Keystroke dynamics is the time series data describing when and which keys are pressed and released as someone is typing on a keyboard.

By applying methods from behavioral biometrics, this data has been proven to be an effective unique identifier of a person and can therefore be used for authentication [1].

Plenty of previous work on this problem (e.g., using neural nets, Gaussian mixtures) but these methods fail to generalize to unseen users.

Can we find an approach that generalizes by utilizing metric learning?

Relevant metrics:
  FAR = False Acceptance Rate = FPR
  FRR = False Rejection Rate = 1 − TPR

Large scale typing dataset from a 2016 study [2].

Raw typing data from 148 users, both free text and transcribing for 150 minutes each.

Baseline Models

- **GMMs:**
  FAR = 14.6 %, FRR = 6.7 %

- **CNN Classifiers (OVR):**

Results for 5 random users

<table>
<thead>
<tr>
<th>error type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>17 %</td>
<td>8 %</td>
<td>9 %</td>
<td>23 %</td>
<td>0.1 %</td>
</tr>
<tr>
<td>FRR</td>
<td>8 %</td>
<td>9 %</td>
<td>6 %</td>
<td>11 %</td>
<td>23 %</td>
</tr>
</tbody>
</table>

Methods were trained on data from 30 random users.

Test data sampled from the same 30 users as well as from 30 random, and previously unseen, users.

Two prediction methodologies:
- Predict by comparing to the embedding of a single reference sample.
- Compare with the embeddings of five different reference samples and predict based on the majority vote.

Key idea: Learn an embedding of typing samples into a lower dimensional space, where samples from the same user are close and samples from different users are distant.

**Triplet learning:** Form triplets (Anchor, Positive Negative), where A, P are samples from the same user, and N is a sample from a different user.

Train an embedding model using the triplet loss: $L = \max(\|A_e - P_e\|_2 - \|A_e - N_e\|_2 + \alpha, 0)$

Problem: Most triplets already yield zero loss which results in small gradients.

Solution: Online mining for semi-hard triplets.

When the tower model is trained, we train an SVM on the elementwise difference between embeddings:

- Online triplet-mining in order to improve convergence.
- Inception-style embedding network in combination with the choice of feature representation results in embeddings that accurately represent the data without overfitting.

Future work: Extend the system to work well on users that switch between different keyboards.

References


Available: https://cubs.buffalo.edu/research/datasets

Discussion and Future Work

- **t-SNE of training data**
- **t-SNE of unseen users**

Introduction

Feature Representation

- **Digraph:** Sequence of two key presses.
- **Digraph feature representation:**
  $d = [KD, H_1, H_2, PP, RP] \in \mathbb{R}^5$
  $KD = \text{Distance between keys.}$
  $H_1 = \text{Hold time of } i^\text{th} \text{ key.}$
  $PP = \text{Press-to-press time.}$
  $RP = \text{Release-to-press time.}$

Results

- **Key distance model**
- **BatchNorm**
- **Conv1D Filters:** 16
  Kernel: 3
- **Inception**
- **GlobalAvgPool**
- **Softmax**
- **Embedding**
- **SVM**

Method

- **Sample KD:** Y → O = (2, 2) + (10, 7) = 3
- **A typing sample** is a sequence of digraphs.
  We use samples of length 100.
- **One sample:** $x(t) \in \mathbb{R}^{5 \times 100}$

Dataset

- **Large scale typing dataset from a 2016 study** [2].

- **Raw typing data from 148 users, both free text and transcribing for 150 minutes each.**