TYPE S#!T

User Authentication through Typing Patterns

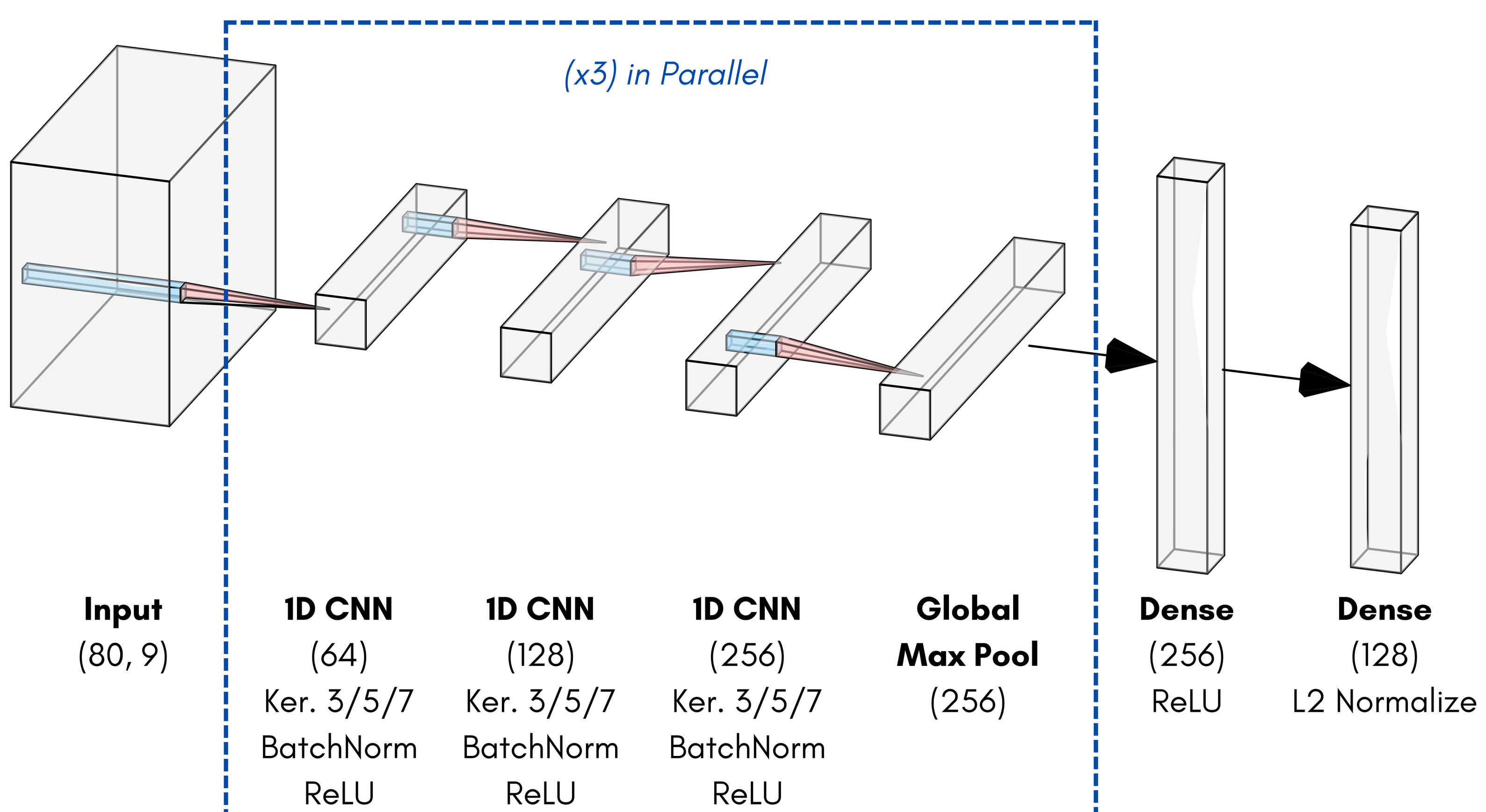
Amber Bajaj Hayzie Chu Eric Wu

Introduction

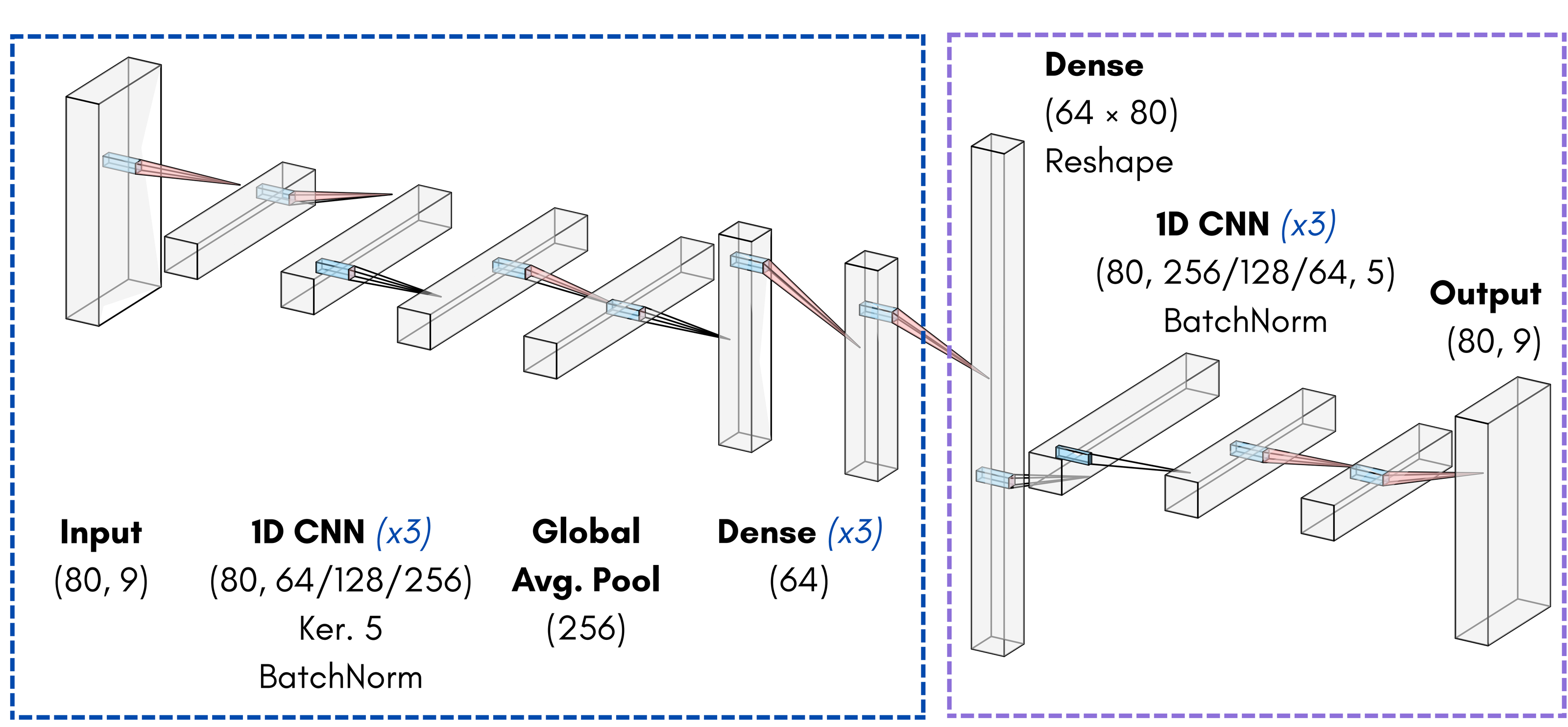
As digital security demands continue to rise, authentication methods must evolve beyond vulnerable password-based systems. Inspired by behavioral biometrics, we developed a typing-based **few-shot classifier** designed to ***verify whether a new typing behavior belongs to a given user***. Our approach uses a **1D CNN** with a **meta-learning** framework to capture typing patterns and quickly adapt to new users. We strengthened it by training on both real and **GAN-generated** keystroke data.

Architecture & Methodology

CNN Embedding



VAE



Data *University of Buffalo Keystroke Dataset*

148 subjects typed on the same keyboard layout in 3 typing sessions. Each session consists of roughly 3000-4000 keystrokes, each recording when (in ms), which (key code) and how (pressed/released). For each user, the typing sequence is separated into shorter windows of 80 events, which is more realistic for authentication, and has been shown in previous literature to improve accuracy. We then constructed digraph vectors for each consecutive pair of key presses, where each embedding contains 9 spatial and timing features, normalized to characterize an user typing pattern.

Meta-Learning Training Pipeline

(1) Pick a target user and create a support set from 2 writing samples

Randomly choose a user and sample k windows from their typing data

(2) Create *digraph vectors* for press-down-adjacent keys

[x-dist, y-dist, dist, key1 held, key2 held, down-down, up-up, down-up, up-down]

(3) Create an embedding of each support window via CNN & average to obtain the user's *prototype*

(4) Build query samples to make predictions on

Sample just q genuine and q imposter windows

(5) Compute Euclidean distance of prototype to query embeddings

Pass distance into an MLP (Verification Head), outputting P(same user)

Discussion

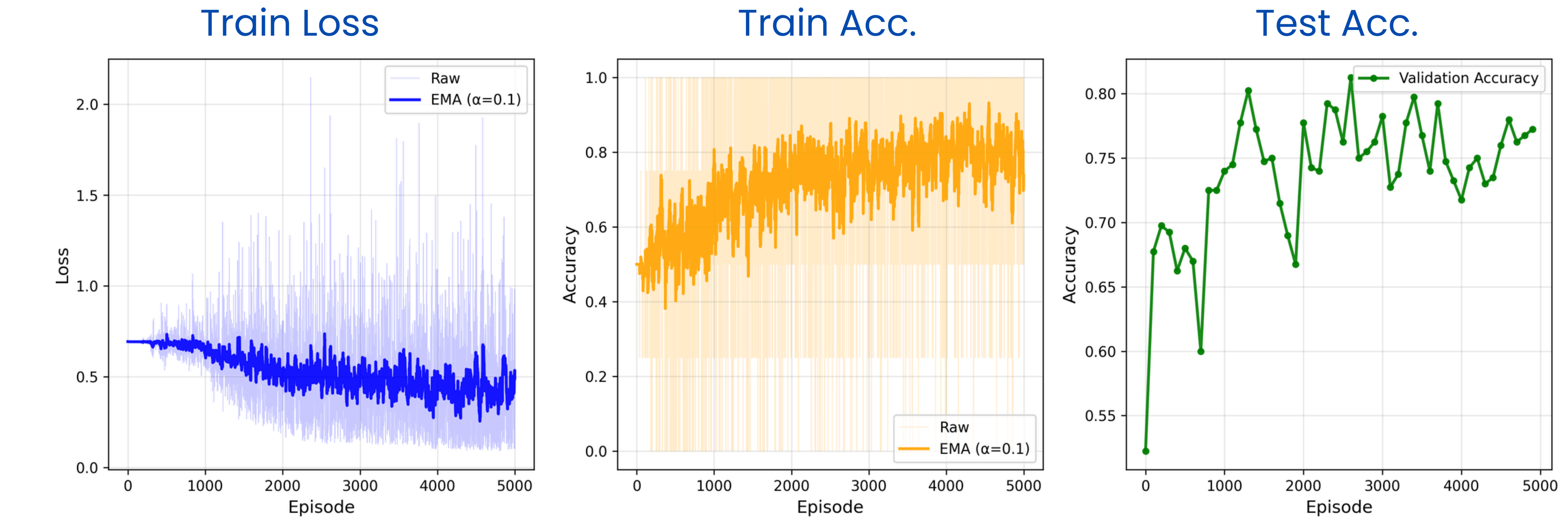
We only use and generate data on a single keyboard layout, since different layouts can lead to digraph variation and can confuse the model. Therefore, one future improvement can be to adapt the model to learn dynamic keyboard layouts. Future research might consider exploring how much each spatial or timing features have the greatest impact on verification accuracy. Digraph feature choices can be modified accordingly. Additionally, one could model the evolution of users' typing patterns overtime to ensure the system adapts to natural changes and long-time reliability of verification accuracy.

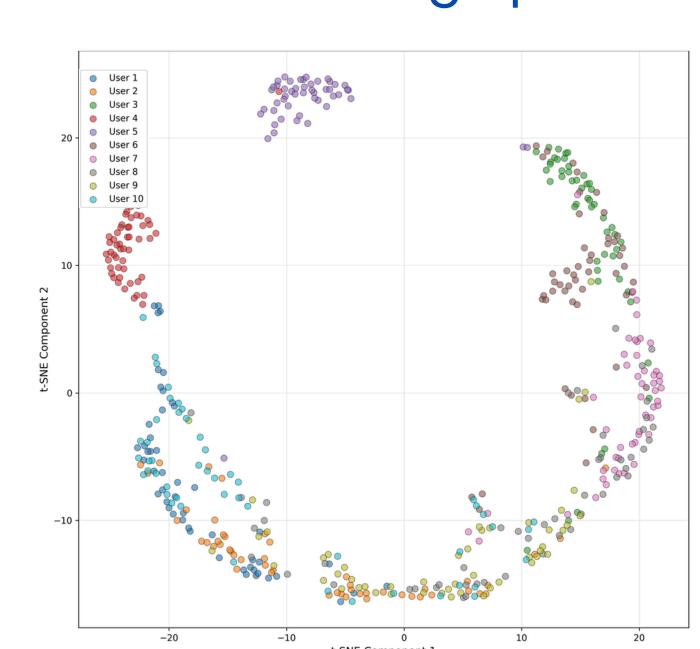
Results

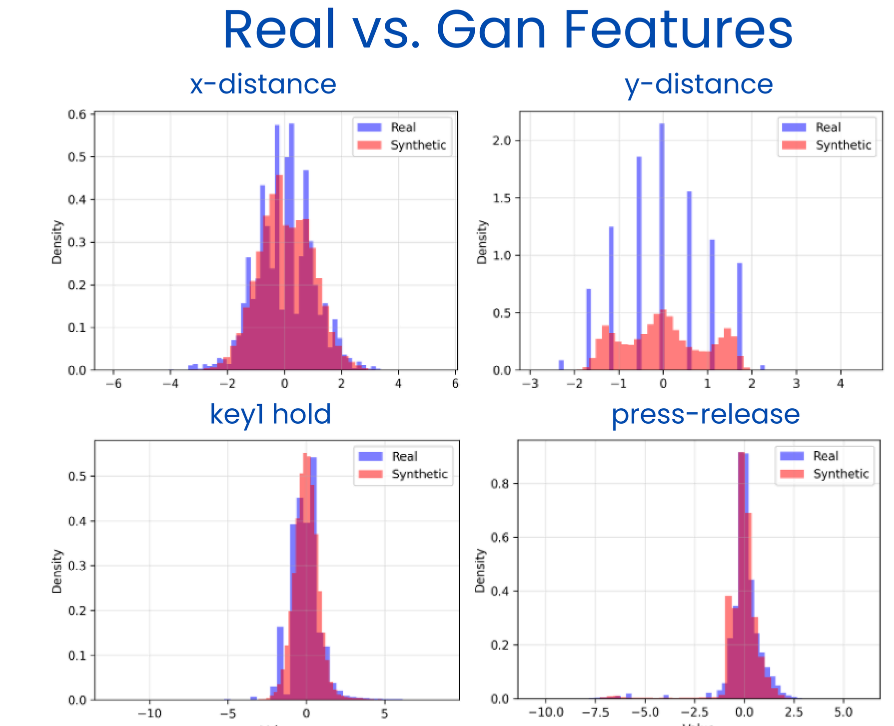
Test Accuracy on Mixed Data: 0.8060

Predicted \ True	True Genuine	True Imposter
Predicted Genuine	**918**	306
Predicted Imposter	82	**694**

window_size=80, q_query=15, k_shot=2, lr=0.001, kernel_sizes=[3, 5, 7] embedding_dim=128







GAN Wasserstein distance (μ): *0.090578*

GAN Discriminator Test Accuracy: *0.64*

Experimenting with various model parameters, we found that training on a window size of 80 with 2 support and 15 query samples worked well with our GAN and meta-learning pipelines. This result of a small window size is consistent with Lu et al. (2019). However, the meta-learning training was unstable with small parameter changes, suggesting a small range of stable values; further studies could explore various hyperparmeter tweaks. While meta-learning aims to generalize across users with minimal examples, the high variability in individual typing styles can make this generalization difficult. This may explain why the meta-training exhibited instability under slight parameter changes.

Sources

Benoit Azanguezet Q., Junie Toukêm T., Elie Tagne Fute. *"LSTM-based Free-Text Keystroke Dynamics for Continuous Authentication"*
A. A. Ahmed & I. Traore. *"Biometric Recognition Based on Free-Text Keystroke Dynamics."*
Xiaofeng Lu, Shengfei Zhang & Shengwei Yi. *"Continuous authentication by free-text keystroke based on CNN plus RNN."*