% would pay up to 1000$, regardless the price of the device, in order to recover the smartphone information when stolen.

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11868, pp. 12–24, 2019.
https://doi.org/10.1007/978-3-030-31321-0_2

However, some surveys have shown that about 34% or more smartphone users did not use any form of authentication mechanism on their mobile devices [4]. Among the reasons for this, inconvenience of use was cited to be the main reason. They find out that mobile device users considered unlock screens unnecessary in 24% of the situations and they spend up to 9% of time they use their smartphone unlocking the screens despite of many modern smartphones have fingerprint and face recognition algorithms.

In order to deal with this problem of convenience of use, the research community is developing transparent biometric authentication mobile systems [5]. These approaches analyze behavioral information (e.g. touch gestures, keystroke patterns) stored by the smartphone during normal user-device interaction and check the user's identity in the background. This way, the smartphone will be able to assist in user authentication avoiding to disturb the owner with traditional authentication mechanisms (e.g. passwords, swipe patterns).

Despite some of these biometric authentication mobile systems work really well achieving a good performance under certain conditions (e.g. limited number of users, supervised scenarios), these systems are not usually tested in a real life scenario in which a new user installs the authentication system in the device and starts using it. In that moment, the amount of behavioral data available for the biometric authentication will be scarce and the performance may be low. The device will need a traditional authentication mechanism until it has enough behavioral biometric information to check the identity of the user by itself with good performance.

The aim of this paper is to analyze how the performance of these mobile biometric authentication systems evolve according to the amount of behavioral information available from the owner. Our experiments include up to four different information channels (Keystroke, GPS location, WiFi signals, and App Usage) and the fusion of all of them to train a reliable authentication system by employing each time more user's information to authenticate. Finally, we will analyze how much information these biometric systems need to work with a good performance. For this, our experiments are conducted on the UMDAA-02 mobile database [6], a challenging mobile dataset acquired under unsupervised conditions.

The rest of this paper is organized as follows: Sect. 2 makes an overview of the state-of-the-art works related and links with this work. Section 3 describes the architecture followed to implement the different systems proposed. Section 4 explains the experimental protocol, describing the database and the experiments performed. Section 4.3 presents and analyzes the results achieved and Sect. 5 summarizes the conclusions and future work.

2 Background and Related Works

Authentication systems based on keystroke dynamics have been widely studied in computer keyboards, achieving very good results in fixed text [7] (i.e. the keystroke sequence of the input authentication system is prefixed) and free text [8] (the input can be any kind of keystroke sequence). The feature set usually employed in keystroke recognition is generated using the elapsed time of press and release events between consecutive keys [9]: hold time, inter-key latency, press and release latency. In the authentication stage; Manhattan distances, DTW and digraphs achieve the best results in most of the cases for fixed text scenarios [7, 10, 11], whereas binary classifiers (SVM, KNN), Hierarchical Trees and Recurrent Neuronal Networks work better in free text [12–14].

Regarding keystroke authentication in smartphones, similar architectures have been applied with little adaptations. In [15], they take advantage of the hand postures while holding the device during typing as discriminative information, and combining this with time features they reduce the error rates up to 36.8% in a fixed text scenario with binary classifiers (SVM, NB, and KNN). In [16], Monaco *et al.* proposed Partially Observable Hidden Markov Models (POHMM) as an extension of the traditional Hidden Markov Models (HMMs), but with the difference that each hidden state is conditioned on an independent Markov chain. The algorithm is motivated by the idea that typing events depend both on past events and also on a separate process. More recently, [17] proposed a Siamese Long Short-Term Memory network architecture in which the keystroke authentication is performed by calculating the Euclidean distance between two embedding vectors (the outputs of the Siamese model).

The WiFi networks detected by our smartphone provide useful information about when and where we go, and hence, they can detect possible variations in our daily routines. This discriminative information is considered as behavioral biometric and it could help in the mobile authentication process. In this assumption, [18] explores a WiFi authentication system based on templates. They store in a template the time and the name of the WiFi networks detected during the training process, then they test by comparing the template with the new WiFi networks detected and compute a kind of confidence score.

Regarding Geo-location based authentication approaches, Mahub *et al.* [19] developed a modified HMM to characterize the mobile trace histories, they suggest that the human mobility can be described as a Markovian Motion, and they predict the new user location exploiting the sparseness of the data and past locations. In [13], they classify mobile user location with SVM by using the latitude and longitude as features and calibrating the scores with logistic regression. They also implement App Usage based authentication by ranking the top 20 mobile applications most visited by the user that appear in the training set. The classification process is performed by comparing these top ranks of more used applications with the new test data and calculating a similarity score. In the other hand, [20] suggests that the unknown applications and unforeseen events have more impact in App Usage authentication than the top N-apps, and they should be incorporated in the models by adopting smoothing techniques with HMMs. They are capable of detecting an intrusion in less than 3 min of application usage with only 30 min of historical data to train.

Finally, how to integrate all these different modalities in a multimodal mobile authentication architecture is not trivial [21]. Due to many differences between the architectures proposed for each biometric trait, the fusion is usually done at decision level. For example in [13], they fused at decision level web browsing, application usage, GPS location, and keystroking data using information from slice time windows. They suggest that the performance increases according to the size of the time window. In [22], they merge also at decision level touch dynamics, power consumption, and physical movements modalities with a dataset captured under supervised conditions. In

[17], they merge up to 8 modalities (keystroke dynamics, GPS location, accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, and rotation sensors) at score level with a Siamese Long Short-Term Memory network architecture and 3 s window. The fusion approach enhances the performance more than 20% compared to each modality separately. In [23], they designed a mobile authentication app that collects data from WiFi, Bluetooth, accelerometer, and gyroscope sources during natural user interaction and fused them at score level achieving up to 90% of accuracy in the best scenario.

Previous works fusing different modalities [13, 17, 22, 23] have focused their approaches on obtaining time windows from the different modalities and then, they carry out the fusion with the architectures previously trained for each user. However, this does not represent a realistic scenario because biometric information is not always available at the beginning and therefore, the lack of these biometric information could decrease the performance.

The major contributions of this paper are: (i) a performance analysis of user mobile authentication based on keystroke biometrics traits and 3 behavioral-based profiling techniques (GPS location, WiFi, and App usage) separately and the fusion of all them at score level for a multimodal approach, and (ii) a study of the performance evolution of these authentication systems across the time according to the amount of user biometric information available in each moment.

3 Systems Description

In this paper we will analyze 4 mobile sources of information: Keystroking, GPS Location, App Usage, and WiFi. According to the literature, keystroke patterns are related to the neuromotor skills of the people based on, for instance, time differences between consecutive keys, which are directly related to muscles activation/deactivation timing [24]. On the other hand, GPS location, WiFi, and App Usage belong to behavioral based-profiling systems that describe daily habits and manners from the user according to the services they use or the places they visit [5]. In the next subsection we describe the approach followed for each of the 4 systems taking into account the above definitions.

3.1 Keystroke System

In keystroking, the discriminative user information is allocated in the temporal relationships of press and release events between two or more consecutive keys. For this reason, we decided to implement a Recurrent Neural Network (RNN) algorithm for keystroking authentication. To the best of our knowledge, RNN has demonstrated to be one of the best algorithms to deal with temporal data and works well with free-text keystroke patterns [14, 17]. The feature set chosen is as follows: (i) Hold Latency (HL): the elapsed time between press and release key events; (ii) Inter-key Latency (IL): the elapsed time between releasing a key and pressing the next key; and (iii) Press Latency

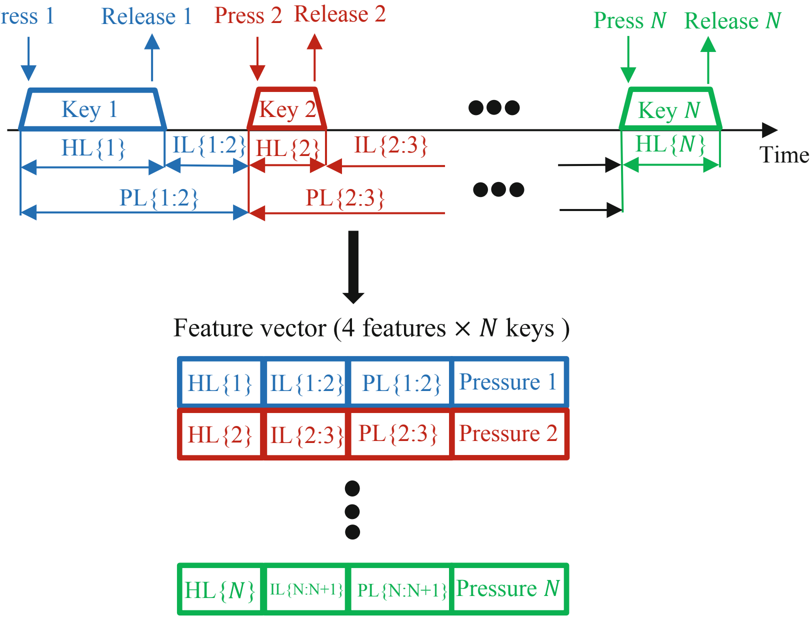

Fig. 1. Example of feature extraction for a keystroke sequence of N keys. The number in brackets shows the key corresponding to each feature.

(PL): the elapsed time between two consecutive press events. Additionally, we add the pressure as another feature to provide more behavioral user information (see Fig. 1 for details).

Our RNN model has a fixed length input N. To handle keystroke sequences of varying length, we concatenate them until we have the length necessary to feed the RNN, as proposed in [17]. The longer sequence we choose, the better performance the RNN model usually achieves. However, the system has to wait until the user has pressed enough number of keys to authenticate the user. So there is a trade-off between the performance and the authentication time delay.

The architecture of the RNN model that achieved our best results is depicted in Fig. 2. That RNN consists of two LSTM layers of 32 units with batch normalization and dropout rate of 0.5 between layers to avoid overfitting. We suggest that the next keys typed are as relevant as past keys, therefore, in order to consider forward and backward time relationships between consecutive keys, we decided to set up the LSTM layers in a bidirectional mode (duplicating the number of neurons in each layer, one for each forward and backward direction). The output of the RNN model is an embedding vector of 64 units' size (32×2), this embedding vector is a feature representation of the input keystroke sequence that we will use to distinguish a keystroke sequence from genuine and impostor users. By training the RNN model in a Siamese setup, the RNN model will learn discriminative information of the keystroke sequence and transform this information into an embedding space where keystroke sequences of the same user (genuine samples) are close, and far in the opposite case.

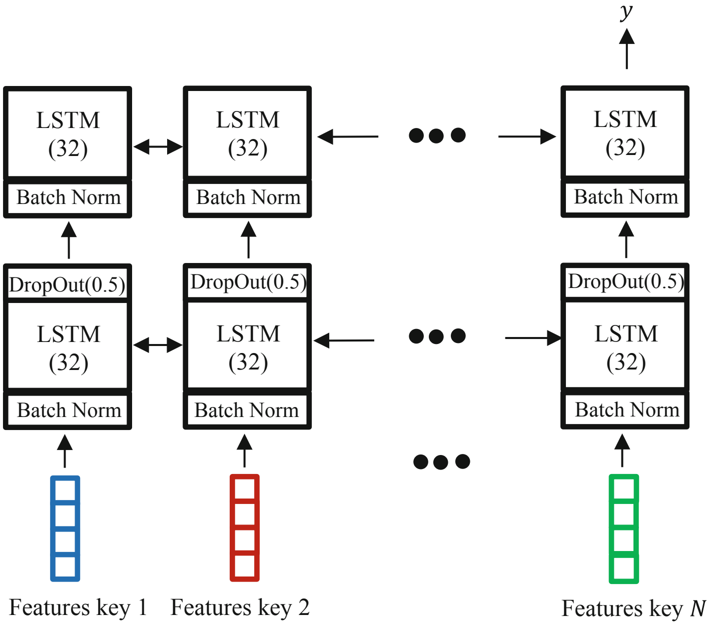

Fig. 2. Architecture of the Bidirectional RNN model proposed. The output of the model y is a embedding vector of 64 (32×2) features (the bidirectional mode duplicates the number of neurons in each layer).

In this setup, the RNN model has two inputs (the two keystroke sequences to compare) and outputs two embedding vectors (see Fig. 3 for details). By calculating the Euclidean distance between this pair of embedding vectors we will obtain a score between 0 and α, where 0 means that both keystroke sequences belong to the same user and α means that they come from different users.

For this, the contrastive loss is defined to regulate large or small distances depending on the label y_{ij} associated with the pair of samples [17]. Let's define X_i and X_j as both inputs of the Siamese model, the Euclidean distance between the pairs $d(X_i, X_j)$ is defined as:

$$d(X_i, X_j) = \left\| f(X_i) - f(X_j) \right\| \quad (1)$$

where $f(X_i)$ and $f(X_j)$ are the outputs (embedding vectors) of the RNN Model. Finally, with the contrastive loss, the RNN model will learn to make this distance small for genuine pairs and large for impostor pairs according to the label y_{ij}:

$$Loss = (1 - y_{ij}) \frac{d^2(X_i, X_j)}{2} + y_{ij} \frac{\max^2(0, \alpha - d(X_i, X_j))}{2} \quad (2)$$

where the label y_{ij} is set to 0 for genuine pairs and 1 for impostor pairs and $\alpha > 0$ is called the margin (the maximum margin between genuine and impostor distances).

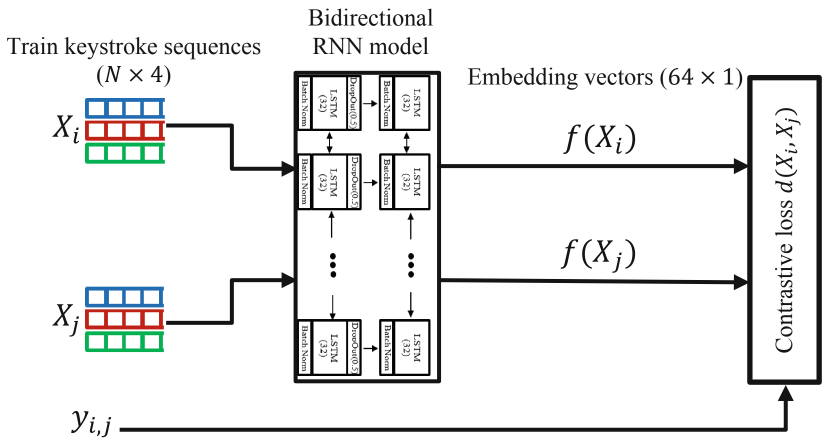

Fig. 3. Siamese keystroke setup for training. N is the number of keys in each sequence.

3.2 Behavioral-Based Profiling Systems

For WiFi, GPS location, and App Usage behavioral-based profiling systems we employ a template-based matching algorithm that has proved to work well according to [18, 20]. This algorithm consists in user's templates that record the time stamps and the frequency of the events occurred during the daily routines of the user. These events are the WiFi networks detected, the latitude and longitude of a location or the name of the app for WiFi, GPS location, and App Usage systems respectively.

Table 1 shows an example of a template for the WiFi system. First of all, we divided the 24 h of the day in M time slots of fixed duration. For example, for $M = 48$ we will have 48 slots of 30 min length (24/48 = 0.5 h = 30 min). Once the size of the time slots is set, the template records for each WiFi network the time slot it belongs and the name of the network. The frequency column shows the number of sessions that WiFi network was detected in the same time slot. In other words, the template-based algorithm describes the daily routines of the user during a period of time according to the events detected by their smartphone when he/she unlocks the smartphone.

Table 1. Example of a WiFi user template generated according the data captured during a week.

Event (WiFi network)	Time slot	Frequency
Network 1	4	7
Network 2	10	3
Network 3	10	1
Network 1	15	7
Network 4	24	5

Table 2. General UMDAA-02 dataset information.

Description	Statistics
Gender	36M/12F
Age	22–31 years
Avg. Days/User	10 days
Avg. Sessions/User	248 sessions
Avg. Sensors/Session	5.2 Sensors
Avg. Sessions/Day	26 sessions

Finally, we test the system by comparing the new sessions with the user template. We match the new events detected with the events of the template for each time slot and calculate a confidence score as:

$$score = \sum_{i=1}^{S} f_i^2 \tag{3}$$

where f_i is the frequency of the event stored in the user template that matches with the test event i in the same time slot and S is the total number of events detected in that test session. For example, if the test session includes the WiFi networks of '*Network 2'* and '*Network 3'* during the tenth time slot, the score confidence will be $1^2 + 3^2 = 10$ (according to the template showed in Table 1). Based on this, a higher score in the test session implies higher confidence for authentication.

4 Experiments

4.1 Database

The experiments were conducted with the UMDAA-02 database [6] that comprises more than 140 GB smartphone sensor signals collected during natural user-device interaction. Table 2 summarize the characteristics of the database. The users were mainly students from the university of Maryland, they used a smartphone provided by the researchers as their primary device during their daily life (unsupervised scenario) over a period of two months. A huge range of smartphone sensors were captured: touchscreen (i.e. touch gestures and keystroking), gyroscope, magnetic field, GPS location, and WiFi networks, among others. Information related to mobile user's behavior like lock and unlock time events, start and end time stamps of calls, and app usage are also stored.

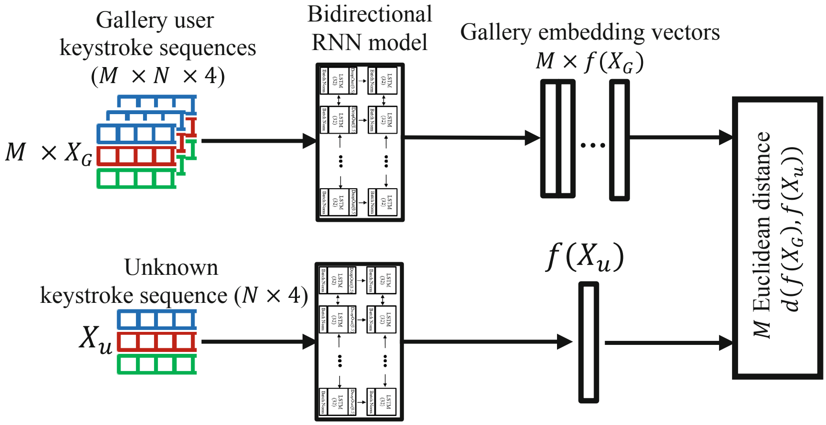

Fig. 4. Siamese keystroke setup for testing. The number M of gallery samples in test varies according to the number of keystroke sessions employed for testing. The number of keys in each sequence N is set to 20.

The structure of the database is divided in sessions (i.e. the elapsed time between the user unlocks the screen until the next lock). For each session, the device stores in a folder all the sensor signals employed in that session. For example, if the user unlocks the smartphone to check the email inbox maybe there are no GPS locations or keystroke signals but WiFi and swipes gestures could be provided. The amount of data and the kind of signals acquired vary according to the user's behavior. This reason motivated us to analyze the temporal performance evolution of our systems at session level instead of fixed time slots like days or weeks. Some users could provide a large amount of information in only one day whereas it could be scarce in other users.

4.2 Experimental Protocol

For the behavioral based-profiling systems (WiFi, GPS Location, and App Usage), we train the templates with the first M sessions acquired for each user and using the remaining sessions as genuine test sessions. Sessions from the others users are considered as impostor data.

The RNN model for keystroke recognition is trained in a Siamese setup, which had showed to perform very well with short time sequences like signatures or smartphone time signals [17, 25]. For this, we train de RNN model with pairs of keystroking sequences from train users (80% of the users). Regarding the training details, the best results were achieved with a learning rate of 0.005, Adam optimizer was used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$ respectively, batch size of 512 pairs and the margin set to $\alpha = 1.5$.

For testing, Fig. 4 shows the details of the setup in which we compare the first M keystroke sequences of each test user (commonly named gallery samples) with new keystroke sequences that belong to the same user (genuine samples) or other test users

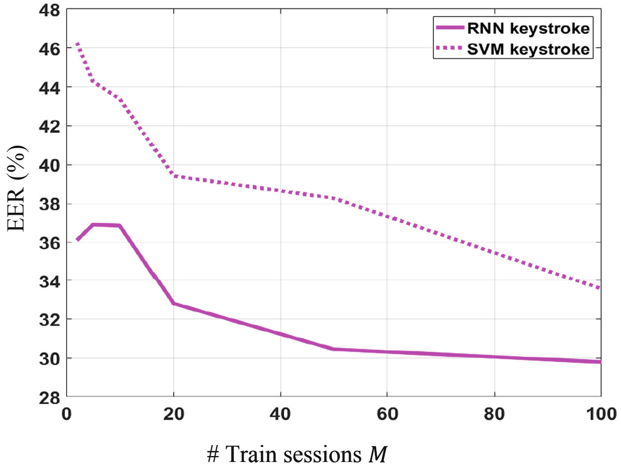

Fig. 5. Evaluation of keystroke performance with bidirectional RNN and SVM models across the number of sessions employed to train.

(impostor samples). Our idea is to build a RNN model able to generalize, distinguishing keystroke sequences from any kind of users.

As we commented before, the keystroke data in UMDAA-02 is stored by sessions and the length of the keystroke sequences vary depending on the session and user, but the RNN model has a fixed length input that we set to $N = 20$, the average of the keystroke sequence length of the database. To avoid zero padding or truncating, we concatenate consecutively keystroke sequences from the user sessions to build the input keystroke sequence of the RNN model so this sequence will belong to only one session in average but could be more or less. In this assumption, we have a total of 8615 keystroke sequence in total for all users in the database.

Finally, to study the temporal evolution of the performance across the time (sessions in this paper) in the keystroke system, we will increase the number of gallery sequences assuming that each gallery sequence is a new genuine user session and then, we test the unknown sample comparing it to all gallery sequences and averaging the *M* resulting distances (see Fig. 4).

4.3 Results and Discussion

We first compare the bidirectional RNN model for keystroke proposed in this paper with a SVM model, following the traditional workflow of global feature extraction and classification. For this, we extract again the same time features as in the RNN model (HL, IL, PL, and Pressure) and then we compute the global features for each time feature: mean, median, standard deviation, 1 percentile, 99 percentile, and 99-1 percentile. According to this protocol, for each keystroke session we have a feature vector of size 24 (6×4). Then, we train a SVM for each user using his/her first *M* keystroke sessions as genuine samples and *M* samples from other users as impostor ones.

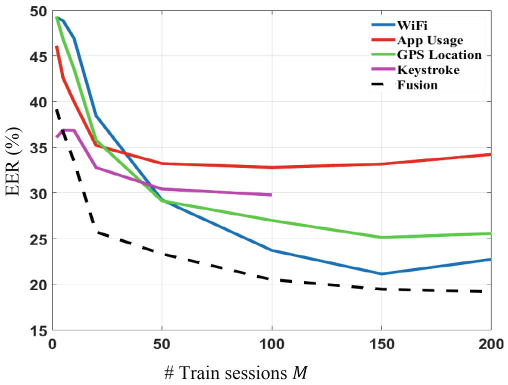

Fig. 6. Evaluation performance for all modalities and the fusion of all across the number of sessions employed to train.

Figure 5 shows the EER curves for keystroke systems using both SVM and RNN algorithms versus the number of sessions employed to train the models. EER refers to Equal Error Rate, the value where False Acceptance Rate (percentage of impostors classified as genuine) and False Rejection Rate (percentage of genuine users classified as impostors) are equal. The results suggest that RNN networks work better than SVM in all cases, even when there are few user samples available for training the RNN algorithm. However, the EER for SVM tends to drop faster than RNN as more user data is available.

Finally, we evaluate the performance of each biometric system individually and the fusion of all of them. Figure 6 shows the performance versus number of sessions employed for training. The results show that the best individual system in terms of EER is the WiFi system, achieving less than 23% of EER with 150 sessions to train the templates. However, the keystroke system drops faster and performs better with few user data. We think that this occurs because the keystroke system (bidirectional RNN) was previously trained with data from other users and learnt discriminative keystroke patterns, being able to authenticate new users with few samples.

The dotted curve shows the fusion of all systems at score level. To get the best of the fusion scheme, we weighted the systems by giving higher weights to the best ones (WiFi and GPS location). The best results are around 19% of EER with more than 150 sessions to train. According to Table 2 (last row), the authentication systems need more than 6 days in average to authenticate users to achieve the best performance possible.

5 Conclusions and Future Work

In this paper we evaluate the performance of mobile keystroke authentication according to: (1) data availability to model the user; and (2) combination with behavioral-based profiling techniques. We have developed an ensemble of three behavioral-based profile authentication techniques (WiFi, GPS Location and App usage) and a keystroke state-of-the-art recognition approach. The results showed that even though behavioral based-profile systems tend to work better with large amounts of training data, the performance gets worse when the amount of data to model the user is scarce. We therefore suggest that behavioral profile systems work well at long terms, when the smartphone has stored enough data to train the templates.

On the other hand, a keystroke system based on bidirectional RNN seems to work better with few samples. We suggest that this happens due to the pre-training phase of the RNN model with development users in a Siamese setup.

Although the keystroke system works better than the others with few samples, the performance is not competitive for large amounts of training data. For future work, we propose to improve the keystroke performance by employing transfer learning techniques and adapting the RNN model to each user when the amount of user data is enough.

Acknowledgments. This work was financed by projects: BIBECA (RTI2018-101248-B-I00 from MICINN/FEDER) and BioGuard (Ayudas Fundacion BBVA).

References

1. Mobile World Congress (2018). https://elpais.com/tecnologia/2018/02/27/actualidad/1519725291_071783.html. Accessed 01 Apr 2019
2. Impacts of Cell Phone Addiction. https://ifpgod.wordpress.com/about/impacts-of-cell-phone-addiction/. Accessed 01 Apr 2019
3. Would you be willing to be in danger to get your cell phone back? http://primerasnoticias.com/2014/05/correr-peligro-recuperar-movil/. Accessed 01 Apr 2019
4. Harbach, M., Von Zezschwitz, E., Fichtner, A., De Luca, A., Smith, A.: It's a hard lock life: a field study of smartphone (un) locking behavior and risk perception. In: 10th Symposium on Usable Privacy and Security (SOUPS), pp. 213–230 (2014)
5. Patel, V.M., Chellappa, R., Chandra, D., Barbello, B.: Continuous user authentication on mobile devices: recent progress and remaining challenges. Proc. IEEE Signal Process. Mag. **33**, 49–61 (2016)
6. Mahbub, U., Sarkar, S., Patel, V.M., Chellappa, R.: Active user authentication for smartphones: a challenge data set and benchmark results. In: Proceedings of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems, New York, USA (2016)
7. Morales, A., Fierrez, J., et al.: Keystroke biometrics ongoing competition. IEEE Access **4**, 7736–7746 (2016)
8. Tappert, C.C., Cha, S.H., Villani, M., Zack, R.S.: A keystroke biometric system for long-text input. In: Optimizing Information Security and Advancing Privacy Assurance: New Technologies, pp. 32–57. IGI Global (2012)
9. Neal, M., Balagani, K., Phoha, V., Rosenberg, A., Serwadda, A., Karim, M.E.: Context-aware active authentication using touch gestures, typing patterns and body movement (No. AFRL-RI-RS-TR-2016-076). Louisiana Tech University, Ruston United States (2016)
10. Monaco, J.V.: Robust keystroke biometric anomaly detection. arXiv preprint arXiv:1606.09075 (2016)
11. Montalvão, J., Freire, E.O., Bezerra Jr., M.A., Garcia, R.: Contributions to empirical analysis of keystroke dynamics in passwords. Pattern Recogn. Lett. **52**, 80–86 (2015)

12. Ceker, C., Upadhyaya, S.: User authentication with keystroke dynamics in long-text data. In: Proceedings of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1–6 (2016)
13. Fridman, L., Weber, S., Greenstadt, R., Kam, M.: Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location. IEEE Syst. J. **11**(2), 513–521 (2017)
14. Xiaofeng, L., Shengfei, Z., Shengwei, Y.: Continuous authentication by free-text keystroke based on CNN plus RNN. Procedia Comput. Sci. **147**, 314–318 (2019)
15. Buschek, D., De Luca, A., Alt, F.: Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Republic of Korea (2015)
16. Monaco, J.V., Tappert, C.C.: The partially observable hidden Markov model and its application to keystroke dynamics. Pattern Recogn. **76**, 449–462 (2018)
17. Deb, D., Ross, A., Jain, A.K., Prakah-Asante, K., Prasad, K.V.: Actions speak louder than (pass) words: passive authentication of smartphone users via deep temporal features. In: Proceedings of the 12th IAPR International Conference on Biometrics, Crete, Greece (2019)
18. Li, G., Bours, P.: Studying WiFi and accelerometer data based authentication method on mobile phones. In: Proceedings of the 2nd International Conference on Biometric Engineering and Applications, Amsterdam, Netherlands (2018)
19. Mahbub, U., Chellappa, R.: PATH: person authentication using trace histories. In: Proceedings of the Ubiquitous Computing, Electronics and Mobile Communication Conference. IEEE, New York (2016)
20. Mahbub, U., Komulainen, J., Ferreira, D., Chellappa, R.: Continuous authentication of smartphones based on application usage. IEEE Transactions on Biometrics, Behavior, and Identity Science **1**(3), 165–180 (2018)
21. Fierrez, J., Morales, A., Vera-Rodriguez, R., Camacho, D.: Multiple classifiers in biometrics. Part 2: trends and challenges. Inf. Fusion **44**, 103–112 (2018)
22. Liu, X., Shen, C., Chen, Y.: Multi-source interactive behavior analysis for continuous user authentication on smartphones. In: Proceedings of Chinese Conference on Biometric Recognition, Urumchi, China (2018)
23. Li, G., Bours, P.: A mobile app authentication approach by fusing the scores from multi-modal data. In: Proceedings of 21st International Conference on Information Fusion, Cambridge, UK (2018)
24. Giancardo, L., et al.: Computer keyboard interaction as an indicator of early Parkinson's disease. In: Sci. Rep. vol. 8 (2016)
25. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Ortega-Garcia, J.: Exploring recurrent neural networks for on-line handwritten signature biometrics. IEEE Access **6**, 1–11 (2018)
26. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR) (2014)