Does Your Phone Know Your Touch?

Stanford University

John Peruzzi, Phillip Andrew Wingard, David Zucker

{jperuzzi, pwingard, zuckerd}@stanford.edu

Introduction

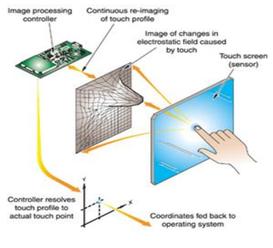
This project is motivated by potential advances in gesture based smartphone security techniques. We implemented various machine learning models using touch screen gesture data in an attempt to to accurately classify users based on gesture patterns. We leveraged **supervised classification** and **anomaly detection** approaches with varying success rates.

Data Collection

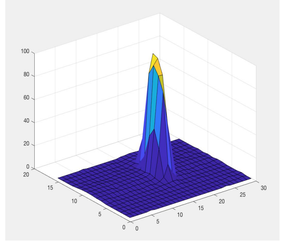
- A capacitive sensor array is used to collect user smartphone gesture data.
- 1515 **taps**, 557 drawn **circles**, and 271 **random** finger movements are measured from four users.
- A constant number of **frames** are selected for each training example
- On each frame, we isolate an **nxn window** of coupling (pressure) values around the central max value.
- Features include **pressure**, **duration**, and **x, y velocity** of finger movement per frame
- Each training example has feature size (for *nxn* window and *f* frames) of:

  $f*(n^2) + 2f + 1$

Capacitive Sensor Array



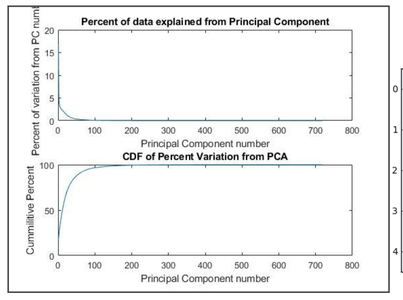
A Single Frame

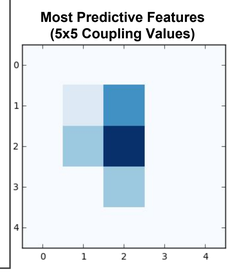


Methodology

I. PCA and Feature Analysis

We conducted PCA and recursive feature estimation (RFE) with 10K cross fold validation to select appropriate model features, particularly in situations where the feature size was significantly larger than the number of examples. We found that PCA could drastically reduce the required feature size and worked very well with Taps data.



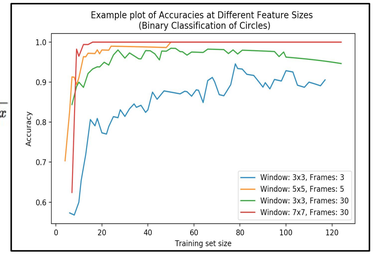


II. Logistic and Softmax Regression for Binary and Multinomial Classification

We used binary and softmax classification with our four user classes. We determined optimal feature sizes and regularization levels.

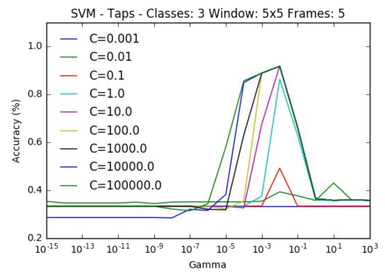
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

(logistic hypothesis)



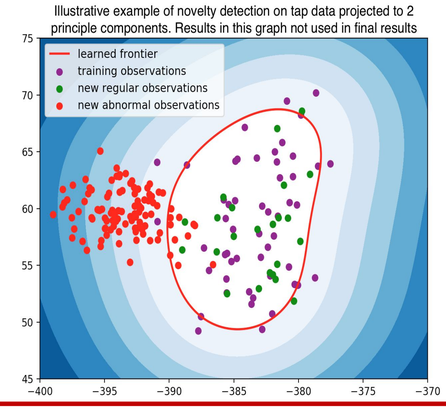
III. SVM for Binary Classification

We ran RBF support vector machines with hyper-parameter grid search on the C (1e-03 thru 1e+10) and gamma (1e-15 thru 1e+03) parameters to determine the optimal configuration for each dataset.



IV, Multiple Gaussians Anomaly Detector

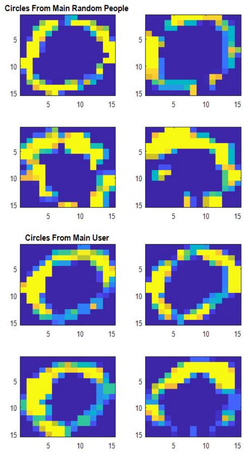
MG is trained using only a single users data. This method is inspired in part to how an actual phone will only collect data from the owner



Discussion

Our classification models, particularly logistic regression, were very successful. Our Gaussian mixture model for the real-world task of anomaly detection was also successful, however, we were much better at detecting the correct user class than passing anomalies. The success of logistic regression (especially without regularization) indicates that our data is repetitive and not entirely realistic. To mitigate this, we recommend future experiments collect a larger dataset across more users in a more organic manner. Our success in leveraging classification models to identify smartphone users indicates the plausibility of using touch screen data for improved security as the data is a sufficient distinguisher. Additionally, less repetitive data might allow for Gaussians which are less rigidly fit, potentially improving further anomaly detection scores.

Results



Model	Taps (Train)	Circles (Train)	Random (Train)	Taps (Test)	Circles (Test)	Random (Test)
Sample Size (4 users)	1060	389	145	455	168	63
Logistic (Binary)	100%	100%	100%	100%	100%	96.9%
Logistic (Softmax)	100%	100%	N/A	94.9%	100%	N/A
SVM	100%	100%	62%	96%	48%	54%
Gaussian Mixture (% pass regular, % detect anomaly)	78% (No anomaly in train)	100%	100%	78% - Pass User, 99% - Reject Anomaly	98%, 60%	98%, 25%