Introduction

- 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 = FNR

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.

<table>
<thead>
<tr>
<th>key</th>
<th>event</th>
<th>time stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>KeyDown</td>
<td>63578429797235</td>
</tr>
<tr>
<td>E</td>
<td>KeyDown</td>
<td>63578429797313</td>
</tr>
<tr>
<td>O</td>
<td>KeyUp</td>
<td>63578429797313</td>
</tr>
</tbody>
</table>

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>
<th>FAR</th>
<th>8%</th>
<th>9%</th>
<th>23%</th>
<th>0.1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>17%</td>
<td>8%</td>
<td>9%</td>
<td>23%</td>
<td>0.1%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRR</td>
<td>8%</td>
<td>9%</td>
<td>6%</td>
<td>11%</td>
<td>23%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Feature Representation

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

Key distance model

Sample: KD: Y \rightarrow O = (2-7) \times (10-7) = 3

- A typing sample is a sequence of digraphs. We use samples of length 100.
- One sample: \( x(t) \in \mathbb{R}^{1 \times 100} \)

Method

- **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 network using the triplet loss:**
  \[ L = \max(\|A_N - P\|_2 - \|A_N - N\|_2 + \alpha, 0) \]
- **Problem:** Most triplets already yield zero loss which results in small gradients.
  **Solution:** Online mining for semi-hard triplets.

Results

- Models 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.

<table>
<thead>
<tr>
<th>Users</th>
<th>error type</th>
<th>single</th>
<th>majority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seen</td>
<td>FAR</td>
<td>8.69%</td>
<td>7.63%</td>
</tr>
<tr>
<td></td>
<td>FRR</td>
<td>12.29%</td>
<td>6.61%</td>
</tr>
<tr>
<td>Unseen</td>
<td>FAR</td>
<td>12.05%</td>
<td>10.14%</td>
</tr>
<tr>
<td></td>
<td>FRR</td>
<td>19.75%</td>
<td>15.26%</td>
</tr>
</tbody>
</table>

Discussion and Future Work

- We see that our approach is on par with other methods in terms of FAR and FRR.
- This method generalizes reasonably well to users that were not in the training set. Using a single sample from a previously unseen user, we can output decently accurate predictions. The performance is then only further improved as more samples are collected.
- Key aspects of the approach:
  - 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