Siamese Networks for Static Keystroke Dynamics Authentication

Romain Giot, Anderson Rocha

Siamese Networks for Static Keystroke Dynamics Authentication

Romain Giot*, Anderson Rocha†,
*Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
Email: romain.giot@u-bordeaux.fr
†University of Campinas, Campinas - SP, Brazil
Email: anderson.rocha@ic.unicamp.br

***Abstract*—Keystroke dynamics authentication aims at recognizing individuals on their way of typing on a keyboard. It suffers from a high intra-class variability as any behavioral biometric modality; to provide a large quantity of enrollment samples often overcomes this issue.**

In this paper, we analyze the feasibility of using siamese networks to rely on biometric samples provided by other users instead of requesting a new user to provide a large number of enrollment samples. Such networks aim at comparing two inputs to compute their similarity: the authentication process consists then at comparing the query to an enrollment sample.

The proposed method is compared to several compatible baselines in the literature. Its EER outperforms the best baseline of 28% in a oneshot context and 31% when using 200 enrollment samples. This proves the viability of such approach and opens the path to improvements for using it in other contexts of keystroke dynamics authentication.

I. Introduction

Biometric authentication allows to verify the identity of individuals based on what they are or how they behave. Keystroke dynamics [1] is the behavioral modality that allows recognizing one individual based on his way to type on a keyboard. The main information collected correspond to various timing deltas between key presses. Static keystroke dynamics flavor relies on the input of a predefined text to make this verification. It is mainly used with the verification password of a single user or the verification of a shared password within a group of users (which is the focus of this paper).

Deep learning-based methods [2] generally perform better than most other historical methods in various domains and biometrics authentication is not an exception [3]. However, most related works concern face and speaker recognition while keystroke dynamics has not attracted much attention. This is explained for practical reasons because deep neural networks need a large collection of samples to train; that implies numerous inputs of the password by each user of the system. This is definitively not doable in practice. However, siamese networks [4] may alleviate this issue by requesting users to provide a reasonable amount of samples.

The feasibility of such kind of architecture to improve the performance of static keystroke dynamics authentication system is verified through different experiments. As this first step is proved to be efficient, it will be necessary to generalize the approach to other kind of keystroke dynamics authentication systems.

The originality of the paper relies on three main points. (i) So far from our knowledge, this is the first time a siamese network is applied in the context of keystroke dynamics authentication. Although the concept is not new, it proves its effectiveness on such subject, and will open the path for future experiments and improvements. (ii) Different suitable architectures are tested in order to select the simplest model having the best performance. (iii) A fair evaluation protocol that ensures results are not over-optimistic is applied: individuals used to train the siamese network are different from individuals used to evaluate the biometric authentication system.

The paper is organized as follows. Section II presents previous works related on (a) keystroke dynamics authentication with a focus on static password and neural networks, and (b) siamese networks for biometrics. Section III describes the proposed method. Section IV presents the experimental protocol used to evaluate the proposed method and Section V describes and discusses the obtained results. Section VI lists some limitations of the current study associated to various ways to tackle then. Finally, Section VI concludes this paper.

II. Related Work

Static keystroke dynamics aims at verifying the identity of individuals based on their way of typing a predefined password. Among the public compatible databases [5]–[7], the CMU dataset [5] has been deeply studied in reason of its huge number of samples per individual. The baseline evaluated by its creators achieves an average (over individuals) Equal Error Rate (EER) of 9.6%. EER corresponds to a standard evaluation metric indicating an equal ratio of False Rejection of genuine samples and False Acceptance of imposters samples.

Other studies rely on this dataset and outperform this baseline. However, they also use different experimental protocols that may make their comparison unfair. We focus on neural network-based methods. DeepSecure [8] obtains an EER of 3% using a four layers Multi-Layers Perceptron (MLP) individually trained for each user. Each model is trained with 200 genuine samples and 5 impostors samples per each other individual. Çeker and Upadhyaya [9] obtain an EER of 2.3%

WIFS'2019, December, 9-12, 2019, Delft, Netherlands. 978-1-7281-3217-4/19/$31.00

with a single multiclass Convolutional Neural Network (CNN) model trained using a tailored data augmentation technique and 80% of the samples. An inductive transfer encoder [10] obtains an EER of 6.3% by mapping the gallery samples in a manifold similar to the one of the query sample and using the Manhattan distance for comparison.

Other datasets have also been experimented. DeepService [11] focuses on mobile-phone free-text authentication. Using a multi-class (one per user) and multi-view (alphabet data, other char data, accelerometer) deep learning model, it reaches 93% of identification accuracy in a closed-set scenario of 40 individuals. Lin *et al.* [12] evaluate a CNN on a database of 10 users. One instance is trained per user; the False Rejection Rate is of 13% for a False Acceptance Rate of 0%.

Additionally, siamese networks have already been successfully applied to other biometric modalities. Some examples follow. OSVNet [13] is a CNN-based siamese network trained using the contrastive loss [14] in order to output two vectors on which the euclidean distance is computed. Results are competitive in comparison to other works on the MCYT-100 database. Facenet [15] proposes the triplet loss on a CNN-like encoder. It tries to enforce a margin between each pair of faces from one person to all other faces. Thus, the training of the network needs triplets of faces: the anchor, a positive sample and a negative sample and the comparison is done on the latent space. Zhang *et al.* [16] use the contrastive loss to learn to generate features specific to image-based gait recognition. Euclidean distance is used for the comparison. The system performs better than most baselines.

III. Methodology

We propose to use a single siamese network shared with all users in the verification process. Such choice is justified by the fact that, in opposite to other network architectures, siamese networks can be trained with samples of individuals that are not real users of the system. It allows real users to provide fewer samples for training: this is more convenient for them.

For the context of biometric authentication, it is possible to add a novel user to the system without collecting myriads of samples and training a complex model. Indeed, the network is already able to verify if two samples of a user are similar enough to grant such user access to a system or consider them different users. And to compute one global model instead of one model per individual is enough for such task. The number of enrollment samples to collect in order to have a system performing properly is reduced.

The enrollment consists of storing the user's samples in a gallery while the verification consists of feeding the query with one enrollment sample into the network and computing their similarity score. It allows the use of neural networks without having enough user's samples to train them. Under the hood, such system also allows to change the manifold of the biometric data space to another one where the comparison is done more efficiently.

The rest of this section describes the workflow.

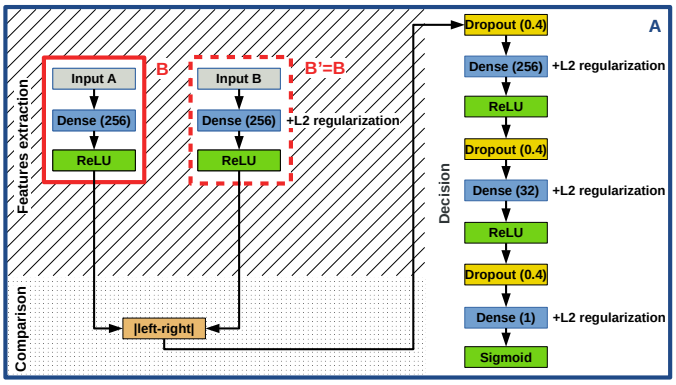


Fig. 1. Siamese network architecture (part A). Each leg (part B&B') extracts features from the samples to compare, a layer computes their absolute difference and the rest of the network decides if both samples are similar.

A. Samples comparison through a siamese network

Figure 1 summarizes the proposed architecture. It has been designed on a trial and error process (see Section V) on the CMU dataset [5]. The two legs of the siamese network (features extraction part) consist of a simple multi-layer perceptron sharing the same weights and architecture; they aim at projecting two keystroke dynamics samples onto the same latent space in which it is easier to compare them. Part of the training consists in optimally searching the transformations to project samples onto this space. Then, the network (comparison part) computes the absolute difference of the two input samples projected onto the latent space. Finally (decision part), another MLP is used to decide if the two biometric samples share the same identity by computing a similarity score. It is thus a full stack network that computes the latent space and the decision.

The feature extraction part consists of a single 256 nodes layer followed by a ReLU activation. An L2 norm regularizes its weight. We have chosen not to use a dropout before the comparison layer. The decision part consists of three dense layers; each one is preceded by a dropout of 40% and its weights are regularized by an L2 norm. The number of neurones is respectively 256, 32, and 1, while their activation is respectively ReLU, ReLU and Sigmoid.

B. Pairs construction for training the network

The siamese network needs to be trained with pairs of samples that belong to two categories: (a) similar pairs contain two samples of a single individual and (b) dissimilar pairs contain two samples of two distinct individuals. Training quality is directly impacted by the construction of these pairs.

The *similar pairs* are built by generating the cross-product of the samples of each user: each sample of each user is matched one time with each other sample of the same user.

The *dissimilar pairs* are randomly generated by matching the sample of one user to the sample of another one. Selection is done with replacement and the number of generated dissimilar pairs is chosen to be the double than the number of similar pairs, while the selected pairs are among the most difficult ones to compare. They are generated as follows: for

10 pairs randomly generated with replacement, we keep only the two ones having the shortest Euclidean distance. This way, we expect the boundary decision to be of better quality.

C. Biometric authentication

A trained siamese network can be used to compute the similarity score of two samples. As for any biometric authentication system, we consider a match when this score is higher than a threshold (that needs to be specified by the operator of the system) and a non match otherwise. Two error rates can be computed given a threshold and a set of comparison scores: False Match Rate (FMR) corresponds to the ratio of dissimilar comparisons considered to be similar and False Non Match Rate (FNMR) corresponds to the ratio of similar comparisons considered to be dissimilar.

However, several samples are collected during the enrollment and any of them could be compared with each query. For this reason, we propose different strategies for the verification process. (i) To compare each sample of the gallery to the query by feeding each pair in the siamese network and to compute the mean of the produced scores. All gallery samples are used whatever is their closeness to the query. (ii) To compare each sample of the gallery to the query by feeding them in the siamese network and to compute the mean of the k best scores. This way, only enrollment samples close enough to the query are taken into account. It should decrease the FNMR but could negatively affect the FMR. (iii) To use the feature extraction (part B only) of the network in order to project the query onto the latent space for comparison. Thus, each biometric sample of the gallery, as well as the query, is mapped onto this latent space thanks to this subnetwork. The score is then computed on this space using a standard distance measure. Although we are confident in the feature extraction part of the network, we prefer to outsource decision-making.

IV. Experimental protocol

This paper focuses on the CMU dataset [5]: 51 individuals each typed 400 times the password ".tie5Roanl" during 8 collection sessions each of 50 inputs. Each input is represented by a 31-sized vector containing keydown-keydown times, keyup-keydown times, and hold times for all keys in the password. Section II has shown that different studies have used it. However, due to the various constraints of their classifiers, most of them used a different experimental protocol. We also need to use a different one, especially on the data partition side, because of our siamese approach.

A. Data partition

The dataset must be split in several sub-datasets, each with a specific semantic. Firstly, a sub-dataset is needed to train the siamese network. It is built using all samples of randomly chosen users S from the whole set of users U. All samples of S are used to generate the training pairs (see Section III-B). The rest of users $B = U \setminus S$ is used for the biometric verification part. The number of samples in the enrollment process is a parameter of our experiments. First samples are used to build their gallery, while the others feed the probe used to compute the biometric scores in order to evaluate the biometric system. Each biometric sample in the probe dataset is compared to each biometric reference. A k-fold approach is used to split the users in S and B. k is fixed to 3, which gives for each run: 34 users for training the network and 17 users for testing the biometric authentication system. The various experiments are repeated three times; we present the aggregated results.

B. Training procedure

Nadam optimizer [17] (with a learning rate of 0.003 and other parameters set to Keras [18] default) is used with the binary crossentropy loss. No extra loss is used to force the comparison layer to generate activations of small (respectively large) norm when fed with similar (respectively dissimilar) samples. Each batch contains $51\,200$ pairs. The network is trained during 150 epochs while monitoring the binary accuracy on the validation set (25% random pairs of the training set). If it does not improve after 10 epochs, the learning rate is decreased by a factor of 0.2. If it does not improve after 30 epochs, the training is stopped. The model having the best binary accuracy among the various epochs is kept.

C. Configuration of the proposed methods

We have previously seen that different combinations can be used for the verification part. The tested one are:

- *network*, that corresponds to the siamese network when all the scores computed against the gallery are averaged;
- *network knn*γ, that corresponds to the siamese network when the best γ scores are averaged (with $\gamma \in [20, 50]$);
- *network manhattan scaled*, that corresponds to the use of the part B of the network to extract features of gallery samples and the query and compute the *manhattan scaled* (see (3)) score on them.

D. Comparison baselines

We have selected different baselines compatible with our training constraints. This is not the case of other neural network based approaches (see Section II) that need users samples during their training whereas we ensure it is not the case in our protocol.

Let $X^i_{gal} = \{x^i_1, ..., x^i_n\}$ be the n training samples of size s of user i, μ^i the mean of X^i_{gal}, θ^i its standard deviation, and $q \in X_{prob}$ a query sample.

The *simple* baseline corresponds to a statistical distance that has been proven to be efficient in [19] and later studies:

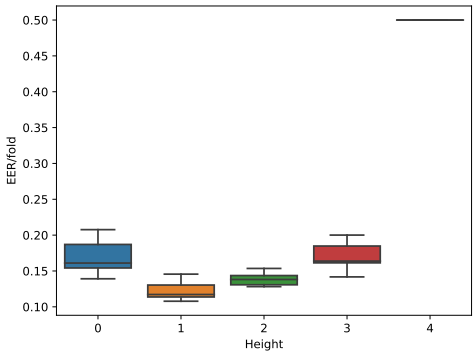
$$1 - \frac{\sum_{j=1}^{s} exp\left(\frac{-|\mu_i(j) - q(j)|}{\theta(j)}\right)}{s} \tag{1}$$

The *manhattan* distance is one of the best classifiers in [5]:

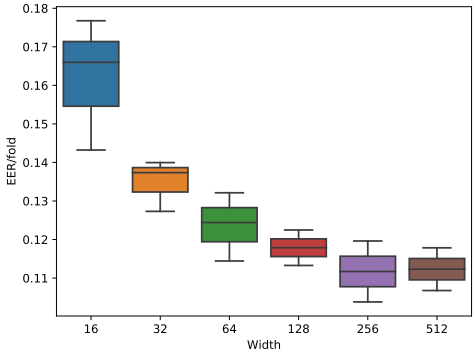
$$\frac{\sum_{k=1}^{n}\sum_{j=1}^{s} |x_k(j) - q(j)|}{n} \tag{2}$$

The *manhattan scaled* version also usually performs better [5]:

$$\frac{\sum_{k=1}^{n}\sum_{j=1}^{s} \frac{|x_k(j) - q(j)|}{a^i(j)}}{n} \tag{3}$$



(a) EER per network part B height. Performance decreases with the complexity of the feature extraction process.



(b) EER per layer width. Performance increases with the width of the network up to 256 neurones.

Fig. 2. Evaluation of different architectures for the proposed method. (a) evaluates different heights and (b) evaluates different width.

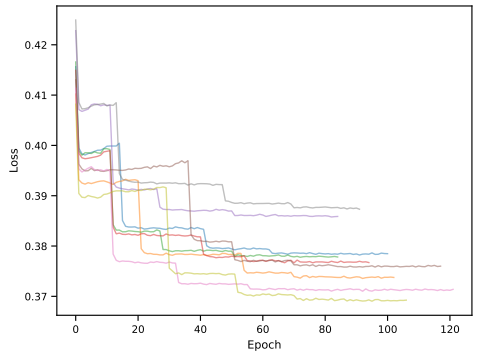


Fig. 3. Loss over epochs for each run. 150 epochs maximum where allowed, the loss corresponds to the binary crossentropy summed with the normalization parameters.

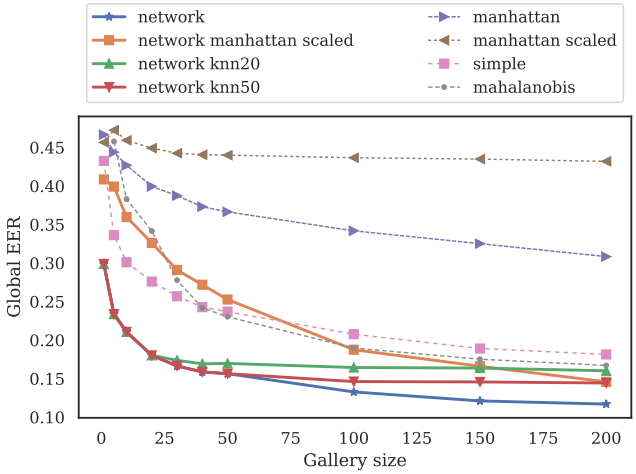


Fig. 4. Global EER for each configuration. After reaching a certain amount of training samples, our method always outperform the baselines.

where a^i corresponds to the mean of the absolute difference of each sample of X^i_{gal} to μ^i.

Finally, the *mahalanobis* distance performs better in [5]:

$$\left(q-\mu^i\right)^T C^{i-1}\left(q-\mu^i\right) \tag{4}$$

where C^i is the covariance matrix computed with X^i_{gal}.

E. Evaluation of the proposed methods

Several questions of interest have to be answered to assert the performance and usefulness of the proposed methods.

Q1 What is the minimum number of layers needed for good performances ? We modified the number of dense layers of part B from the set $[0, 1, 2, 3, 4]$ (0 means there are no features extraction) and compare their mean EER using 200 enrollment samples (similar to most studies evaluating this dataset). Experiment is repeated 2 times, width is fixed to 64.

Q2 What is the typical features dimensional? We modified the number of neurons of the dense layer of part B as well as on the first layer of the decision from the set $[16, 32, 64, 128, 256, 512]$ and compare their mean EER using 200 enrollment samples. Experiment is repeated 2 times, height is fixed to 1.

Q3 What is the impact on the number of enrollment samples on the final decision? Is it possible to do oneshot authentication? There is no pressure to collect several samples for enrollment as the network has been trained with other individuals samples and expect that few samples allow obtaining good performances with an increase of performance with the number of samples. To verify this point, we try different gallery size: $[1, 5, 10, 20, 30, 40, 50, 100, 150, 200]$. We expect to obtain good performance even for few samples.

Q4 How the proposed methods perform compared to the baselines? Nowadays, deep learning-based systems outperform traditional methods and we expect it is the case here; the difficulty being on the fact that baselines do not need masses of data for training in opposite to the proposed methods. We thus compare the performance of the proposals to various baselines.

V. Results and Discussion

Fig. 2a shows the performance of the network depending on the depth of the feature extraction part. The best configuration

TABLE I

AVERAGE (AND STANDARD DEVIATION) OF EER (IN %) OVER USERS AND FOLDS FOR EACH COMBINATION. RED CELLS REPRESENT THE WORST SYSTEM, BLUE CELLS REPRESENT THE BEST SYSTEM, ORANGE CELL REPRESENT THE BEST BASELINE AND GREEN CELLS ARE BETTER THAN THE BEST BASELINE FOR A GIVEN GALLERY SIZE. **BEST COMBINATION** (SYSTEM AND GALLERY SIZE) IS IN BOLD.

Gallery size **System**	1	5	10	20	30	40	50	100	150	200
manhattan scaled	35.1(16.5)	28.8(17.6)	25.8(17.3)	22.9(15.8)	21.1(14.8)	20.3(14.5)	19.4(14.0)	15.8(11.0)	14.3(10.3)	13.2(9.0)
mahalanobis	∅	43.4(16.3)	35.5(13.9)	29.5(11.4)	24.5(12.1)	21.2(11.4)	20.1(11.3)	17.1(9.3)	16.1(9.4)	15.3(8.9)
manhattan	37.9(16.6)	34.1(16.7)	30.9(15.8)	27.0(14.6)	24.9(13.9)	23.9(13.4)	23.2(13.2)	20.5(11.1)	18.8(10.1)	18.0(9.4)
simple	43.3(5.2)	31.5(16.4)	29.0(16.0)	25.3(15.1)	23.1(14.4)	21.7(14.1)	20.8(13.8)	18.1(12.8)	16.6(12.4)	15.7(12.4)
network	25.2(12.8)	21.0(14.0)	18.8(13.2)	15.8(11.1)	14.4(10.5)	13.5(10.3)	13.1(10.4)	10.9(8.7)	9.8(8.1)	**9.1(7.0)**
network knn50	25.2(12.8)	21.0(14.0)	18.8(13.2)	15.8(11.1)	14.4(10.5)	13.5(10.3)	13.1(10.4)	11.8(8.9)	11.3(8.5)	10.9(8.0)
network knn20	25.2(12.8)	21.0(14.0)	18.8(13.2)	15.8(11.1)	14.7(10.4)	14.1(10.2)	13.9(10.2)	13.1(9.2)	12.7(9.1)	12.3(8.6)
network manhattan scaled	32.4(16.0)	34.8(19.1)	30.8(19.0)	26.5(17.7)	23.7(16.8)	22.5(16.4)	21.2(16.0)	16.4(12.4)	14.4(11.5)	12.5(9.1)

has a depth of 1; to add extra layers decreases performances, while using 4 makes the network unable to with our protocol. To do no feature transformation is less efficient that having a single layer. We had expected that a deepest network would allow extracting more complex and discriminative features. We assume it is not the case because of the simplicity of keystroke features and the fact that they already correspond to extracted features. Fig. 2b shows the performance of the network depending on the width of the feature extraction part. Performance increases with the width of the network until reaching a maximum at 256: the preferred feature extractor is wider than the features (256 vs 31) and only needs one layer. Note that this architecture suits well the CMU dataset but may be less effective with another dataset.

Additional modifications have been tested without providing significant improvements. Minmax scaling has been initially used during the first iterations of the study; removing it has improved the results: the network learns faster, has fewer plateaux and then improves during more epochs. We can explain that because the domain of the biometric characteristics is already in $[0; 1]$ and the number of users (and thus variability) is not sufficent to compute a coherent scaling factor. The use of the LeakyReLU as in DeepSecure [8] has not changed the results. Finally, a three inputs network trained using the triplet loss [15] was unable to learn.

The number of parameters to train for part B of the selected network is 8192 while the total number of parameters for the remaining part is $74\,049$: the feature extraction part is less complex than the decision part. Fig. 3 draws the loss (sum of the binary crossentropy and the regularization parameters) over the epochs for each training round. Although 150 epochs were allowed, all rounds early stopped because of a lack of improvement during 30 epochs; most of the training was done during the first 40 epochs.

Fig. 4 presents the global EER (i.e., computed with all aggregated scores) evolution over gallery size for each evaluated system. For each system, to augment the number of enrollment samples increases the performance. With 200 enrollment samples, the performance of the baselines are greatly inferior to the one announced in their original study [5]. This is explained by the difference of evaluation protocol: in our case we have less inter-scores, the k-fold partitioning may have generated more difficult configurations, and we present a global EER. Most of the proposed methods always perform better than the baselines whatever is the number of enrollment samples. Only *network manhattan scaled* shows lower performance with few enrollment samples. It can be explained because the network is trained to minimize the binary accuracy and is not forced to explicitly extract features that differ for similar/dissimilar pairs. The siamese network is then less efficient in extracting features in comparison to decision-making. The *network* systematically performs better than any other methods (*network knn*γ are equal to *network* when the number of enrollment samples is lower than γ which explains the same result in these cases).

Tab. I provides the average (and standard deviation) of EER over users and folds for each combination (i.e., the EER of each user of each fold is independently computed). Each user has a different decision threshold; although it may not be realistic and easy to apply, most papers using the CMU dataset use such evaluation. We can observe that this way of computing results is more optimistic for all systems and *manhattan scaled* jumps from the worst system to the best baseline. From this table, it is still clear that *network* always outperforms the baseline *manhattan scaled*. Standard deviation for the proposed methods is also often smaller than the one of the baselines. *network* EER in the oneshot scenario is about 25% whereas the best baseline achieve about 35% (improvement of 28.2%) and the *mahalanobis* cannot be computed. *network* EER with 200 enrollment samples is about 9% whereas the best baseline achieves about 13% (improvement of 31%). The *network knn*γ methods do not bring additional performance in comparison to *network* which means that intra-variability of this modality is so important that the biometric model have to encode most of it.

From these results: better performances are obtained with a wide but shallow network (Q1 & Q2); to increase the number of training samples increases the recognition performance (Q3); proposed methods provides best results than baselines in the oneshot scenario, but performances are still too low to be usable in an operational scenario (Q3); and the proposed

method performs better than the baselines (Q4).

VI. Limitations of the proposed method

There are still some limitations in this work that deserve to be investigated with additional experiments.

The pairs selection to train the siamese network is based on a greedy algorithm executed only one time before the training process and the choice is only done from the biometric sample space and not from the latent space. It is highly probable that training would be improved if the selection was done on a per epoch basis within the latent space.

This work focuses on the CMU dataset [5] but could be generalized with other ones [6], [7]. A naive approach would be to use a tailored network for each dataset. However, it is more interesting to design a single architecture that works properly with these three datasets.

The system targets static keystroke dynamics authentication with a shared password. A generalization to a user-based password authentication [7] or even free text is expected. Artificial samples generation with handcrafted methods [9], [20] or Generative Adversarial Networks [21] could help to generate additional training data if required for such system.

Finally, it is known that keystroke dynamics suffers from high intra-class variability and adaptive systems [22] are needed to achieve interesting performances. A tailored update system is expected in order to update the gallery of each user as well as the network to take into account user data.

VII. Conclusion

Keystroke dynamics authentication systems allow authenticating individuals based on their way of typing on a keyboard while siamese networks allow to project a pair of inputs onto a latent space in which they are compared in order to compute a similarity score. Such network architecture has been successfully studied for several biometric modality, but never studied for keystroke dynamics.

This paper has presented the feasibility of using a siamese network in the context of keystroke dynamics authentication. For this purpose, a siamese network tailored for the CMU dataset [5] has been proposed. Several widths and depths has been tested in order to select the best performing architecture that remains quite simple as the feature extraction part contains only one layer.

The proposed network has been evaluated and compared to various baselines and has shown its superiority in all cases, especially when few enrollment samples are available. Thanks to these experiments, we know that such kind of architecture can be used in the context of static keystroke dynamics authentication with password shared. It is then necessary to improve it in order to be usable in more scenarios less constrained for the users, such as systems having a different password per individual or free-text based systems.

References

[1] R. Giot, M. El-Abed, and C. Rosenberger, *Keystroke Dynamics Overview*, Jul. 2011, vol. 1, ch. 8, p. 157–182.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[3] K. Sundararajan and D. L. Woodard, "Deep learning for biometrics: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, p. 65, 2018.

[4] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[5] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009, pp. 125–134.

[6] R. Giot, M. El-Abed, and C. Rosenberger, "Greyc keystroke: A benchmark for keystroke dynamics biometric systems," in *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, 2009, pp. 1–6.

[7] ——, "Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis," in *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2012, pp. 11–15.

[8] S. Maheshwary, S. Ganguly, and V. Pudi, "Deep secure: A fast and simple neural network based approach for user authentication and identification via keystroke dynamics," in *IWAISe: First International Workshop on Artificial Intelligence in Security*, 2017, p. 59.

[9] H. Çeker and S. Upadhyaya, "Sensitivity analysis in keystroke dynamics using convolutional neural networks," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, 2017, pp. 1–6.

[10] J. V. Monaco and M. M. Vindiola, "Crossing domains with the inductive transfer encoder: Case study in keystroke biometrics," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2016, pp. 1–8.

[11] L. Sun, Y. Wang, B. Cao, S. Y. Philip, W. Srisa-An, and A. D. Leow, "Sequential keystroke behavioral biometrics for mobile user identification via multi-view deep learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 228–240.

[12] C.-H. Lin, J.-C. Liu, and K.-Y. Lee, "On neural networks for biometric authentication based on keystroke dynamics," *Sensors and Materials*, vol. 30, no. 3, pp. 385–396, 2018.

[13] C. Sekhar, P. Mukherjee, D. S. Guru, and V. Pulabaigari, "Osvnet: Convolutional siamese network for writer independent online signature verification," *arXiv preprint arXiv:1904.00240*, 2019.

[14] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1735–1742.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[16] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2832–2836.

[17] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[18] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[19] S. Hocquet, J.-Y. Ramel, and H. Cardot, "User classification for keystroke dynamics authentication," in *International Conference on Biometrics*, 2007, pp. 531–539.

[20] D. Migdal and C. Rosenberger, "Statistical modeling of keystroke dynamics samples for the generation of synthetic datasets," *Future Generation Computer Systems*, 2019.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[22] P. H. Pisani, A. Mhenni, R. Giot, E. Cherrier, N. Poh, A. C. P. d. L. Ferreira de Carvalho, C. Rosenberger, and N. E. B. Amara, "Adaptive biometric systems: Review and perspectives," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 102:1–102:38, Sep. 2019.