CIS 700 Advances in Deep Learning Project Report

DeepStroke: Multi-device Keystroke spoofing using GANs

Amith Kamath Belman and Shivanand Venkanna Sheshappanavar

Introduction

Predict "typing behavior" or "Keystrokes" of a person on one device, after learning their typing behavior on other types of devices is a novel research idea which needs to be addressed to identify individuals across devices correctly. For example, given a person's typing behavior on Desktop and Tablet, Can we predict their typing behavior on Phone?.

Related Works

Deep learning architectures are being widely used in synthetic sensor data generation especially using LSTM models[1]. However, most of the advanced deep learning models have been demonstrated for their ability to work using image datasets which is why there is a need for transforming sensor data into images[2]. Recent research like [3] presents a multiview and multiclass framework called deepservice gives an idea about representation of keystroke data. Generative Adversarial Networks are used in attacking speaker recognition[4] systems and generating fingerprints to spoof fingerprint recognition[5] systems.

Dataset

The dataset for our experiment was not available beforehand in the form of any open source dataset. Therefore, we collected the data from a group of 20 participants on 3 devices, Desktop, Tablet and Phone. The Figures 1 and 2 illustrate the various transformations that the data goes through.

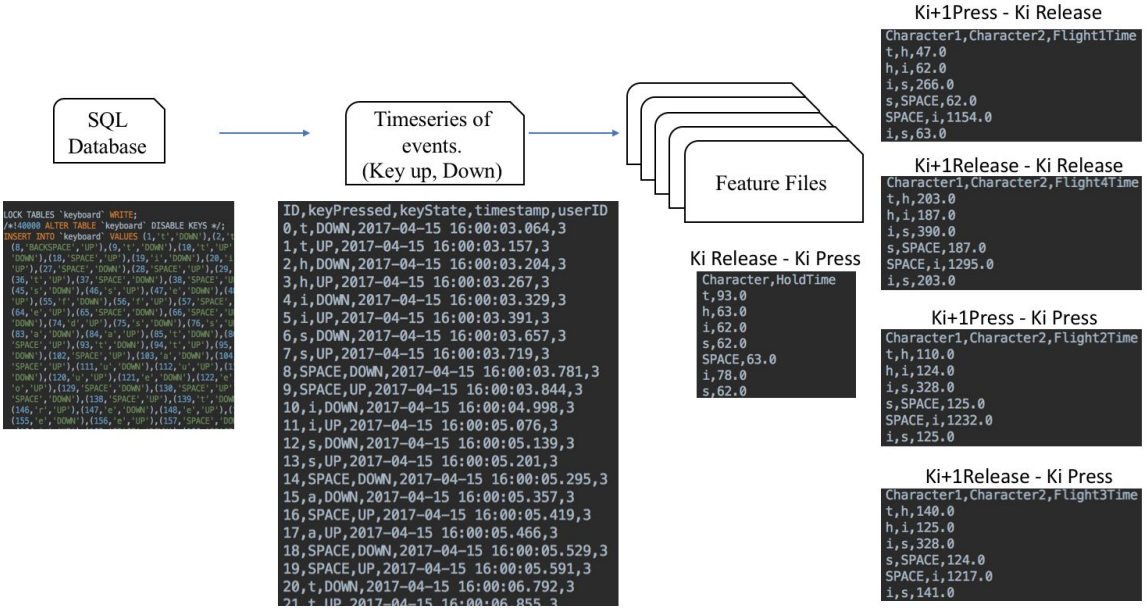
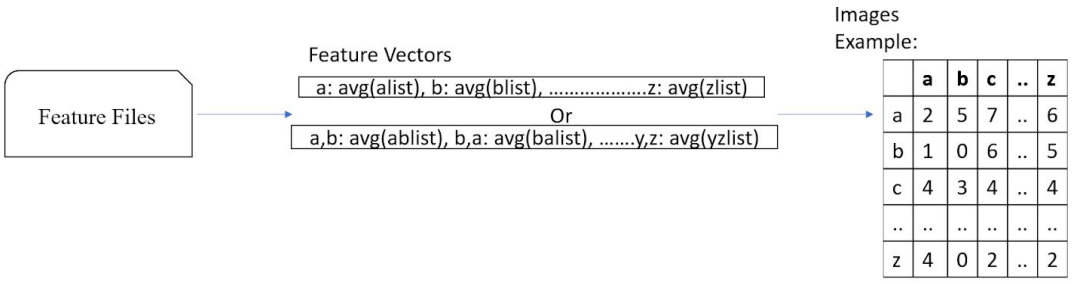


Figure 1:Phase 1 of the data transformation.

As shown in Figure above, the data collected onto SQL database, is first converted into a Time Series format, which is then used to extract the features from unigraphs (single characters) and digraphs (two consecutive characters). The 5 features extracted are Hold (unigraph), Flight1 through Flight4 (digraph). For any two keys Ki and Ki+1 the following temporal features can be extracted:

(a) KeyholdKi : Ki Release - Ki Press
(b) KeyholdKi +1 : Ki+1Release - Ki+1Press
(c) Flight 1Ki Ki +1 : Ki+1Press - Ki Release
(d) Flight 2Ki Ki +1 : Ki+1Release - Ki Release
(e) Flight 3Ki Ki +1 : Ki+1Press - Ki Press
(f) Flight 4Ki Ki +1 : Ki+1Release - Ki Press



	a	b	c	..	z
a	2	5	7	..	6
b	1	0	6	..	5
c	4	3	4	..	4
..	..	..	..	..	..
z	4	0	2	..	2

Figure 2: Phase 2 of the data transformation.

As shown in figure above, once the features have been extracted we further transform these features into feature vectors, which contain of dictionaries with the corresponding list of features values. We then convert these vectors into images by using the keys in dictionaries as the indices for the pixel values. An example is shown in Figure 2. We select the final 30 Character list as: [a,b,c …. z, space,backspace,.,',’] which were used as indices. During this phase we faced a setback, we could generate only 1 image per person, per feature and per device (1*20*5*3= 300 images in total). This was not enough images to train/test GAN. As a solution we created base matrices where each entry was [μ , σ] for the corresponding unigraph/digraph. Using these base matrices generated 500 images per person, per feature and per device.(500*20*5*3= 150,000 images in total). Each pixel is a random value generated in the range μ+σ to μ-σ from its corresponding base matrix.

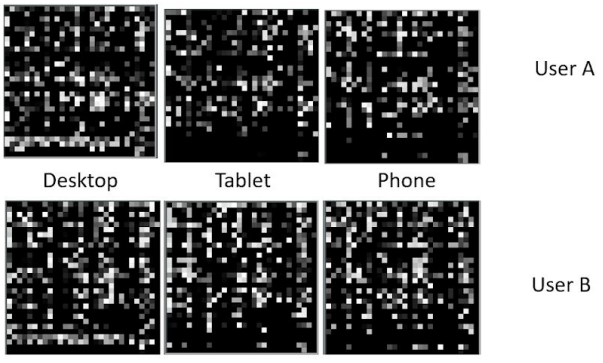


Figure 3: Examples of final images created for 2 users on all three devices.

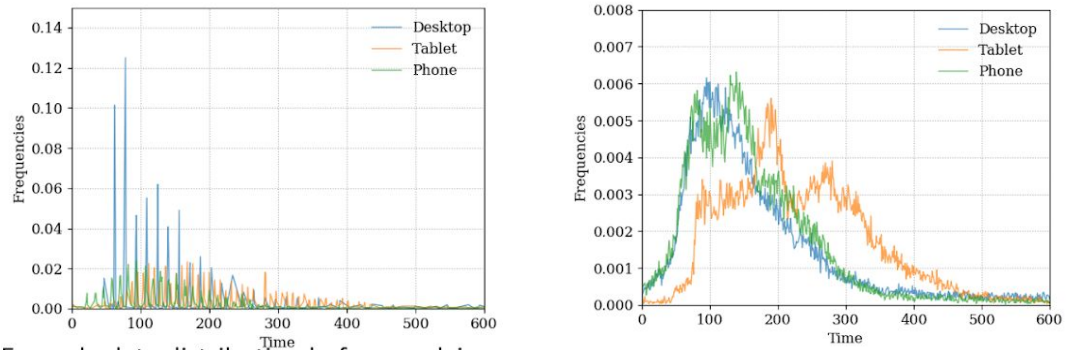


Figure 4: The data distribution before (left) and after (right) using oversampling technique.

The figures above show an example of the images generated for each user, and the distribution of the pixel values before and after the oversampling technique. In total we generated 150,000 images to be used in our work.

Method

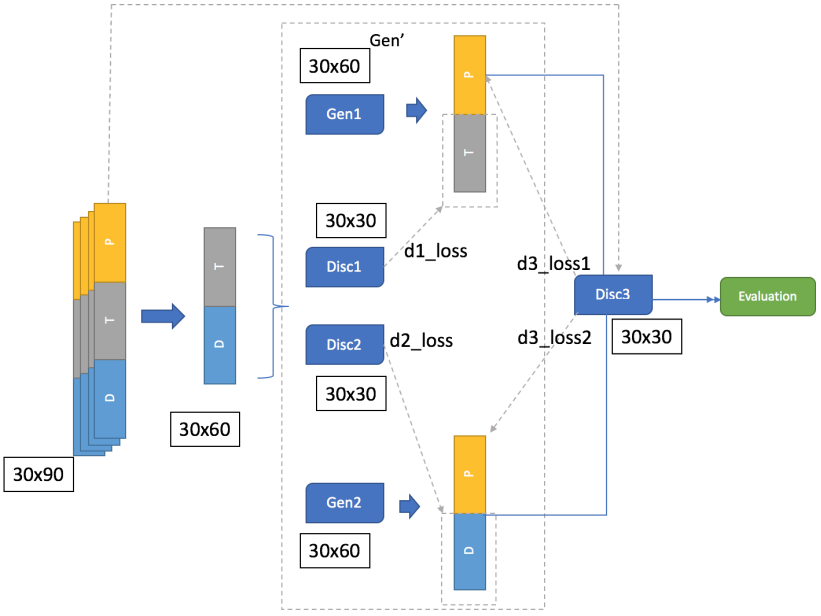


Figure 5: Model uses two generators Gen1 and Gen2 to generate (tablet, phone) and (desktop, phone) combinations together. Three discriminators Disc1, Disc2 and Disc3 each trained only by tablet, desktop and phone real data respectively.

Disc3 is an unique discriminator in this model which discriminates phone data images generated by both generator Gen1 and Gen2. Gen1 and Gen2 also generate tablet and desktop data along with the phone data. The parameters which we used in the model are training batch size of 32, input image height of 30, input image width 30, input image size of 900, one output image of height 30 and width 60, 128 units in fully connected layer, leaky relu and sigmoid generator's activation functions,leaky relu as discriminator activation function, softmax loss function, adam optimizer with generator and discriminator learning rate of 0.01

Results

We are listing the losses of generators (g1,g2) and discriminators (d1, d2, d3 both 1 and 2) after step iterations of 2000,4000,6000,8000 and 10000. For all five features we found the discriminator loss is decreasing rapidly indicating that the discriminators are improving at their tasks where as generators loss though not decreasing but the increase in loss is minute.

Table 1: FlightOne

Iterations	g1_loss	d1_loss	g2_loss	d2_loss	d3_loss1	d3_loss2
2000	7.04	7.69	7.05	7.50	5.58	5.55
4000	7.26	6.68	7.27	6.43	2.73	2.77
6000	7.39	5.98	7.39	5.73	1.46	1.50
8000	7.46	5.40	7.46	5.20	0.82	0.85
10000	7.51	4.97	7.52	4.79	0.55	0.55

Table 2: FlightTwo

Iterations	g1_loss	d1_loss	g2_loss	d2_loss	d3_loss1	d3_loss2
2000	7.02	7.93	7.03	7.83	5.99	5.96
4000	7.20	7.14	7.21	7.03	3.72	3.74
6000	7.32	6.03	7.32	5.96	2.54	2.51
8000	7.41	4.94	7.40	5.02	2.08	2.11
10000	7.44	4.36	7.44	4.52	2.03	2.06

Table 3: FlightThree

Iterations	g1_loss	d1_loss	g2_loss	d2_loss	d3_loss1	d3_loss2
2000	7.10	6.95	7.11	6.94	4.38	4.37
4000	7.24	6.37	7.25	6.44	2.86	2.91
6000	7.35	5.66	7.35	5.69	1.85	1.94
8000	7.42	5.08	7.42	5.08	1.55	1.54
10000	7.46	4.66	7.46	4.56	1.50	1.45

Table 4: FlightFour

Iterations	g1_loss	d1_loss	g2_loss	d2_loss	d3_loss1	d3_loss2
2000	7.01	7.74	7.01	7.78	6.12	6.04
4000	7.15	7.07	7.17	7.11	4.28	4.27
6000	7.25	5.92	7.26	6.02	3.10	3.14
8000	7.32	4.68	7.30	4.98	2.56	2.36
10000	7.35	3.69	7.34	4.07	2.39	2.10

Table 5: Hold

Iterations	g1_loss	d1_loss	g2_loss	d2_loss	d3_loss1	d3_loss2
2000	7.06	7.57	7.07	7.38	5.30	5.25
4000	7.29	6.48	7.30	6.30	2.36	2.39
6000	7.43	5.74	7.43	5.63	0.87	0.95
8000	7.50	5.30	7.50	5.29	0.25	0.30
10000	7.54	5.05	7.54	5.10	0.13	0.02

The figure below illustrates the visual meaning of the

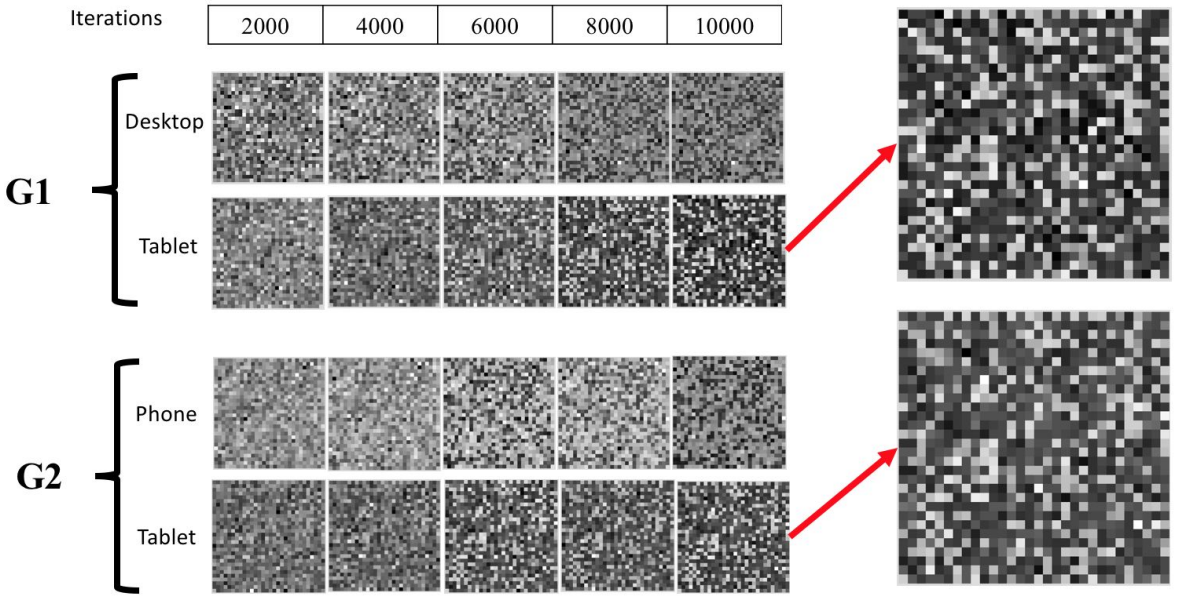


Figure 6: Illustration of similarity in tablet data generated by both G1 and G2 while generating desktop and phone data respectively starting from random noise.

Conclusion

Generation of keystroke data belonging to multiple devices is possible using a single generator. However, discriminators are key in building GAN based models. Learning mappings between multiple devices by models is the key in spoofing keystroke data. This work encourages adopting single generator based GAN models described in StarGAN[10].

References

[1] Alzantot, Moustafa, Supriyo Chakraborty, and Mani Srivastava. "Sensegen: A deep learning architecture for synthetic sensor data generation." Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. IEEE, 2017.

[2] Singh, Monit Shah, et al. "Transforming sensor data to the image domain for deep learning—An application to footstep detection." Neural Networks (IJCNN), 2017 International Joint Conference on. IEEE, 2017.

[3]Sun, Lichao, et al. "Sequential Keystroke Behavioral Biometrics for Mobile User Identification via Multi-view Deep Learning." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2017.

[4] Cai, Wilson, Anish Doshi, and Rafael Valle. "Attacking Speaker Recognition With Deep Generative Models." arXiv preprint arXiv:1801.02384 (2018).

[5] Bontrager, Philip, Julian Togelius, and Nasir Memon. "DeepMasterPrint: Generating Fingerprints for Presentation Attacks." arXiv preprint arXiv:1705.07386 (2017).

[6] A machine learning approach to keystroke dynamics based user authentication, Kenneth Revelt et. al, Int. J. Electronics Security and Digital Forensics, 2007

[7] Performance of Keystroke Biometrics Authentication System Using Artificial Neural Network (ANN) and Distance Classifier Method, N. Harun et. al, International Conference on Computer and Communication Engineering, 2010.

[8] Performance of Keystroke Biometrics Authentication System Using Multilayer Perceptron Neural Network (MLP NN), N. Harun et.al, CSNDSP 2010.

[9] Neural Network Based Authentication and Verification for Web Based Keystroke Dynamics, Raghu et. al, IJCSI, 2011.

[10] Choi, Yunjey, et al. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation." arXiv preprint arXiv:1711.09020 (2017).

Code:

https://github.com/sheshappanavar/DeepStroke.git