Synthetic Keystroke Dynamics Generation Using a Generative Adversarial Network GAN

Abir Mhenni*†, Christophe Rosenberger* and Najoua Essoukri Ben Amara†
*University Caen Normandy, ENSICAEN, CNRS, Normandy Univ,
GREYC UMR6072, F-14000 Caen, France
† University Of Sousse, National School of Engineers of Sousse ENISo,
LATIS- Laboratory of Advanced Technology and Intelligent Systems, 4002, Sousse, Tunisia;
Emails: abirmhenni@gmail.com, christophe.rosenberger@ensicaen.fr and najoua.benamara@eniso.rnu.tn

***Abstract*—Keystroke dynamics, a behavioral biometric modality, offers promising applications in authentication and intrusion detection systems. However, the scarcity of publicly available datasets due to privacy concerns limits research progress. This paper presents a Generative Adversarial Network (GAN) framework to generate synthetic keystroke dynamics data that closely mimics real-world patterns. Using the DSL-StrongPasswordData dataset, we pre-process and normalize timing features and train a GAN with 100-dimensional latent space, LeakyReLU activations, and binary cross-entropy loss. We evaluated the synthetic data through visual comparisons (boxplots, t-SNE projections) and statistical tests (Kolmogorov-Smirnov), demonstrating that the generated distributions align with real data (p-value $>$ 0.05 for key features). Our results highlight the potential of the GAN for sharing data that preserve privacy and increase training sets for keystroke-based models.**

***Index Terms*—User Authentication, Security, Deep Learning, Keystroke Dynamics, Generative Adversarial Network, GAN, Kolmogorov-Smirnov, Biometrics.**

I. Introduction

Keystroke dynamics is an emerging behavioral biometric modality that leverages typing patterns to authenticate users [1]–[3]. Unlike traditional authentication methods such as passwords or fingerprint recognition, keystroke dynamics offers a non-intrusive, continuous authentication approach based on an individual's unique typing behavior [4]. This technique has gained significant attention due to its potential applications in cybersecurity, user authentication, and fraud detection [5], [6].

Machine learning techniques have played a crucial role in enhancing keystroke dynamics-based authentication [7]. However, conventional classification models often require large amounts of labeled data to achieve high accuracy, making them impractical for real-world applications where the number of labeled samples per user is limited [8]. Moreover, research in this field faces a critical challenge: the scarcity of publicly available datasets due to privacy concerns and proprietary restrictions [9].

Traditional methods for addressing data scarcity (e.g., oversampling or Gaussian noise injection) often fail to capture the complex temporal dependencies and user-specific variances in keystroke dynamics [10]. Generative Adversarial Networks (GANs), which learn data distributions adversarially, offer a promising alternative. Although GANs have succeeded in generating synthetic images and time series data [11], their application to keystroke dynamics, particularly to preserve user-specific patterns, remains underexplored.

The main contributions of this work are as follows:

- Implementation of a GAN framework to generate synthetic keystroke sequences, trained on the DSL-StrongPasswordData dataset [7].
- Evaluation of synthetic data quality through:
  - Visual analysis (t-SNE, boxplots).
  - Statistical tests (Kolmogorov-Smirnov) to compare distributions with real data.
- Discussion of applications in biometric research and limitations for future improvement.

The paper is structured as follows: Section 2 reviews related work, Section 3 details the methodology, Section 4 presents results, and Section 5 concludes with future directions.

II. Related Work

Keystroke dynamics have been widely studied as a behavioral biometric for user authentication and intrusion detection [12]. However, research in this field faces significant challenges due to data scarcity [13], as collecting large-scale keystroke datasets raises privacy concerns and requires extensive user participation [14], [15]. This limitation has motivated recent explorations into synthetic data generation, with GANs emerging as a promising solution to overcome data availability constraints while preserving the statistical properties of genuine typing patterns.

Acien et al. [16] demonstrated this for keystrokes in the context of digital phenotyping, using a GAN to augment datasets for authentication systems while avoiding privacy violations. The synthetic data aims to support biomarker development for psychomotor impairments due to neurodegenerative diseases like Parkinson's. Their work revealed that synthetic data could maintain user-specific patterns (e.g., inter-key latencies) crucial for discriminative tasks.

An other study [17] investigates the use of Conditional GAN cGANs to generate synthetic keystroke dynamics data for user impersonation during the identification stage. The findings suggest that cGANs can successfully imitate user keystroke behavior, posing potential threats to keystroke authentication systems.

cGANs demonstrated also its efficiency to generate synthetic keystroke data aimed at impersonating authorized users in keystroke authentication systems [18]. The research considers scenarios where the sequence of typed words is either known or unknown, demonstrating that cGAN-generated keystroke patterns can effectively deceive authentication mechanisms.

Moreover, keystroke dynamics faces two core challenges that GANs address:

- Data Scarcity: Collecting large-scale keystroke datasets is labor-intensive. Traditional oversampling like SMOTE [10] fails to model complex timing distributions, whereas GANs generate plausible sequences [16].
- Privacy Concerns: Real keystroke data may leak sensitive information (e.g., typing content) [19]. GANs enable anonymous synthetic data sharing.

III. Methodology

A. Dataset Description

We utilized the DSL-StrongPasswordData dataset from Carnegie Mellon University [20], containing keystroke timing data from 51 subjects typing passwords in 50 sessions (8 repetitions per session). Each sample includes:

- Temporal features: Hold time (e.g., H.period), key-down-down times (e.g., DD.t.i), and key-up-down intervals
- Metadata: Subject ID (subject), session index (sessionIndex), and repetition number (rep)

B. Preprocessing Pipeline

The methodology begins with preprocessing the DSL-StrongPasswordData dataset by :

- Metadata Removal: Discarded non-temporal features (subject, sessionIndex, rep) to focus on typing patterns.
- Normalization: Applied MinMaxScaler(feature-range=[-1, 1]) to all features, ensuring consistent input ranges for GAN training

C. GAN Architecture

The GAN architecture consists of:

1) Generator: The Generator (G) is designed to transform random noise into synthetic keystroke dynamics data that mimic real typing patterns. Its architecture consists of:

- Input Layer:
  - Takes a 100 dimensional latent vector (z) sampled from a standard normal distribution $N(0, 1)$).
  - This noise vector provides the stochastic foundation for generating diverse samples.

- Hidden Layers:
  - First Linear Layer: Expands the latent vector to 256 neurons with LeakyReLU activation ($\alpha = 0.2$) to avoid dead gradients.

  - Second Linear Layer: Further expands to 512 neurons with another LeakyReLU.

  - Dropout (Optional): A dropout layer (p=0.3) can be added to prevent overfitting during training.

- Output Layer:
  - Projects the 512D representation to the 31-dimensional feature space (matching the real keystroke data).

  - Uses Tanh activation to ensure outputs are scaled to [-1, 1], consistent with the normalized input data.

2) Discriminator: The Discriminator (D) acts as a binary classifier to distinguish real keystroke data from synthetic samples:

- Input Layer:
  - Accepts 31-dimensional vectors (real or synthetic keystroke features).

- Hidden Layers:
  - First Linear Layer: Compresses input to 512 neurons with LeakyReLU.

  - Second Linear Layer: Further compresses to 256 neurons with LeakyReLU.

  - Dropout (Optional): Dropout (p=0.3) can regularize the discriminator.

- Output Layer:
  - Reduces to a single neuron with Sigmoid activation, outputting a probability (0 = fake, 1 = real).

D. Training Protocol

The models as show in Table I are trained adversarially for 1,000 epochs (batch size=64) using the Adam optimizer (lr=

TABLE I: Training parameters

Parameter	Value	Rationale
Batch size	64	Balances memory/quality
Epochs	1000	Early stopping checked
Optimizer	Adam	$\beta_1 = 0.5, \beta_2 = 0.999$
Learning rate	0.0002	Stable GAN convergence

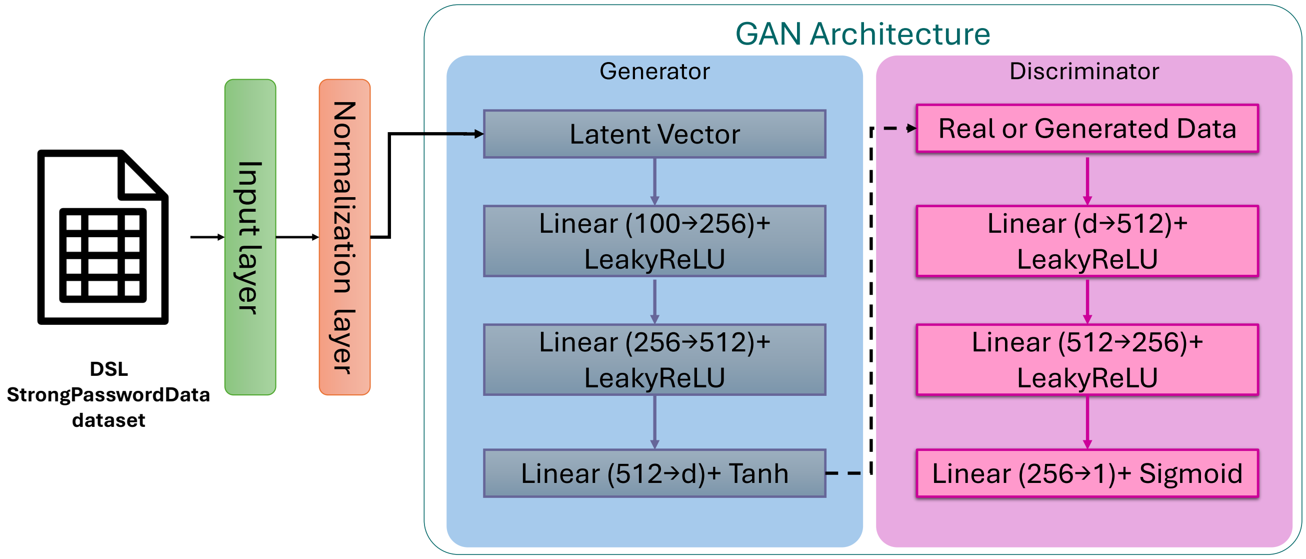

Fig. 1: Architecture

0.0002) and Binary Cross-Entropy loss, with the Generator aiming to fool the Discriminator and the Discriminator improving its detection accuracy.

The discriminator D aims to maximize classification accuracy for both real and generated samples. Its loss combines two terms:

$$\mathcal{L}D = \underbrace{\mathbb{E}x \sim p_{data}(x)[\log D(x)]}_{\text{Real Data}} + \underbrace{\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]}_{\text{Fake Data}} \tag{1}$$

IV. Results and Evaluation

The Gan architecture is trained to generate synthetic data. In this section, we will compare the generated data to the real one and discuss them.

A. Distribution Similarity

We compared the synthetic keystroke dynamics against real data using:

1) Feature-wise Boxplots: This method quantifies how well synthetic data preserve the distributional properties of individual keystroke features as depicted in Figure 2. The hold time feature (H.period) showed nearly identical quartile distributions:

Real Data:

- Median = 0.09 ms
- IQR = [0.07 ms, 0.11 ms]

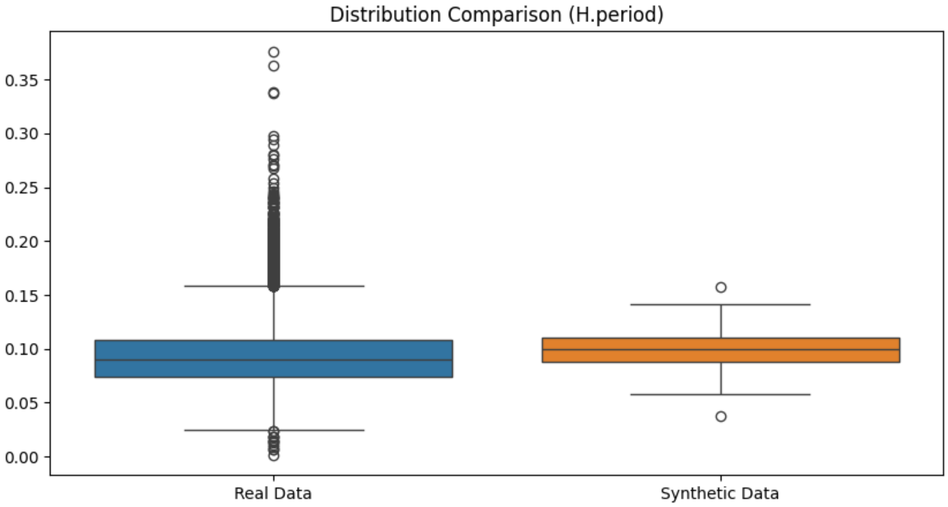

Fig. 2: Feature-wise Boxplots

Synthetic Data:
- Median = 0.10 ms
- IQR = [0.09 ms, 0.11 ms]

Relative Difference:
- $\Delta Median$ = 0.01 ms
- Relative Variation = 10.9%

The synthetic data's median hold time (0.10 ms) is very close to the real data (0.09 ms), with a minor absolute difference of 0.01 ms. However, the 10.9% relative increase suggests the GAN slightly overestimates typical hold times.

The Interquartile Range (IQR) of the synthetic data (IQR = 0.02 ms) is narrower than that of the real data's (IQR = 0.04 ms), indicating:

- The GAN generates less variability in hold times.
- Synthetic values are more concentrated around the median.

2) t-SNE Dimensionality Reduction: It helps to visualize high-dimensional structural similarity between real and synthetic data in 2D [21].

As demonstrated in Figure 3, the synthetic samples generated by the model are closely clustered with the real data points, indicating a high degree of similarity in structure and distribution. The overlap between the two types of data suggests that the generator has successfully captured the underlying patterns of human typing behavior. This visual evidence supports the quality and realism of the synthetic data, making it a viable substitute for real samples in subsequent machine learning tasks or biometric evaluations.

B. Statistical Validation

For stattistical validation we used the Kolmogorov-Smirnov (KS) Test [22]. It is a non-parametric statistical test used to compare two probability distributions.

The KS statistic measures the maximum distance between two empirical CDFs:

$$D_{n,m} = \sup_x |F_n(x) - G_m(x)| \tag{2}$$

where:

- $F_n(x)$ is the empirical CDF of the real data (sample size n)
- $G_m(x)$ is the empirical CDF of the synthetic data (sample size m)
- sup denotes the supremum

The p-value in the KS test estimates the probability of observing a test statistic (D_n, m):

$$p \approx 2 \exp\left(-2\frac{D_{n,m}^2 nm}{n+m}\right) \tag{3}$$

For GAN evaluation, it quantifies how well the synthetic keystroke data matches the real data distribution. The obtained results are depicted in Table II.

TABLE II: Statistical Comparison of `H.period`

Metric	**Value**	**Interpretation**
KS Statistic (D)	0.044	Negligible divergence
p-value	0.07	Small distributional differences

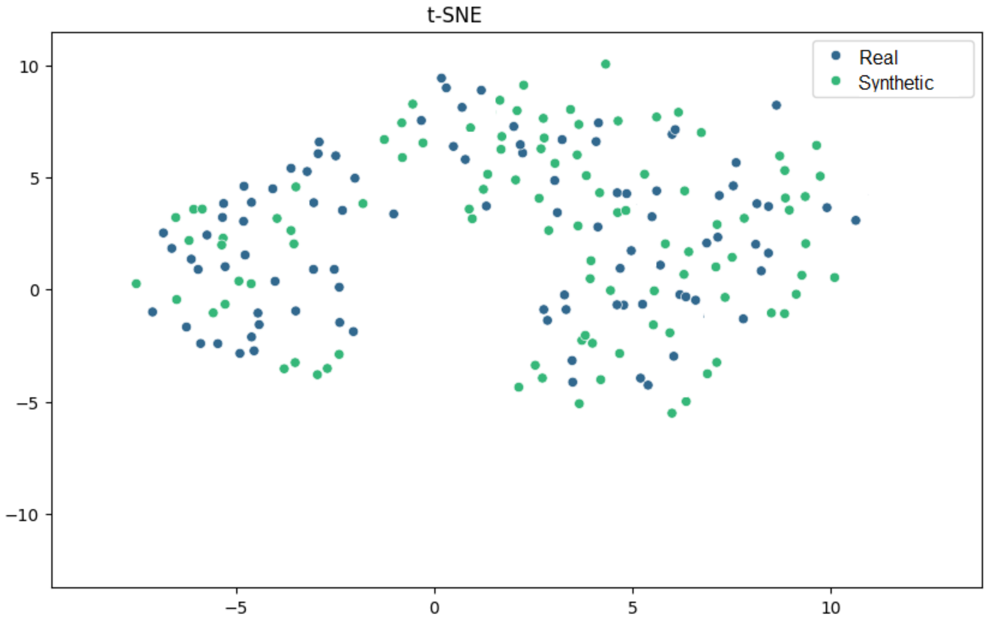

Fig. 3: t-SNE Dimensionality Reduction

D = 0.04397 suggests only a 4.4% max cumulative probability difference. P-value also demonstrates a small distribution difference.

V. Conclusion and future work

This study demonstrated the potential of Generative Adversarial Networks (GANs) for synthesizing keystroke dynamics data while preserving key statistical properties. Our framework successfully generated synthetic sequences that closely matched real data distributions, as evidenced by the obtained Marginal similarity. the experiments conducted A low Kolmogorov-Smirnov statistic (KS D = 0.044) for critical features like H.period and a small distibution difference with p-value =0.07. In addition, the t-SNE visualization showed Relative Variation = 10.9% between real and synthetic clusters.

To address current limitations and enhance generative performance, we propose the implementation of Temporal GANs: Replace feedforward generators with LSTM or Transformer blocks to model keystroke timing dependencies. We can also test the Conditional Generation: Incorporate user ID as a latent variable for user-specific synthesis (e.g., cGAN).

References

[1] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, "Authentication by keystroke timing: Some preliminary results," DTIC Document, Tech. Rep., 1980.

[2] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: A key to user identification," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 40–47, 2004.

[3] A. Mhenni, E. Cherrier, C. Rosenberger, and N. Essoukri Ben Amara, "Adaptive biometric strategy using doddington zoo classification of user's keystroke dynamics," in *14th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2018.

[4] A. Mhenni, E. Cherrier, C. Rosenberger, and N. Essoukri Ben Amara, "Double serial adaptation mechanism for keystroke dynamics authentication based on a single password," *Computers & Security*, vol. 83, pp. 151 – 166, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404818306059

[5] P. Kang, S.-s. Hwang, and S. Cho, "Continual retraining of keystroke dynamics based authenticator," *Advances in biometrics*, pp. 1203–1211, 2007.

[6] A. Mhenni, E. Cherrier, C. Rosenberger, and N. Essoukri Ben Amara, "User dependent template update for keystroke dynamics recognition," in *2018 International Conference on Cyberworlds (CW)*, 2018, pp. 324–330.

[7] A. Mhenni, C. Rosenberger, and N. Essoukri Ben Amara, "Keystroke dynamics classification based on lstm and blstm models," in *2021 International Conference on Cyberworlds (CW)*. IEEE, 2021, pp. 295–298.

[8] A. Morales, J. Fierrez, and J. Ortega-Garcia, "Towards predicting good users for biometric recognition based on keystroke dynamics," in *European Conference on Computer Vision*. Springer, 2014, pp. 711–724.

[9] E. Yu and S. Cho, "Keystroke dynamics identity verification—its problems and practical solutions," *Computers & Security*, vol. 23, no. 5, pp. 428–440, 2004.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[12] A. Mhenni, E. Cherrier, C. Rosenberger, and N. Essoukri Ben Amara, "Analysis of doddington zoo classification for user dependent template update: Application to keystroke dynamics recognition," *Future Generation Computer Systems*, vol. 97, pp. 210 – 218, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18331236

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

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

[15] D. Migdal and C. Rosenberger, "Analysis of keystroke dynamics for the generation of synthetic datasets," in *2018 International Conference on Cyberworlds (CW)*. IEEE, 2018, pp. 339–344.

[16] A. Acien, A. Morales, L. Giancardo, R. Vera-Rodriguez, A. A. Holmes, J. Fierrez, and T. Arroyo-Gallego, "Keygan: Synthetic keystroke data generation in the context of digital phenotyping," *Computers in Biology and Medicine*, vol. 184, p. 109460, 2025.

[17] I. Eizagirre, L. Segurola, F. Zola, and R. Orduna, "Keystroke presentation attack: Generative adversarial networks for replacing user behaviour," in *Proceedings of the 2022 European Symposium on Software Engineering*, 2022, pp. 119–126.

[18] I. E. Peral, L. S. Gil, and F. Zola, "Conditional generative adversarial network forkeystroke presentation attack," in *Investigación en Ciberseguridad Actas de las VII Jornadas Nacionales (7º. 2022. Bilbao)*. Fundación Tecnalia Research and Innovation, 2022, pp. 228–231.

[19] A. Mhenni, D. Migdal, E. Cherrier, C. Rosenberger, and N. Essoukri Ben Amara, "Vulnerability of adaptive strategies of keystroke dynamics based authentication against different attack types," in *2019 International Conference on Cyberworlds (CW)*, 2019, pp. 274–278.

[20] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *2009 IEEE/IFIP international conference on dependable systems & networks*. IEEE, 2009, pp. 125–134.

[21] A. Gisbrecht, A. Schulz, and B. Hammer, "Parametric nonlinear dimensionality reduction using kernel t-sne," *Neurocomputing*, vol. 147, pp. 71–82, 2015.

[22] R. H. Lopes, I. Reid, and P. R. Hobson, "The two-dimensional kolmogorov-smirnov test," 2007.