Using Keystroke Dynamics to Authenticate a User Based on their Typing

Jack Francis

April 10, 2022

Abstract

Hello

Contents

Chapter 1

Introduction

Chapter 2

Survey of Literature

Chapter 3

Design and Implementation

3.1 Data gathering and Forming

3.1.1 Forming Words

3.1.2 Data Selection

3.2 KD Signal

Once the raw keystroke data has been formed into words, the next step is to transform this from a series of keystroke pairs into numerical data that can be used by later algorithms such as Dynamic Time Warping (DTW) and the correlation coefficient. The best way to do this is to transform the data into a measure of how many keys are being held down at a particular point in time. The resulting output is known as a key down signal (KDS).

To convert a word into a key down signal, the start and end times of the word being transformed are used. Assume that w is the array of times that key actions occur in a particular word. w_1 being the time of the first action and w_n being the time of the last action. This part will loop through all timestamps until it ends with the final time which is denoted by w_n. The accuracy of this step is paramount as it is the level of detail that is the base accuracy for the rest of the steps. A higher accuracy means that the program will check more data points within this range at the cost of reduced performance as the level of points being checked increases. The current level of this is set to 4 decimal points which seems to provide a good balance between accuracy and performance. However this is customisable.

$$KDS(w) = \sum_{i=w_1}^{w_n} K(w_i) \tag{3.1}$$

K is the next step of the algorithm. n is the array of key presses that is used in the previous step. This step of the algorithm iterates through all the key presses and uses a modified Heaviside step function which is run twice per pair with the time input from the previous denoted as t and the 'down' action denoted as n_i^1 and the 'up' denoted as n_i^2. The value returned by the Heaviside step function with the 'up' action is subtracted from the value returned by the 'down' function.

$$K(t) = \sum_{x=1}^{n_k} h(t, n_i^1) - h(t, n_i^2) \tag{3.2}$$

The reason for this subtraction is that the purpose of this measure is to return the number of keys pressed down at the time input. Once a key has been released it is essential that the key is removed from the measure. For example, if a pair exists with down action time being at 1 second and up action time being 1.1 seconds. At time, 1.5 the equation will equal $1 - 1 = 0$. However, if the time put in is 1.05 then the equation will be $1 - 0 = 1$ which indicates that one key was being held down at this particular time.

For every time input, each pair is checked with the sum of all the results stored in a dictionary along with the time input as the key. It's this dictionary that forms the KD signal and is used in further steps.

3.2.1 Heaviside Step

This is the bottom layer of the KD signal algorithm. It is a modified version of the Heaviside Step function.

$$h(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 > x_2 \\ 0.5 & \text{if } x_1 == x_2 \\ 0 & \text{if } x_1 < x_2 \end{cases} \tag{3.3}$$

The modification done is very simple, the only change is the addition of a third case which tests if the two times are equal to one another. Due to the nature of the use case for my project, there is a relatively high chance that the two times are equal to one another. In this case this means that the user at this time is currently in the process of performing that action whether that be pressing or releasing the key.

3.3 Dynamic Time Warping

3.3.1 Path

3.3.2 Cost Matrix

3.4 Validation Measures

- Talk about selecting values
- What effect does semantics have?
- Weighting of Euclidean vs correlation
- Chosen auth method
- In all cases what happens?
  - If not seen word before

  - If all validated
  - If all bar one are validated
  - etc

3.4.1 Euclidean Distance

3.4.2 Correlation Coefficient

3.4.3 Semantics??

3.5 Training

- first x vs ded training

3.6 Update

- Update everything

3.7 Storage

- Compression - file sizes too large
- Keyboard storage info

3.8 Pausing

- Uses auth method
- why? Sensitive info
- Implementation processes??

Chapter 4

Results and Discussion

- Test Results
- Calc and use FP, FN, TP, TN - get a percentage
- Discuss in relation to validation measure
- Mention struggling with small words maybe???
- Speed, security??

Chapter 5

Critical Appraisal

1. Summary and crit analysis
   - System works very well - provide examples using test data??
   - System is lightweight and secure
   - Compared to og planned, system is more complicated
   - Struggles with smaller words - less data points
   - NEED TO COME BACK TO THIS, NOT DETAILED AT ALL
2. Impact
   - Benefits
     - better security in combo with other sec methods
     - Lightweight and users won't notice
     - Doesn't spy on people due to only storing KDS and can turn off when user is doing something sensitive
     - Can be adapted to be used in the real word easily
   - Risks
     - Greater surveillance
     - Could easily be adapted maliciously - key logger
     - NEED MORE
3. Personal Development
   - Maths, maths, maths
   - Further git knowledge???
   - Exp Project Development
   - MORE

Chapter 6

Conclusion