MixStrokes: MLP Mixers for Keystroke Dynamics Authentication

Abstract

Passwords and pin codes are ubiquitous in securing various digital applications, but their susceptibility to breaches has led to significant security concerns. With more than three billion passwords stolen annually, there is a pressing need for augmenting traditional security measures. Keystroke dynamics, a behavioral biometric trait that recognizes individuals based on their typing habits, has recently been explored as a non-obstructive secondary user verification method. In this paper, we present "MixStrokes," an extension of MLP mixers applied to the task of keystroke dynamics. Leveraging insights from recent advancements in free-text keystroke biometric authentication and the simplicity of the Mixer architecture, MixStrokes performs comparably to LSTM and transformer-based methods in terms of accuracy. Furthermore, it offers advantages in runtime efficiency and model complexity.

1 Dataset

This study employed two datasets from Aalto University: the *Dhakal et al.* dataset [3] and the *Palin et al.* dataset [7]. The former consists of over 5GB of keystroke data from 168,000 desktop users, while the latter comprises nearly 4GB from 260,000 mobile device users. Both datasets used controlled free-text collection [11], with participants typing memorized English sentences ranging from 3 words to 70 characters. The Dhakal dataset had all participants complete 15 sessions, whereas only 23% of the Palin dataset's participants did the same. Demographically, the Dhakal dataset included 72% participants with typing courses and 85% native English speakers, while the Palin dataset had 31% and 68%, respectively.

2 Methodology

Building upon previous works [1], the raw data captured in each session consists of a three-dimensional time series: keycodes, press times, and release times of the keystroke sequence, with timestamps in UTC format and millisecond resolution. Four temporal features are extracted from each sequence: Hold Latency (HL), Inter-key Latency (IL), Press Latency (PL), and Release Latency (RL), along with the keycodes as an additional feature, all of which are commonly used in keystroke systems [33]. Each sequence, therefore, has a shape of $N \times 5$ (N keystrokes by 5 features), where N is the length of the keystroke sequence. All features are normalized, with keycodes scaled between 0 and 1 by dividing by 255, and timing features converted to seconds, between 0 to 1, based on an average typing rate of 3 to 7 keys per second across the dataset.

The MLP Mixer [10], originally proposed for tasks in computer vision [12, 2], NLP [4], and audio processing [14], has demonstrated its efficacy across a range of challenges. In our study, we employ a standard MLP Mixer with hyperparameters specifically tuned for our keystroke dynamics dataset. The preprocessed feature vectors, derived from the keystroke sequences, serve as inputs to the initial layer of the MLP Mixer. To enhance the model's capacity to capture temporal relationships within the keystroke sequences, these input feature vectors are decorated with Gaussian range encodings

Preprint. Under review.

before being fed into the Mixer. The core idea behind the MLP Mixer is to use MLPs to mix information both locally (per feature) and across features. This is achieved through alternating layers of token-mixing and channel-mixing MLPs. The token-mixing MLPs allow for communication between different positions, while the channel-mixing MLPs mix information across the feature channels. This architecture, while simple, has been shown to achieve competitive results with more complex models, especially when trained on large datasets. For a comprehensive understanding and in-depth details of the MLP Mixer architecture, we recommend referring to the original paper [10].

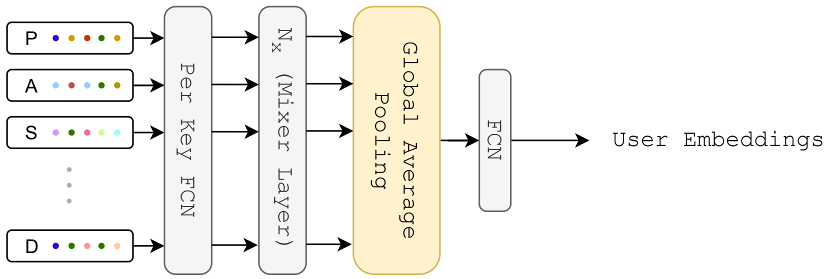

Figure 1: MixStokes - Block Diagram

Training the proposed model using a conventional softmax function presents certain limitations. Firstly, the number of classes must be predetermined during training, necessitating a retraining of the entire model when adding new users. Secondly, the softmax function tends to underperform in situations with high intra-class variance. To address these challenges, we employ contrastive loss and triplet loss. For the contrastive loss approach [5], two keystroke sequences, denoted as x_a and x_b, serve as an input pair to the model. This loss function measures the Euclidean distance between the model outputs. The goal is to reduce this distance for genuine pairs and increase it for impostor pairs. The triplet loss function [13], in contrast, facilitates learning by comparing both positive and negative pairs concurrently. In this context, a triplet consists of three samples from two distinct classes: The Anchor (A) and Positive (P) are keystroke sequences from the same subject, whereas the Negative (N) originates from a different subject.

3 Results

The evaluation of the model is conducted by comparing test sample *p* with a query sample *q*, either from the same subject (genuine match) or a different subject (impostor match). The test score is calculated using the Euclidean distances between test and query embedding vectors, with the formula: $\frac{1}{G}\sum_{g=1}^{G}\|\mathbf{f}(\mathbf{p})-\mathbf{f}(\mathbf{q})\|$, where G is the number of enrollment samples. Each subject has 5 genuine scores and K - 1 impostor scores, where K is the number of enrolled subjects. The evaluation metric used is the Equal Error Rate (EER), representing the point where the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR), calculated for each subject and averaged over all K subjects. Our preliminary findings are summarized in Table 1.

Table 1: Comparison of MixStrokes with state of the art literature for keystrokes dynamic verification on the Aalto Datasets

	Meta-architecture	**Score**	
		Desktop [3]	Mobile [7]
Lie et al. [6]	CNN + RNN	5.40	12.46
Acien et al. [1]	RNN (LSTM)	2.20	9.20
Stragapede et al. [9]	Transformer	1.94	3.84
Stragapede et al. [8]	Transformer	1.77	3.25
MixStrokes (with CL)	MLP Mixer	1.91	3.92
MixStrokes (with TL)	MLP Mixer	1.79	3.51

References

[1] A. Acien, J. V. Monaco, A. Morales, R. Vera-Rodriguez, and J. Fierrez. Typenet: Scaling up keystroke biometrics. In *Proc. of the IEEE/IAPR International Joint Conference on Biometrics (IJCB)*, 2020.

[2] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. In *European Conference on Computer Vision*, pages 620–640. Springer, 2022.

[3] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[4] Francesco Fusco, Damian Pascual, and Peter Staar. pnlp-mixer: an efficient all-mlp architecture for language. *arXiv preprint arXiv:2202.04350*, 2022.

[5] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[6] Jianwei Li, Han-Chih Chang, and Mark Stamp. Free-text keystroke dynamics for user authentication. In *Artificial Intelligence for Cybersecurity*, pages 357–380. Springer, 2022.

[7] Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. How do people type on mobile devices? observations from a study with 37,000 volunteers. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–12, 2019.

[8] Giuseppe Stragapede, Paula Delgado-Santos, Ruben Tolosana, Ruben Vera-Rodriguez, Richard Guest, and Aythami Morales. Typeformer: Transformers for mobile keystroke biometrics. *arXiv preprint arXiv:2212.13075*, 2022.

[9] Giuseppe Stragapede, Paula Delgado-Santos, Ruben Tolosana, Ruben Vera-Rodriguez, Richard Guest, and Aythami Morales. Mobile keystroke biometrics using transformers. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2023.

[10] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

[11] Ahmed Anu Wahab, Daqing Hou, Stephanie Schuckers, and Abbie Barbir. Utilizing keystroke dynamics as additional security measure to protect account recovery mechanism. In *ICISSP*, pages 33–42, 2021.

[12] Ziyu Wang, Wenhao Jiang, Yiming M Zhu, Li Yuan, Yibing Song, and Wei Liu. Dynamixer: a vision mlp architecture with dynamic mixing. In *International Conference on Machine Learning*, pages 22691–22701. PMLR, 2022.

[13] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[14] Jingfei Xia, Mingchen Zhuge, Tiantian Geng, Shun Fan, Yuantai Wei, Zhenyu He, and Feng Zheng. Skating-mixer: Long-term sport audio-visual modeling with mlps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2901–2909, 2023.