SwipeGAN: Swiping Data Augmentation Using Generative Adversarial Networks for Smartphone User Authentication

Attaullah Buriro
Faculty of Computer Science
Free University of Bolzano
Bolzano, Italy
attaullah.buriro@unibz.it

Francesco Ricci
Faculty of Computer Science
Free University of Bolzano
Bolzano, Italy
fricci@unibz.it

Bruno Crispo
University of Trento
Trento, Italy
bruno.crispo@unitn.it

ABSTRACT

Behavioral biometric-based smartphone user authentication schemes based on touch/swipe have shown to provide the desired usability. However, their accuracy is not yet considered up to the mark. This is primarily due to the lack of a sufficient number of training samples, e.g., swiping gestures[1]: users are reluctant to provide many. Consequently, the application of such authentication techniques in the real world is still limited.

To overcome the shortage of training samples and make behavioral biometric-based schemes more accurate, we propose the usage of Generative Adversarial Networks (GAN) for generating synthetic samples, in our case, or swiping gestures. GAN is an unsupervised approach for synthetic data generation and has already been used in a wide range of applications, such as image and video generation. However, their use in behavioral biometric-based user authentication schemes has not been explored yet. In this paper, we propose SwipeGAN - to generate swiping samples to be used for smartphone user authentication. Extensive experimentation and evaluation show the quality of the generated synthetic *swiping* samples and their efficacy in increasing the accuracy of the authentication scheme.

CCS CONCEPTS

• **Security and privacy**; • **Security Services**; • **Authentication**; • **Biometrics**;

KEYWORDS

Smartphone, Behavioral Authentication, Generative Networks

ACM Reference format:
Attaullah Buriro, Francesco Ricci, and Bruno Crispo. 2021. SwipeGAN: Swiping Data Augmentation Using Generative Adversarial Networks for Smartphone User Authentication. In *Proceedings of 3rd ACM Workshop on Wireless Security and Machine Learning, Abu Dhabi, United Arab Emirates, June 28–July 2, 2021 (WiseML '21),* 6 pages.

[1] *Swiping* is a sequence of generated touch-events as a results of user's finger movement on the smartphone's touch screen [11] [24].

WiseML '21, June 28–July 2, 2021, Abu Dhabi, United Arab Emirates

ACM ISBN 978-1-4503-8561-9/21/06…$15.00
DOI: 10.1145/3468218.3469039

1 INTRODUCTION

Smartphone user authentication is the process of verifying the claimant's identity and to ensure a safe, secure, and legit session. Most popular authentication approaches, e.g., PIN/passwords, face and fingerprint recognition, have shown to be neither secure [23][1] nor usable [17][12]. Hence, researchers have proposed to exploit user behaviors to profile users and authenticate them. Behavioral biometric-based solutions, e.g., keystroke [2], mouse [8, 25], phone-movement [5, 6], offer various advantages, such as, (i) relevant data can be collected unobtrusively, (ii) they have interesting security properties (i.e., they are application dependent), and (iii) they often do not require additional hardware. However, due to high intra-class variations and limited number of training samples, these systems are not yet able to attain the required accuracy. Thus, their actual deployment is still lagging.

Generative Adversarial Network (GAN) is an unsupervised approach for synthetic data generation, proposed by Goodfellow et al. [14]. GAN performs its functionality with the help of two sub-models: the generator and the discriminator. The discriminator model needs to be initially trained on the real data. The generator model generates new samples and the trained discriminator model classifies these samples into either "real" or "not real" ones. Based on this feedback from the discriminator, the generator model iteratively improves the quality of the generated samples. The technique has been already exploited for image generation [30], video generation [29], wireless applications (for spectrum sensing [10] and spoofing wireless signals [26]), and voice and language processing [19, 31], just to name a few. However, very little work has been done to generate tabular (one dimensional 1D) data, especially, in the user authentication domain [21, 22].

This paper presents a data augmentation[2] model powered by GAN to generate synthetic *swiping* samples from "real" samples of a user. This paper, being the first to explore the feasibility of using GAN in *swiping-based* user authentication, tries to establish the foundation of the research in this domain.

We downloaded the publicly available dataset [16], used in DriverAuth [15] and exploited the contained 10, 320 *swiping* samples collected from 86 users. Using a Random Forest (RF) model in the form of a one-class[3] classifier, we achieved an initial True Acceptance Rate (TAR) of 84.66% (compared to 84.7% obtained in the original study) and False Acceptance Rate (FAR) of 14.78%. To demonstrate

[2] Data augmentation [13] refers to the process of generating synthetic samples to make the classifier learn so well that the classification error is decreased, significantly. This technique is believed to improve the classifier's generalization capabilities and so has been widely used in computer vision tasks, e.g., image recognition, etc.
[3] One-class classification is used to model the problem of biometric-based user authentication on smartphones [7, 27].

the quality of the generated synthetic *swiping* samples and their impact on the classifier's accuracy, we added some of them (25%, 50%, 75%, and 100%) to the original training samples and witnessed a significant increase in TAR, from 84.66% to 91.65%, and a reduction of FAR from 14.78% to 11.04%.

In summary, the main contributions of this paper are:

- The proposal of a GAN based solution, named SwipeGAN, to generate high-quality synthetic *swiping* modality samples.
- The demonstration of the usefulness of synthetic *swiping* modality samples for improving the classifier's accuracy, hence coping with a well known limitation of the considered behavioral biometric-based user authentication scheme.

The rest of the paper is organized as follows. Section 2 describes the necessary background to understand the paper. Section 3 presents a high level overview of our approach. Section 4 explains the experimental evaluation and our obtained results. Finally Section 5 concludes the paper with the summary of our findings, and suggestions for possible future works in this research area.

2 BACKGROUND

In this section, we present some necessary background information to help readers in understanding the remainder of the paper.

2.1 Swiping-based Smartphone User Authentication Schemes

Touchscreens, i.e., electronic visual displays that users can control through simple or multi-touch gestures [20], have become leading input devices on mobile platforms. Due to wide popularity of this input method, touch-biometrics (touch dynamics), which is a method aimed at verifying the identity of a person based on the way he/she interacts with the touchscreen, has become widely used and acceptable by end users and security researchers [4, 11, 27]. Hence, some touch-based authentication mechanisms, such as those based on *swiping*, have been made available on most of the latest smartphones.

2.2 One-class vs. Two-class Classification

The smartphone user authentication problem is mainly solved using two approaches: two-class classification, that is, training the authentication classifier on positive (phone owner) and negative (non phone owner) samples; and one-class classification, i.e., training the authentication classifier on positive samples only. Two-class classification is a very powerful approach in discriminating between the positive and negative samples: the classifier learns the decision boundary, in a convenient representation space, between these two groups of samples. Whereas, in a one-class approach the system checks if the test sample is likely to come or not from the distribution of the available positive examples, based on the measurement of the deviation between the incoming sample and the saved samples. In case the measured deviation is higher than a threshold, the samples are considered as negative [4][27].

Two-class classification could be used for an initial assessment and understanding, however, one-class classification is considered as a more practical and feasible approach for user authentication in smartphones [27][7]. In fact, a two-class classifier requires data from both the owner and the non-owners, and such a sharing of biometric information of users leads to privacy concerns.

As already stated, when smartphone user authentication is modeled as a one-class classification problem, the implemented classifier is trained only on *swiping* samples of the smartphone owner. This approach ensures privacy but at the cost of a reduced accuracy, compared with two-class solutions: binary classifiers are better at differentiating classes compared to one-class classifiers. In order to improve their performance, one class classifiers require a much larger number of samples to generalize well, but users are still reluctant to provide an initial large number of samples to the classifiers. To cope with this problem, we propose to use a GAN based solution, named SwipeGAN; it generates, at no cost for the user, more samples and therefore helps a one-class classifier to become more accurate.

2.3 Generative Adversarial Network

In a Generative Adversarial Network (GAN) two models, i.e., the discriminator and the generator, complement each other to achieve better accuracy and stability in the training process. We illustrate the architecture of a GAN in Figure 1. The basic purpose of the generator G is to generate "non-real" samples that are as similar as possible to "real" samples, based on the feedback of the discriminator D. Whereas the goal of the discriminator is to distinguish between "real" samples and samples generated by the Generator G. It is worth mentioning that the input of the generator is a random noise vector z of fixed size (e.g., 1×100). This noise vector is mapped to a new data space to obtain a fake sample (transformed to a multi-dimensional vector), e.g., $G(z)$. After receiving this GAN-generated "non-real" sample from the generator model, the discriminator compares this sample with the "real" samples from the training data and outputs a probability score for the two classes ("real" or "non-real"). As soon as the discriminator D fails to correctly differentiate the two types of samples, the optimal state is reached.

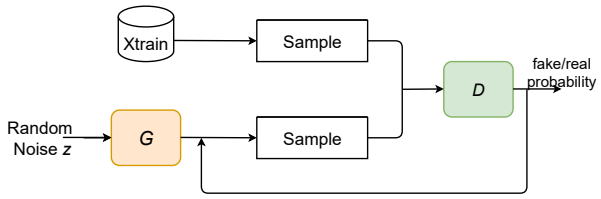

Figure 1: The architecture of a GAN [18]

We have implemented a GAN, in python using keras[4] library, to generate synthetic *swiping* samples. Figure 2 depicts the architecture of the implemented Generator G and Discriminator D for creating GAN-based synthetic *swiping* samples. The implemented generator and discriminator networks are small neural networks with just three layers each. The first layer of the generator accepts a fixed-size (100-dimensional) noise vector. The final output of the generator G network is a 33-dimensional vector (equivalent to the 33-dimensional vector of "real" samples). These 33-dimensional vectors, i.e., from "real" samples and "non-real" samples serve as

[4]https://keras.io/

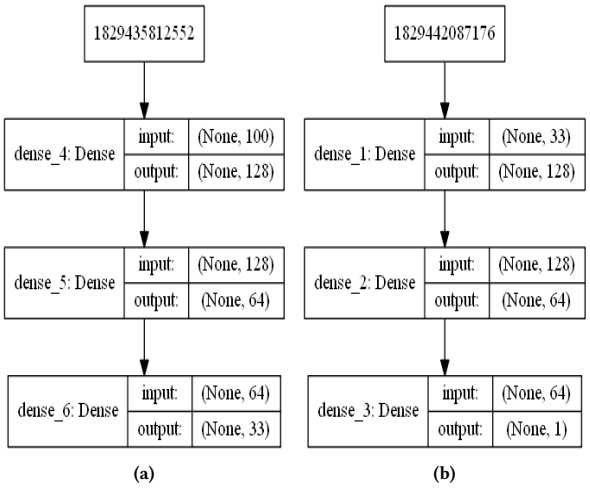

Figure 2: Building blocks of our Simple GAN: (a) generator and (b) discriminator.

the input to the discriminator. The output of the discriminator is binary: the sample is either "real", or "non-real". The size of the hidden layer is 128 neurons in both networks.

We created a lightweight GAN, using small discriminator and generator neural networks, for two reasons: firstly we wanted to demonstrate the efficacy of our approach leveraging even a basic algorithm, like the traditional multilayer perceptron model. Secondly, the implemented GAN needs to be executed on smartphones, which needless to say, have limited resources. So leveraging a more advance scheme (e.g., 1D-CNN) could not be feasible.

2.4 Dynamic Time Warping (DTW)

In [3] 24 different techniques proposed over the years to evaluate the performance of Image-based GANs are surveyed. However, specific measures for the evaluation of 1D-GAN (GANs meant to generate tabular data) are still missing.

Dynamic Time Warping (DTW)[5] has widely been used in several applications including smartphone user authentication [9, 28]. We have decided to evaluate our implemented GAN by adapting to our application the DTW algorithm, and compute the dissimilarity score (termed distance) between real samples of a user and GAN-generated samples for that user.

2.5 Evaluation Metrics

We now define the considered evaluation metrics.

- **True Acceptance Rate (TAR)**: is the percentage of legit samples correctly accepted as legit.
- **False Reject Rate (FRR)**: is the percentage of legit samples incorrectly rejected (erroneously classified as impostors attempts)
- **False Acceptance Rate (FAR)**: is the percentage of impostor samples incorrectly accepted as legit.
- **True Rejection Rate (TRR)**: is the percentage of impostor samples correctly rejected (classified as impostor attempts).
- **Dynamic Time Warping Distance (DTW-d)**: is the measure of dissimilarity between the original matrix, i.e., containing the real samples of the user, and a matrix of samples generated for that user by the GAN. The DTW-d between two identical matrices is 0. The larger the distance is the more different the two matrices are.

3 THE SWIPEGAN APPROACH

The approach proposed in this article includes 3 steps (as shown in Figure 3). In the first step, we train a one-class Random Forest classifier with the samples from the *owner/legit user* and then test the system with the samples belonging to that particular owner. This setting results in either accept or reject. We used a 10-fold stratified cross-validation method in this setting. We preferred cross-validation method because we wanted to test each and every sample of a user. We used the same protocol, i.e., 10-fold cross-validation, for testing the samples of the same user in step 3 as well, however, we tested all the samples of *non-owners* in one go.

The training and testing (using cross-validation) results in TAR and FRR (for the same users). Similarly, non-owner testing results in FAR and TRR. In order to avoid redundancy, we present our results only in TAR and FAR, as FRR and TRR can be calculated as $FRR = 1 - TAR$ and $TRR = 1 - FAR$, respectively. This training and testing is repeated for all the users and we report average results obtained for all 86 users in terms of TAR and FAR. Step 1 aims at obtaining the baseline user authentication results.

In step 2, we train one discriminator for each user on the available 120 "real" *swiping* samples and we generate a new group of 120 "non-real" *swiping* samples. These samples are obtained during the course of 10,000 GAN epochs[6].

We empirically found this number of epochs suitable for this task, as after approximately 8000 epochs, the generator became stable and was generating "non-real" samples very similar to the "real" ones.

In step 3, we again perform the validation procedure by training and testing the same one-class RF classifier that was considered in step 1, but, now it is trained on increased number of samples (generated in step 2) and tested on the same set of "real" samples of all *non-owners*, as in step 1. More specifically, for each user, we train the one-class RF classifiers on a larger number of user's samples, such as, 150, e.g., by adding 25% of the GAN generated samples (30) to the "real" samples of a user and apply 10-fold cross-validation to train and test for intra-user setting and test this trained RF classifier on the "real" samples of all non-owners in one go.

4 EXPERIMENTAL RESULTS

4.1 Dataset

The aim of the paper is to demonstrate the effectiveness of SwipeGAN in generating high-quality synthetic *swiping* samples that can be used to increase the original classifier accuracy. To this end, we leverage the publicly available dataset [16] for our analysis. This dataset has already been used in a latest study [15]. This dataset

[5] https://medium.com/walmartglobaltech/time-series-similarity-using-dynamic-time-warping-explained-9d09119e48ec

[6] An epoch refers to a cycle through the complete train set.

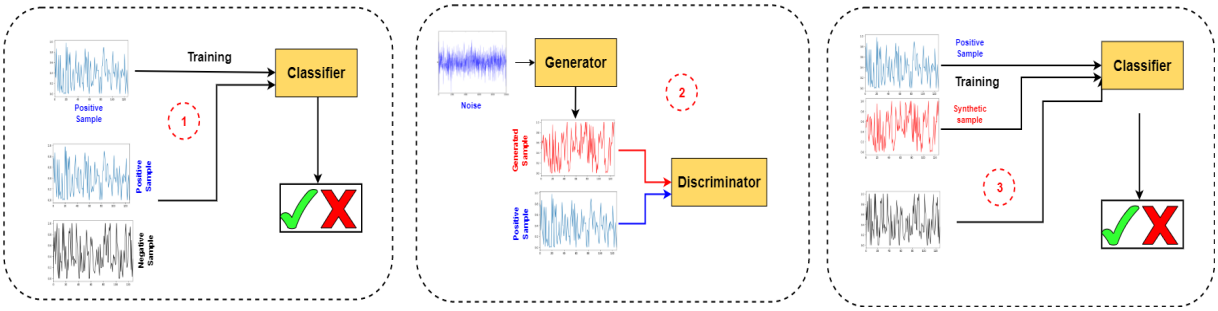

Figure 3: SwipeGAN approach, in 3 steps. (1) User authentication with a one-class classifier based on real swiping samples; (2) generation of additional *swiping* samples; (3) training the one class classifier on the increased number of samples and testing it.

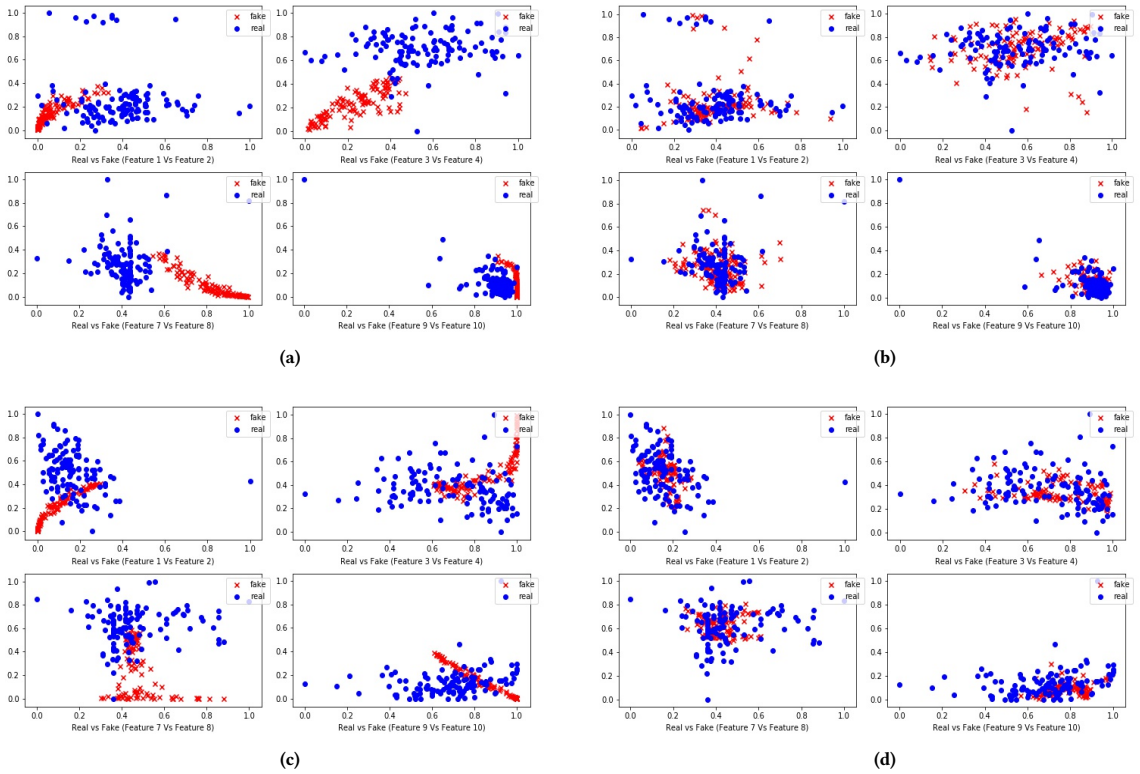

Figure 4: The comparison of eight features of the real (blue) and the GAN generated non-real (red) samples for 2 randomly chosen users (user 31, user 37). Non real samples are generated after 500 epochs for user 31 (a), and after 10000 epochs (b). Non real samples are generated after 500 epochs for User 37 (c) and after 10000 epochs (d)

provides the extracted features (listed in Table 1) collected from samples of 86 users. The final feature vector for each user is 33-dimensional vector in length.

The shared dataset columns have different minimum and maximum ranges, thus to effectively train the RF classifier, we perform normalization using the Min-Max Scaler[7] utility available in Scikit-learn[8]. In Min-max normalization, commonly used in these applications, the minimum and maximum values of a feature become 0 and 1, respectively, and the other values range between 0 and 1.

[7]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[8]https://scikit-learn.org/

Table 1: Swipe features [16] (the number in brackets show feature's position in the final feature vector)

#	Features
1	Duration (*1*), Average event size (*2*), Event size down (*3*), Pressure down (*4*)
2	Start X (*5*), Start Y (*6*), End X (*7*), End Y (*8*)
3	Velocity X Min (*9*), Velocity X Max (*10*), Velocity X Average (*11*), Velocity X STD (*12*), Velocity X VAR (*13*)
4	Velocity Y Min (*14*), Velocity Y Max (*15*), Velocity Y Average (*16*), Velocity Y STD (*17*), Velocity Y VAR (*18*)
5	Acceleration X MIN (*19*), Acceleration X Max (*20*), Acceleration X AVG (*21*), Acceleration X STD (*22*), Acceleration X VAR (*23*)
6	Acceleration Y MIN (*24*), Acceleration Y Max (*25*), Acceleration Y AVG (*26*), Acceleration Y STD (*27*), Acceleration Y VAR (*28*)
7	Pressure Min (*29*), Pressure Max (*30*), Pressure AVG (*31*), Pressure STD (*32*), Pressure VAR (*33*)

4.2 Results

We present here the results obtained in the above-mentioned settings.

4.2.1 ***Quality of the GAN generated "non-real" swiping samples.*** By applying the generator G network architecture we tried to generate as similar vectors as possible to the real vectors. The generated vectors ("non-real" samples) are saved after every 500 epochs. We started saving the samples after 500 epochs, because before that, the generated samples were just random noise. This could be visually checked in Figure 4. Even after 500 epochs, the generated samples are very different from the real ones. We ran the experiment for 10,000 epochs and generated, after every 500 epochs, 120 "non-real" samples for each user, hence, in total 2400 samples per user.

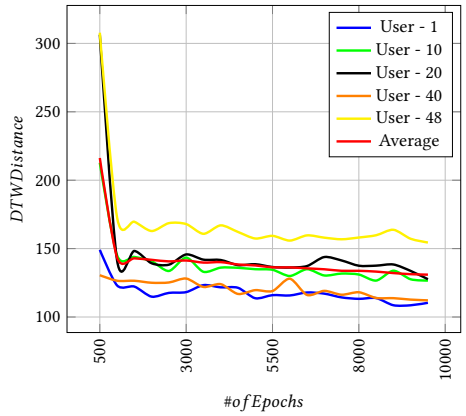

Figure 5: Visualization of the DTW distance between a user's real samples and their corresponding GAN-generated samples. Five users are here considered. The red line shows the averaged DTW distance (for all 86 users).

The samples generated after 500 epoch were much less similar to the "real" samples, compared to the samples generated in the final epoch (10000^{th}). In Figure 5 the DTW distance is used to compare real and false samples. The obtained DTW score for the initial epochs is clearly higher than the final ones. For better readability, we plot the DTW distance for 5 users, including also the average DTW distance (averaged over all 86 users, red line).

4.2.2 ***Classification Results.*** We measured an average TAR of 84.66% and FAR of 14.78% for our basic setting, i.e., without using any GAN generated non-real sample (step 1). We recall that in this scenario the RF classifier was trained only on (a subset) of the real samples of a user and tested against the remaining "real" samples of the user and all the real samples of the other users. We note that in the original study [15] the authors obtained a similar TAR (84.7%) by using Ensemble Bagged Tree (EBT), which is just a variant of the same RF classifier we have used. In fact, it uses the "Breiman's random forest algorithm[9]".

4.2.3 ***Accuracy improvement by adding GAN generated synthetic swiping samples.*** We then increased the number of training samples by adding 25%, 50%, 75%, and 100% of the last GAN generated samples to the original "real" samples of a user to check the impact of this data augmentation solution. We note that in this way, the generated synthetic samples are treated as positive samples of the user. The obtained results show that by increasing the number of synthetic samples the accuracy of classifier is improved: TAR increases and FAR reduces (Figure 6).

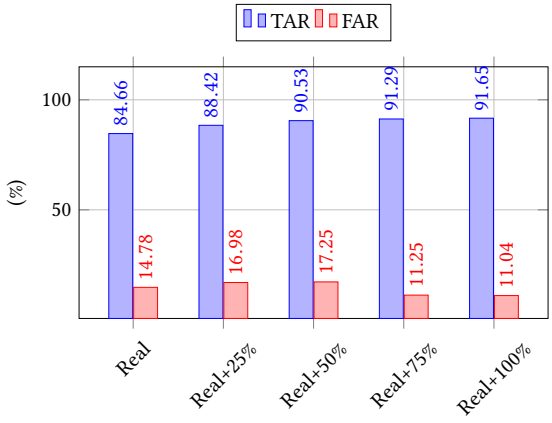

Figure 6: Performance comparison of RF classifier on real observations and augmented dataset, averaged on 86 users.

5 CONCLUSIONS AND FUTURE WORK

We have proposed a GAN based approach, named SwipeGAN, to generate synthetic *swiping* samples. We demonstrate the quality of the generated "non-real" samples and the benefits of using them in user authentication.

[9]https://www.mathworks.com/help/stats/choose-a-classifier.html

The main purpose of using SwipeGAN is to generate synthetic "non-real" swiping samples to improve the accuracy of a one-class user authentication classifier. We demonstrate that by adding to the training set these GAN-generated "non-real" samples result in a significant improvement in TAR (from 84.66% to 91.65%) and a reduction of FAR (from 14.78% to 11.04%). It is worth noting that the reported accuracy (TAR of $\approx$ 92%) is achieved by using the default settings of the RF classifier. As future work, we will investigate if by optimizing the configuration parameters one can further improve the accuracy.

We are in the process of implementing SwipeGAN for Android smartphones based on the findings of this work. Then, we are planning to conduct in-the-wild experiments to validate our approach. We will report the obtained results in terms of computational cost (power, processing and memory consumption), incurred time (for signal generation, training and testing time), and accuracy.

Moreover, we note that "non-real" samples generated using SwipeGAN, could also be used to investigate the robustness of such a classifier towards such carefully-crafted high quality samples. However, for this attack scenario, the attacker, in order to generate "non-real" samples, needs to have some real samples of the target user. It is understandable that accessing the real samples of a target user (stored on his/her device) would be extremely difficult, however, an attacker could try to generate acceptable swiping samples by exploiting some services that require user *swiping*. We leave this analysis also as our future work.

Multimodal systems (involving two or more factors) are believed to be more secure and more robust compared to uni-modals systems. As future work, we also plan to extend the proposed approach to multimodal system(s) and check their robustness against GAN-generated well-crafted samples. Furthermore, we plan also to exploit advanced networks such as Convolutional Neural Network (1D-CNN version) for developing the discriminator and the generator.

Finally, we also plan to implement and test SwipeGAN for Android smartphones and report the results in terms of power/memory consumption and/or processing time.

REFERENCES

[1] Zahid Akhtar. 2012. Security of multimodal biometric systems against spoof attacks. *Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy* 6 (2012).

[2] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. 2002. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)* 5, 4 (2002), 367–397.

[3] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41–65.

[4] Attaullah Buriro. 2017. *Behavioral biometrics for smartphone user authentication.* Ph.D. Dissertation. University of Trento.

[5] Attaullah Buriro, Bruno Crispo, and Mauro Conti. 2019. AnswerAuth: A bimodal behavioral biometric-based user authentication scheme for smartphones. *Journal of information security and applications* 44 (2019), 89–103.

[6] Attaullah Buriro, Bruno Crispo, Filippo Delfrari, and Konrad Wrona. 2016. Hold and sign: A novel behavioral biometrics for smartphone user authentication. In *2016 IEEE Security and Privacy Workshops (SPW).* IEEE, 276–285.

[7] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. 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.* 1393–1402.

[8] Penny Chong, Yi Xiang Marcus Tan, Juan Guarnizo, Yuval Elovici, and Alexander Binder. 2018. Mouse authentication without the temporal aspect–what does a 2d-cnn learn?. In *2018 IEEE Security and Privacy Workshops (SPW).* IEEE, 15–21.

[9] Mauro Conti, Irina Zachia-Zlatea, and Bruno Crispo. 2011. Mind how you answer me! Transparently authenticating the user of a smartphone when answering or placing a call. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security.* 249–259.

[10] Kemal Davaslioglu and Yalin E Sagduyu. 2018. Generative adversarial learning for spectrum sensing. In *2018 IEEE International Conference on Communications (ICC).* IEEE, 1–6.

[11] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. 2012. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 987–996.

[12] Alexander De Luca, Alina Hang, Emanuel Von Zezschwitz, and Heinrich Hussmann. 2015. I feel like I'm taking selfies all day! Towards understanding biometric authentication on smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems.* 1411–1414.

[13] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455* (2018).

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems.* 2672–2680.

[15] Sandeep Gupta, Attaullah Buriro, and Bruno Crispo. 2019. Driverauth: A risk-based multi-modal biometric-based driver authentication scheme for ride-sharing platforms. *Computers & Security* 83 (2019), 122–139.

[16] Sandeep Gupta, Attaullah Buriro, and Bruno Crispo. 2020. A chimerical dataset combining physiological and behavioral biometric traits for reliable user authentication on smart devices and ecosystems. *Data in brief* 28 (2020), 104924.

[17] Marian Harbach, Emanuel Von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It'sa hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *10th Symposium On Usable Privacy and Security ({SOUPS} 2014).* 213–230.

[18] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. [n. d.]. LOGAN: Evaluating Information Leakage of Generative Models Using Generative Adversarial Networks. ([n. d.]).

[19] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547* (2017).

[20] Weizhi Meng, Duncan S Wong, Steven Furnell, and Jianying Zhou. 2014. Surveying the development of biometric user authentication on mobile phones. *IEEE Communications Surveys & Tutorials* 17, 3 (2014), 1268–1293.

[21] John V Monaco, Md Liakat Ali, and Charles C Tappert. 2015. Spoofing keypress latencies with a generative keystroke dynamics model. In *2015 IEEE 7th international conference on biometrics theory, applications and systems (BTAS).* IEEE, 1–8.

[22] Tanya Piplani, Nick Merill, and John Chuang. 2018. Faking it, Making it: Fooling and Improving Brain-Based Authentication with Generative Adversarial Networks. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS).* IEEE, 1–7.

[23] Mudassar Raza, Muhammad Iqbal, Muhammad Sharif, and Waqas Haider. 2012. A survey of password attacks and comparative analysis on methods for secure authentication. *World Applied Sciences Journal* 19, 4 (2012), 439–444.

[24] Napa Sae-Bae, Nasir Memon, Katherine Isbister, and Kowsar Ahmed. 2014. Multitouch Gesture-Based Authentication. *IEEE Transactions on Information Forensics and Security* 9, 4 (2014), 568–582.

[25] Chao Shen, Zhongmin Cai, Xiaohong Guan, Youtian Du, and Roy A Maxion. 2012. User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security* 8, 1 (2012), 16–30.

[26] Yi Shi, Kemal Davaslioglu, and Yalin E Sagduyu. 2020. Generative Adversarial Network in the Air: Deep Adversarial Learning for Wireless Signal Spoofing. *IEEE Transactions on Cognitive Communications and Networking* (2020).

[27] Zdeňka Sitová, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S Balagani. 2015. HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security* 11, 5 (2015), 877–892.

[28] Toan Van Nguyen, Napa Sae-Bae, and Nasir Memon. 2014. Finger-drawn pin authentication on touch devices. In *2014 IEEE International Conference on Image Processing (ICIP).* IEEE, 5002–5006.

[29] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Advances in neural information processing systems.* 613–621.

[30] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. 2017. Lr-gan: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560* (2017).

[31] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence.*