Key Stroke Dynamics

Goal

The aim of our course project is to explore the possibility of using keyboard typing for user classification. The end project would be web resource that asks a user to type a phrase multiple times, after that, it saves the needed data to database, and later on, we use that information to train out models. After that our model would be able to classify a person who typed those phrases versus other people.

Features that we use to train our models consists of the sum of two or more base features. That base features are times between keydown and keyup, and times between keyup and next keydown.

Implementation

To get better performance of the model we decided to cluster several models in one. All the models get the same portion of data(whole data) and make their own prediction based on the data, implementation and unique model`s calculations. At the next stage models vote for the result. Their votes have different coefficients based on their solo performance. For measuring solo performance, we use FFR and FAR.

Models

As for now, we have \_\_\_\_ models.

First one is called statistical or z-score. The main idea is that we create a vector of the mean of all feature of one user, vector of deviation of one user. After that, we read the vector that we want to check. For each feature of that vector, we calculate z-score by the formula: where x – feature values, y – mean of that feature and s – deviation for that feature. After that, we mark each feature as one if it 1.96 deviations away from mean or 0 otherwise. That digit is, later on, equates to the coefficient \* amount of features, and if the digit is bigger than we return true, else false. The right part of the equation  represents a kind of similarity threshold. Information taken from [1], full description [2].

The second one is the euclidian-cityblock model with loop threshold. The training stage of that model is to find the mean vector of one user’s data, and if the euclidian-cityblock distance to it from a tested vector is less than threshold calculated by loop, then classify as true else – false. Loop takes i-th vector from a dataset, and calculates distance from it to mean vector of other vectors; algorithms repeat that i times and returns the mean of all distances. [3]

Nearest Neighbor with Mahalanobis distance

The idea is that we take data of previous entrances of the current user and separate it into train and test. Also, we take some imposter data from other users. After that, as a training stage, we take an inverse of the covariance matrix of the transposed train part of the dataset( ). In the testing stage, we count a Mahalanobis distance between every user's test data and imposter test data and save it into two arrays.

Manhattan distance

First steps are the same as in the previous model. In the training stage, we simply calculate the mean vector of train data. In the testing stage, we calculate manhattan distance between users with mean vector and imposter data with mean vectors and save it into two arrays.

Manhattan filtered

First steps are the same as in the previous model. In the training stage, we also calculate the standard deviation vector and drop all the vectors distanced away from every element of standard deviations vector multiplied by 3.

Site

The second part of our project is a web-based online platform, the site on which users will be verified by keystroke dynamics. The site will have a registration form with a required username and some sentence which will be used for verification in the future. Users will be asked to type the sentence. The results will be saved in the database, connected to our site and used for user verification.  We used HTML, CSS to build the look of the web page, jquery to get the values of keypressings, flask to run server and mysql to keep the data.

References:

https://www.cs.cmu.edu/~maxion/pubs/KillourhyMaxion09.pdf

S. Haider, A. Abbas, and A. K. Zaidi. A multi-technique approach for user identification through keystroke dynamIcs. IEEE International Conference on Systems, Man and Cybernetics, pages 1336–1341, 2000.

https://github.com/deepakjayaprakash/Keystroke-Authentication/blob/master/AI\_Paper.pdf