Authorization of Cross-Device Integration

By: Dr. Jun Kong, Aaron Yang, and Jake Leicht

Introduction

Cyber security is a vastly growing concern in our digital modernizing world, and everyday results in more data breaches. The need for increased digital security and various methods of doing so is a highly needed research and developing industry. An article written by John Zorabedian of Security Intelligence, estimated that the average cost of a security breach resulted in a loss of $4.24 million. Even just recently during the Tokyo Olympics that was a breach of ticket holder's information according to an article on ComputerWeekly.com by Aaron Tan. Our research will show of the effectiveness of using user interaction for verifying individual users for future security implications.

Objective

The objective of our research is to identify the capabilities and possible applications of using user's gestures and interactions with their mobile device as a viable security measure.

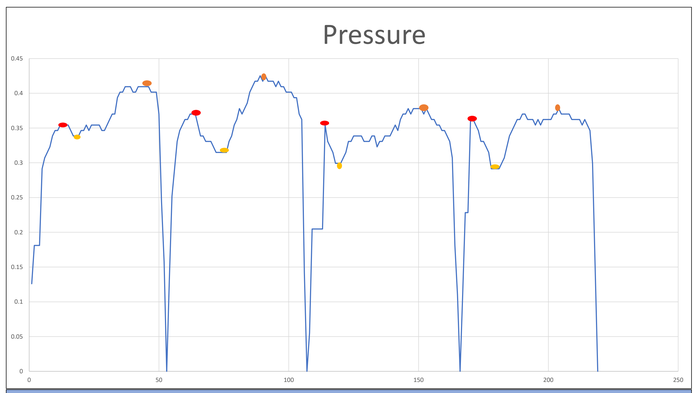
Materials/Methods

For the development of this application, our primary work was completed using Android Studios for our Integrated Development Environment (IDE), Java as our programming language, Weka (Machine Learning Software), and our Google Pixel phone.

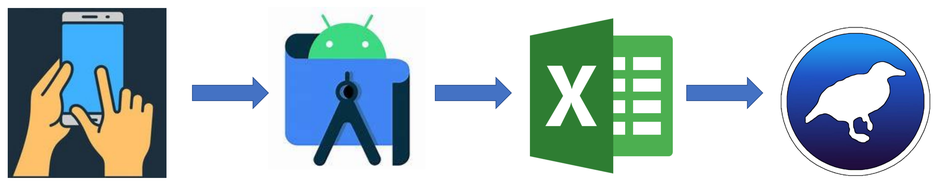
Our process was broken down into developing the code to collect, organize, and export the users' gestures using the Android Studio, and Google Pixel phone to run the program on for testing.

From there we began breaking down the data to find patterns among its sets, these became known as **p_max_1, p_min, p_max_2, and std_dev**. These data points looked to find the initial max, minimum, ending max, and finding the standard deviation of the entire gesture. These points then became what we use In our testing sets and how we identify users.

Finally, was gathering and storing this data and using Weka to verify success rates for comparisons of the data at identifying their users. This is done all through the software to save as time for trial-and-error testing, by doing thousands of comparisons and locating patterns between them.



Legend:
Red: p_max_1 (Initial Max) | Yellow: p_min (Minimum) | Orange: p_max_2 (Final Max)

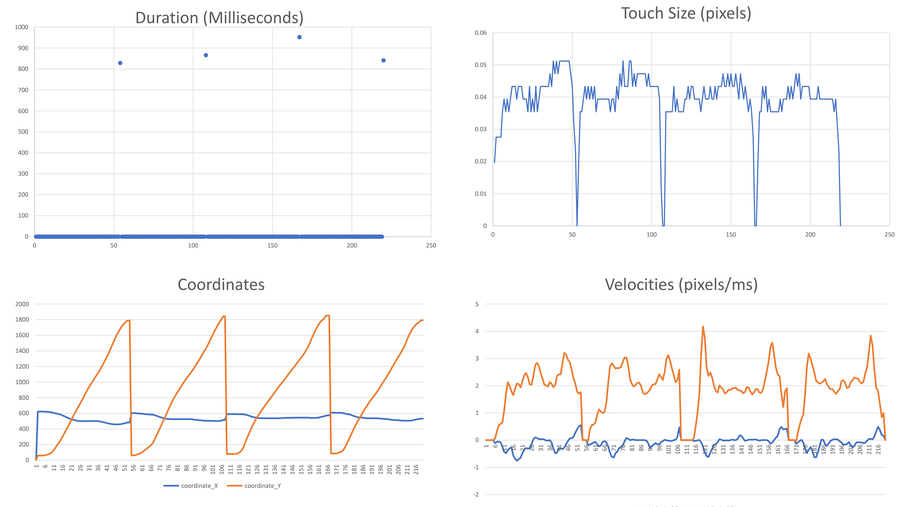


Data Collection **Data Organization** **Data Export** **Data Training and Testing**

Data Samples

TouchID	TouchIndex	Action	startTime (Milliseconds)	endTime (Milliseconds)	durationTime (Milliseconds)	pressure	coordinate_X	coordinate_Y	touchSize (pixels)	Velocity(X)	Velocity(Y)
0	0	null	0	0	0	0	0	0	0	0	0
1	1	Down	19621673	0	0	0.125984	622	58	0.019685	0	0
2	2	Move	19621673	0	0	0.181102	622	58	0.027559	1.76E-04	-2.01E-05
3	3	Move	19621673	0	0	0.181102	622	58	0.027559	-6.01E-05	-3.71E-06
4	4	Move	19621673	0	0	0.181102	622	58	0.027559	-8.12E-05	-1.02E-05
5	5	Move	19621673	0	0	0.291339	620	62	0.027559	-0.11008	0.175721
6	6	Move	19621673	0	0	0.307087	620	68	0.035433	-0.06442	0.384046
7	7	Move	19621673	0	0	0.314961	618	76	0.03937	-0.05201	0.593317
8	8	Move	19621673	0	0	0.322835	616	85	0.035433	-0.08185	0.676305
9	9	Move	19621673	0	0	0.338583	612	102	0.03937	-0.24338	0.888533
10	10	Move	19621673	0	0	0.346457	605	128	0.035433	-0.42949	1.514158
11	11	Move	19621673	0	0	0.346457	600	161	0.03937	-0.48108	2.194427
12	12	Move	19621673	0	0	0.354331	595	195	0.043307	-0.37442	2.504712
13	13	Move	19621673	0	0	0.354331	589	227	0.043307	-0.29515	2.275851
14	14	Move	19621673	0	0	0.354331	584	260	0.043307	-0.27318	2.064762
15	15	Move	19621673	0	0	0.354331	572	301	0.03937	-0.56081	2.210998
16	16	Move	19621673	0	0	0.346457	564	342	0.043307	-0.68521	2.59726
17	17	Move	19621673	0	0	0.338583	552	387	0.043307	-0.75994	2.841776
18	18	Move	19621673	0	0	0.338583	541	428	0.043307	-0.70374	2.757547
19	19	Move	19621673	0	0	0.338583	533	471	0.03937	-0.64662	2.5864
20	20	Move	19621673	0	0	0.346457	525	516	0.03937	-0.47881	2.649902
21	21	Move	19621673	0	0	0.346457	518	561	0.03937	-0.33762	2.72525
22	22	Move	19621673	0	0	0.354331	512	605	0.035433	-0.29288	2.763328
23	23	Move	19621673	0	0	0.346457	507	647	0.043307	-0.31168	2.618675
24	24	Move	19621673	0	0	0.354331	502	685	0.035433	-0.29125	2.339549
25	25	Move	19621673	0	0	0.354331	500	725	0.03937	-0.14775	2.190864
26	26	Move	19621673	0	0	0.354331	500	766	0.035433	0.034564	2.329279
27	27	Move	19621673	0	0	0.354331	500	811	0.043307	0.099429	2.659268
28	28	Move	19621673	0	0	0.346457	500	854	0.035433	0.078444	2.758475
29	29	Move	19621673	0	0	0.346457	500	897	0.03937	0.035842	2.687622

Data Visualization



Results

Currently we have accumulated 10 different data sets from 9 users to test and verify the accuracy of our program.

- Initial tests were to verify the effectiveness of using a user's *pressure*, *touch size*, *and duration time* of their gestures. These features proved to be very effective resulting in our training and testing sets being 100% accurate in depicting the correct users. However, we saw that in our cross-validation results only being 75% effectively; predicted 3 out of 4 users correctly.
- Next, we analyzed the usefulness of the coordinates and velocities collected for location and speed at which the users swiped their phones (initial hypothesis proved these features alone would prove not as effective giving users told to swipe in same location and only vertically). This data, surprisingly, found to be 100% accurate in training and testing data sets. Leading to be 94.7% accurate in our cross-validation sets.
- Lastly, we come to see how **all 5 features**, broken into a total of **25 categories**, are at verifying individuals. ***Training, testing, and percentage split (80/20) tests showed to be 100% accurate, and cross-validation came to 96.875% after 95 folds of the program.*** This without a doubt showed to be the best way at identifying users in our program.

```
Time taken to build model: 0.13 seconds


=== Stratified cross-validation ===
=== Summary ===


Correctly Classified Instances          93               96.875  %
Incorrectly Classified Instances         3                3.125  %
Kappa statistic                          0.9652
Mean absolute error                      0.0071
Root mean squared error                  0.0798
Relative absolute error                  3.9364 %
Root relative squared error             26.3666 %
Total Number of Instances               96

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     IsJake
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     isNathan
               1.000    0.024    0.846      1.000    0.917      0.909    0.998     0.986     isLeftThumb
               0.800    0.000    1.000      0.800    0.889      0.884    0.997     0.973     isAiden
               1.000    0.012    0.909      1.000    0.952      0.948    1.000     1.000     isLuis
               0.889    0.000    1.000      0.889    0.941      0.937    0.999     0.989     isMarisa
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     isMaryam
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     isMiliann
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     isZoe
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     isMitchel
Weighted Avg.  0.969    0.004    0.973      0.969    0.968      0.966    0.999     0.995
```

Conclusion

The ability to use user gesture interaction has great potential for future implications of security measures. Including since our research didn't further into varying the coordinates of such user gestures. Working with the user not only just swiping the phone, but also with drawing other shapes on the screen will only increase the level of variations that can occur. More possibilities will lead to greater security of the user's data, which is the ultimate goal in this project's potential.