CMPE 272 Enterprise Software Platforms

SPRING 2018

Keystroke Signature Project Report

Submitted by: Team 7

Name	Email Id	Student ID
Huy Hunh	Huyhuynh1228@gmail.com	009405513
Deepti Srinivasan	Deepti.srinivasan@sjsu.edu	12557909
Nikhil Agrawal	Nikhil.agrawal@sjsu.edu	012525123

Course Instructor:

Prof. Rakesh Ranjan.

Table of Contents

Abstract

Keystroke Signature project demonstrates how keystroke dynamics - the unique typing patterns of users can be used as a signature to identify genuine users from imposters. Keystroke signature fuses the simplicity of passwords with increased reliability from biometrics. Unlike other biometrics like IRIS, face, fingerprints etc which need special hardware infrastructure, keystroke biometrics are economical to implement and can be easily integrated into the existing computer security systems. They help augment the existing security infrastructure by being part of a multilevel authentication system. Using Machine learning algorithms, a high accuracy rate in detecting imposters has been demonstrated in this project. The machine learning model is trained with the typing patterns of the subjects. Then it is provided with test data with patterns from the subject as well as from imposters posing as the subject. The model demonstrates the ability to discern genuine users from imposters based on the test pattern's similarity to the trained model for the subject.

Introduction

An estimated 35% of users in US have weak passwords amounting to 65% of the data breaches. Compromised passwords and shared accounts are frequently exploited by both external attackers and insiders. If we had some means, other than knowledge of a password, with which to identify exactly who is logging into an account, and to discriminate between the genuine user of an account and an impostor, we could significantly curb these security threats. This is the reason why we need to have something more intrinsic to the user that cannot be easily impersonated. This is where keystroke dynamics – the unique typing patterns of users is useful. With Keystroke dynamics, an imposter who gets hold of a compromised password will be detected even after he has successfully authenticated himself. This is because their typing rhythms differ significantly from a genuine user.

Background and Objectives

Keystroke dynamics has been considered as a strong behavioral biometric-based authentication method. Unlike other biometrics like IRIS, face, palm, fingerprints etc. which need special hardware infrastructure, keystroke biometrics are economical to implement and can be easily integrated into the existing computer security systems. They are also considered less intrusive as compared to the other physical biometrics. Furthermore, they function just like the DNA - unique and hard to impersonate.

Research in the field of keystroke dynamics suggest many anomaly detection algorithms for detecting imposters. The goal of the ongoing and past research has been to determine whether a detector is sufficiently dependable to be put into practice.

In this project , we focus on a particular technique: using Machine learning algorithms (detectors) to analyze password-typing times of users. The objective of the project is to demonstrate how keystroke data can be used as an identifier and can help the security system discern between genuine users and imposters. The machine learning model is trained with the typing patterns of the subjects .Then it is provided with test data with patterns from the subject as well as from imposters posing as the subject. The model then demonstrates the ability to discern genuine

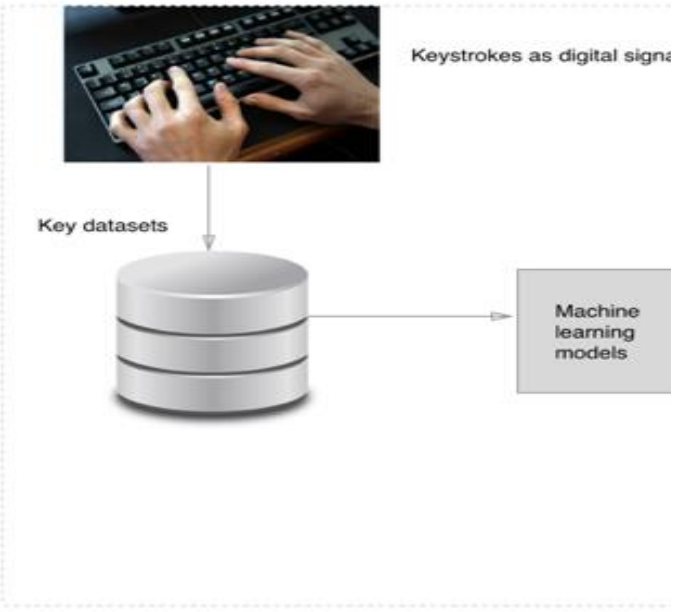
users from imposters based on the test pattern's similarity to the trained model for the subject. Through the Keystroke signature project ,the machine learning models have been able to distinguish between genuine users and imposters with a high rate of accuracy and precision as noted and a low rate of misclassification.

Approach and Methodology

The keystroke signature project can be divided into 3 main stages:

1. Data collection
2. Machine learning model training
3. Testing and evaluation
4. Model deployment and evaluation

The following diagram depicts the high-level project flow of Keystroke Signature project.



1.Password data collection:

The keystroke data is the timing information as follows:

- Hold time – time between press and release of a key.
- Keydown-Keydown time – time between the pressing of consecutive keys.
- Keyup-Keydown time – time between the release of one key and the press of next key.

Data collection application:

Below is a snapshot of the data we have collected :

H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i	H.i	DD.i.e	UD.i.e	...	DD.a.n
0.119	0.272	0.153	0.103	0.208	0.105	0.103	0.288	0.185	...	0.328
0.119	0.272	0.153	0.103	0.216	0.113	0.103	0.287	0.184	...	0.312
0.127	0.352	0.225	0.143	0.240	0.097	0.143	0.344	0.201	...	0.297
0.122	0.315	0.193	0.127	0.200	0.073	0.111	0.400	0.289	...	0.200
0.143	0.232	0.089	0.119	0.208	0.089	0.127	0.264	0.137	...	0.184
0.135	0.424	0.289	0.110	0.184	0.074	0.127	0.296	0.169	...	0.151
0.136	0.216	0.080	0.103	0.200	0.097	0.120	0.241	0.121	...	0.249
0.117	0.287	0.170	0.102	0.183	0.081	0.119	0.248	0.129	...	0.287
0.119	0.232	0.113	0.103	0.192	0.089	0.111	0.256	0.145	...	0.168
0.119	0.280	0.161	0.119	0.232	0.113	0.119	0.264	0.145	...	0.328

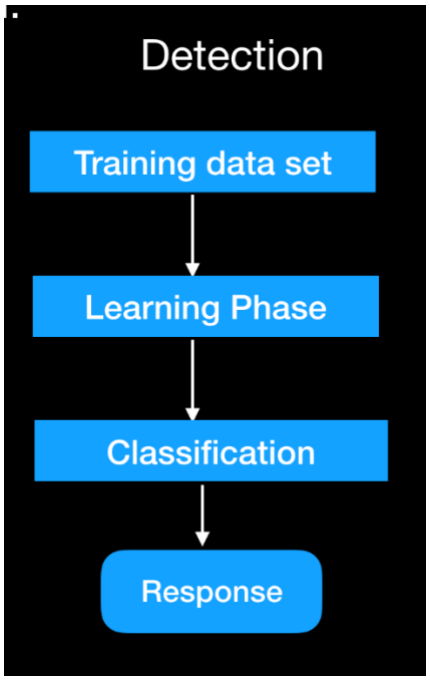
2.Model training:

For the scope of this project 3 machine learning algorithms have been trained and evaluated for the keystroke data:

- ❖ Support vector machines (SVM)
- ❖ Logistic Regression
- ❖ K-Nearest Neighbor (KNN)

These algorithms have been chosen due to their ability to be able to work with complex data and give classifications with low error rate and high accuracy.

The below diagram depicts how the models work as a part of the project methodology:



3.Model testing and Evaluation:

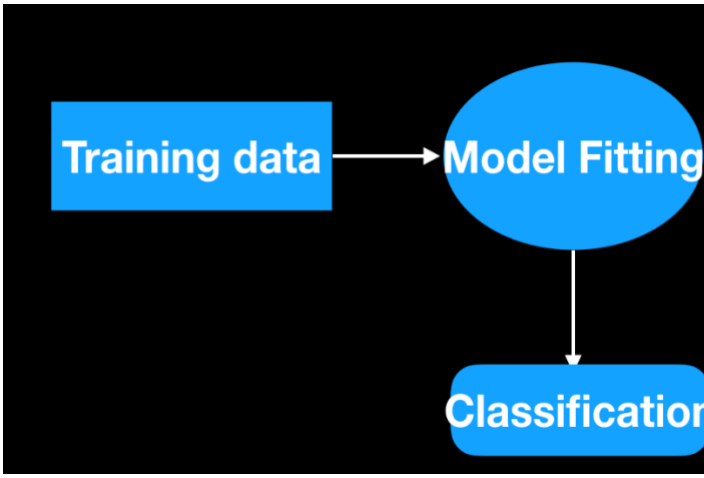
Two methods have been used to evaluate the trained detector models:

1. Train-Test- split:

- Here we split the dataset into 2 parts: training set and testing set.
- Train the dataset with train dataset.
- Test the model on the testing set and evaluate how well we did.

2. Cross -Validation:

- Split the dataset into K equal partitions (or "folds").
- Use fold 1 as the testing set and the (k-1) dataset as the training set.
- Calculate accuracy.
- Repeat steps 2 and 3 'K' number of times, using a different fold as the testing set each time.
- Use the average testing accuracy as the estimate of out-of-sample accuracy.



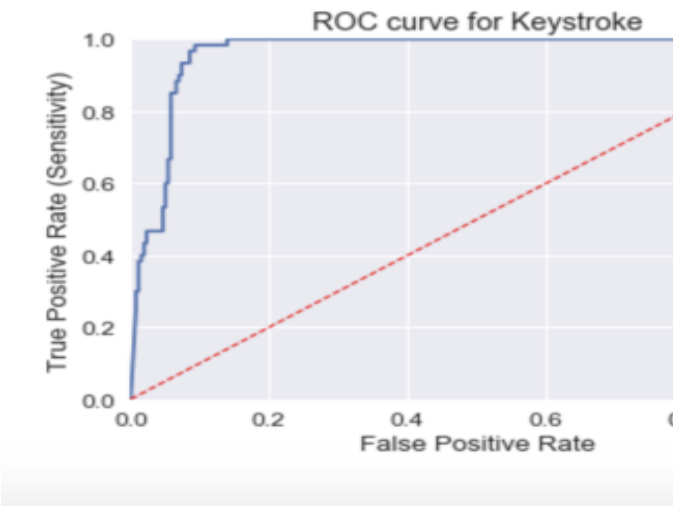
4.Model Deployment/Evaluation on IBM Watson:

Findings /Analysis

To Evaluate the result of the models and the prediction the following evaluators were used:

ROC Curve: ROC curve is the plot of the probability of classifying correctly the positive examples against the rate of incorrectly classifying true negative examples. In this sense, one can interpret this curve as a comparison of the classifier across the entire

range of class distributions. Measure of classifier performance across class distribution.

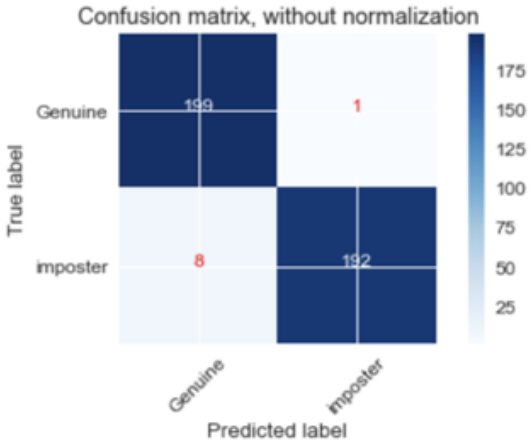


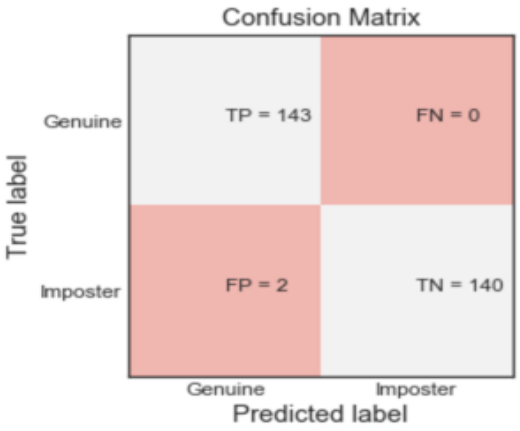
AUC : AUC is an abbreviation for area under the curve. It is used in classification analysis in order to determine which of the used models predicts the classes best. An example of its application are ROC curves. Here, the true positive rates are plotted against false positive rates

Error Rate : The error rate of any classifier is typically the proportion of classifications it gets wrong. Lower the error, better is the performance of the model

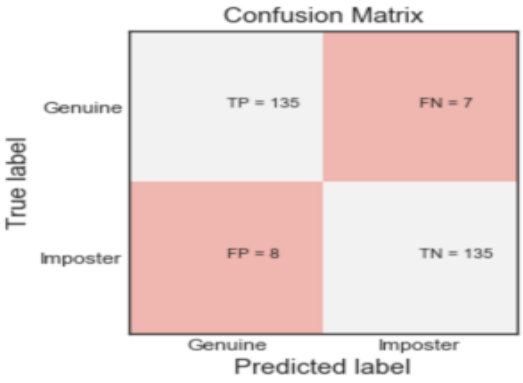
Confusion Matrix: It is a tabular representation of predicted value vs the expected value of the model. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).[2] The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another)

SVM Logistical Regression





KNN



Accuracy: (TP +TN) / P+N : Accuracy is the ratio of the correct prediction and the total samples used for prediction.

Precision: TP/ (TP+FP) : Precision is the ratio of true positive and the sum of true positive and false positive.

Final Best Results:

	Error rate	Accuracy	Precision
SVM	0.0225	0.97	0.978
Logistic regression	0.01	0.99	0.99
KNN	0.06	0.94	0.95

Conclusion:

Our objective in this project has been to collect a keystroke data set, train machine learning models, and measure their ability to distinguish between genuine users and imposters. After evaluating all the 3 different models, Logistical regression performed slightly better than KNN and SVM. Overall all the 3 models have very high accuracy rate & precision and extremely low error rate. We have been able to deploy the logistic regression model on IBM Watson service and the model has demonstrated the ability to discern between genuine and imposter subjects with high accuracy rate.

It has been demonstrated that keystroke dynamics where we identify users based on habitual typing rhythm patterns. It has already been shown that keystroke rhythm is a good sign of identity. Keystroke signature is unique to each individual and holds huge potential as part of a multilevel authentication system and intrusion detection. Keystroke signature can also be useful in fraud detection especially during online examinations where imposters can be easily detected.

References:

[1] Comparing Anomaly- detection Algorithms for Keystroke Dynamics - Kevin S Killourhy and R.A.Maxion - Aug 2009

[2] Support Vector Machines and Area Under ROC curve by Rakotomamonjy (Sept 1, 2004)

[3]Utilizing Keystroke Dynamics as an Additional Security Measure to Password Security in Computer Web based Applications - A Case Study of UEW International Journal of Computer Applications International Journal of Computer Applications Volume 149 –No. 5, September 2016 volume 149 –No.5, September 2016

[4] Keystroke dynamics as a biometric for authentication - Fabian Monrose and Aviel D. Rubin
- AT&T Labs-Research, Florham Park, NJ, USA – 3 March 1999.

[5] www.kaggle.com