Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 5th, 2017

Increasing Accuracy Rate of Behavioural Biometrics for User Authentication on Android-Based Smartphones

Javid Maghsoudi and Charles C. Tappert
Seidenberg School of CSIS, Pace University, Pleasantville, New York
javidm1@gmail.com, ctappert@pace.edu

Abstract

This study examines behavioral biometrics, specifically smartphone motion, to determine potential areas of increasing authentication accuracy on these devices, specifically Android-based phones. The study used the application Sensor Kinetics Pro and the Weka machine learning library, to analyze accelerometer and gyroscope data. It analyzed the impacts on accuracy rate for authentication when the motions data from two sensors were used instead of one (e.g., using the data from both Accelerometer and Gyroscope), when one specific statistical model was used (e.g., Average versus Variance), when specific Machine Learning (ML) algorithm (e.g., Support Vector Machine vs. Multilayer Perceptron), or different random sample sizes (e.g., five versus ten, versus twenty) were used. With these focus, this study supports a) the use of behavioral motion biometrics for authentication, and b) gain higher authentication accuracies.

Keywords - Behavioral biometrics, User authentication, Accelerometer, Gyroscope, Android devices, Weka, Motion Capture

1. Introduction

The purpose of this study is to answer two questions on whether 1) the behavioral biometrics can satisfy the requirements of authentication, as compared to the better known physiological biometrics (iris and fingerprint), and 2) if certain consideration were applied, will the authentication accuracies can increase? The study's objectives include generating training data of different individual's behavioral biometrics, training machine learning algorithms from the Weka (Waikato Environment for Knowledge Analysis) library to recognize an individual. In the end, this study supports the notion that behavioral biometrics could be up to the task of adequately identifying the proper user of a device, and also there are ways to increase the accuracy rate for authentications.

Using behavioral biometrics for user authentication is comparatively less established than the use of physiological biometrics in the arena of authentication (previously solely the province of passwords and later key tokens) [3]. Behavioral biometrics exist in several broad categories, including touch gestures, motion, orientation (that one normally applies to mobile devices), as well as other characteristics such as mouse dynamics, handwriting, grip, or gait/stride [16].

Behavioral biometrics are appealing for several reasons: It may be more difficult for someone with malicious intent to capture a natural motion as compared to passwords or tokens. Further, natural motions are committed continually in holding the device, walking around with it, and holding it up to the user's ear. In this fashion, they could form a continuous form of authentication [11].

There were a total three researches done on this exact area. They were in Spring 2015, Fall 2015, and Spring 2016. Each semester had different test subjects, different phones and different ML algorithms applied.

2. Literature Review

With the explosive growth of mobile devices, the desire to secure private, sensitive, or business information on those devices has motivated the search for alternative means of authentication. As stated in the introduction, physiological biometrics have been studied and utilized, however they can be captured as is the case with older methods, such as PINs and passcodes. Behavioral motion biometrics is an area that could be further exploited [16].

Several previous studies have investigated the use of non-physiological biometrics for authentication in recent years [2, 6, 7, 11]. The reasons for the increase in interest in the field are obvious and numerous: smartphone proliferation [14] and vast expansion of smartphone theft, cyber security issues, an increased desire for password/PIN alternatives [6], physiological biometric use expansion, etc.

While this study will focus on the way users might authenticate themselves to the device manually, chiefly by raising the phone to eye level to check the caller, then raising the phone up to their ear, it is worth noting that there are alternative means of using behavioral biometrics. One study focused on the user's gait to authenticate the user to a device in their pocket, which could be extended to continuous authentication [11].

Using motion biometrics for authentication, this work extends that of two earlier studies [2, 7]. Both studies focused on a user's actions while picking up the phone, holding it to his/her ear, and putting it down. To accomplish this, both studies used the accelerometer built into typical Android-based devices, and the second study incorporated the built-in gyroscope as well, to further improve accuracy.

The accelerometer measures two kinds of acceleration: static and dynamic. Static acceleration refers to the exact angle of the device relative to earth and it helps measure device orientation. Dynamic acceleration refers to a device's movement relative to earth and it helps measure device movement. While the accelerometer oversees monitoring when the device is rotated, and adjusting screen accordingly, the gyroscope maintains and measures the orientation and angular rotation. [5].

Both studies used the Android app, Sensor Kinetics Pro, to capture the motion data and Weka, an open source software library, to analyze the motions.

A front-end application was also designed to capture the Android-based device sensor data. An experiment was conducted using this application to capture finger touch swipe gesture patterns. These patterns were then analyzed using WEKA to determine if they could be used for authentication. The paper written about the experiment conducted with the application mentioned several improvements that were taken into consideration [1].

For the purposes of continuity, the ways in which this study varies from the works of Carlson et al., and Kunnathu, are summarized here:

(a) The sample size is greatly increased: five phones are used, each capturing data for 10 different individuals, performing 20 trials of two motions: bringing the phone up to eye level for review, then bringing the phone to the ear. Thus, the total number of runs is expanded to over 2000, compared to the 500 performed earlier.

(b) The accelerometer and gyroscope remain the sensors of interest. Each motion, rather than average the whole motion of the x, y, and z axes, each motion is broken into quadrants, and then averaged. Each quadrant then has its own variance--this quadruples the resolution of the motion data that is processed.

(c) This study compare the user of Average versus Variance.

(d) This study added Support Vector Machine (SVM) algorithm to the other algorithms tested such as Multilayer Perceptron (MLP), Naive Bayes, and k-Nearest Neighbor algorithms.

(d) This study uses different sample sizes to see how size could impact the accuracies and how individual MLs could behave.

3. Tools

Sensor Kinetics Pro is an Android app created by INNOVENTIONS, Inc. This app detects the sensors that the phone possesses and allows them to be tested. The sensors are broken down into sections. The three-dimensional sensors include the accelerometer, gyroscope and magnetometer. The derived 3-D sensors are the gravity sensor, linear acceleration sensor and a rotation sensor. Sensor Kinetics Pro allows the user to monitor each sensor individually as a chart and save and share the data (Fig. 1). Sensor Kinetics Pro also allows the user to run multiple sensors at once to enable the observation of data captured from one motion from multiple sensors captured [13].

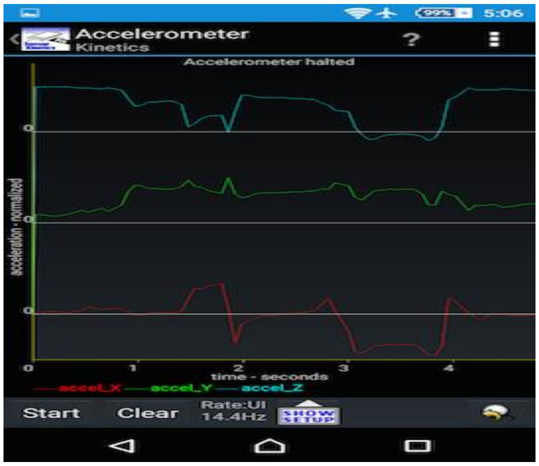


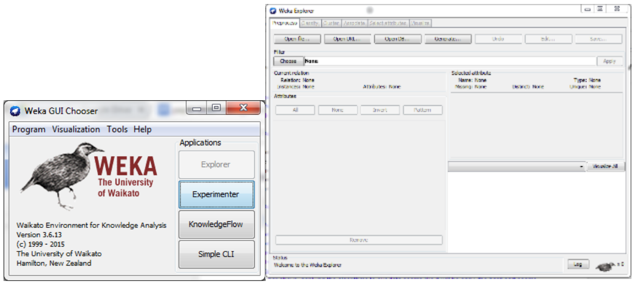
Figure 1: Sensor Kinetics Pro App

This research will use the accelerometer and gyroscope sensors for this experiment because they are some of the most common sensors found in smartphones and together, they cover the desired range of motion. Additionally, because they were incorporated in previous research, they offer a measure of continuity. After the data, has been collected, the app permits saving the data. It saves the motion data points as a .CSV file, visualized as a graph.

Included on the .CSV are numerical values for time and the location of the phone on the x, y and z axes. This .CSV file can then be used with Weka to process the data [13].

Weka is a library with a collection of machine learning algorithms. It is an open source software package developed at the University of Waikato, New Zealand, and it is available under the GNU General Public License. The most recent version of Weka is fully Java-based which makes it easily accessible to most users. Weka is mainly used for data preprocessing, clustering, classification, regression, visualization, and feature selection. [15]

This study uses Weka's numerous machine learning algorithms to build models for identifying users for authentication. For the purposes of this study, Weka's algorithms for k-nearest neighbors, multilayer perceptron, Support Vector Machine and Naive Bayes are used.



Waikato Environment for Knowledge Analysis
Version 3.6.13
(c) 1999 - 2015
The University of Waikato
Hamilton, New Zealand

Welcome to the Weka Explorer

Figure 2: Weka

4. Data Gathering Protocol

Unlike physiological biometrics capture, such as fingerprinting and iris scanning, behavioral biometrics requires a larger set of training data for the phone to learn the users' unique use patterns.

There were six test coordinators and there was a total of five different Android-based phones. Each tester found nine other individuals on whom to perform the tests. There was a total of sixty test participants. Each test participant/subject performed the motion 20 times. About 90% of those served as the training data, while the remaining 10% served as the testing data against which the algorithm had to perform (using a 10-fold cross validation test). Naturally, samples from the different subjects served as tests also, and should be identified properly.

Test subjects were handed one of five possible Android-based phones:

1. LG Nexus 5
2. Motorola Moto G (3rd Gen)
3. HTC One m8
4. Samsung Galaxy Note 4 (2 testers)
5. Samsung Galaxy Note 5

This paper also refers to the prior years' research done with other phones when comparing results for one sensor vs. two sensors.

(Note that the Motorola Moto G did not have a gyroscope, thus it only captured accelerometer data.)

Each test participants performed the following steps:

1. The phone was laid flat on a table. The Kinetic Pro app was started on that phone by the test participants. The test participants then lifted the phone, as they normally would, as if lifting the phone from a pocket or purse. They ceased lifting as if they were stopping to look at the device screen, to view the identity of the caller.

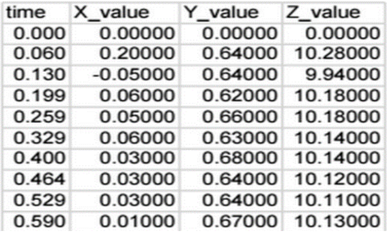
2. Looked at the screen for a moment, and when they were satisfied that they had looked long enough to judge that the call was worth taking. This helped to signify that one motion had ended and another began.

3. This helped to signify that one motion had ended. Then they raised the phone to the ear level.

   The kept the phone at ear level for few seconds.

4. Then they pressed the stop on the app to stop the app.

The tester then stopped the motion capture of the device, and prepped for another trial run as needed. Sample images of these trials are represented in Figures 3, 4 and 5.



time	X_value	Y_value	Z_value
0.000	0.00000	0.00000	0.00000
0.060	0.20000	0.64000	10.28000
0.130	-0.05000	0.64000	9.94000
0.199	0.06000	0.62000	10.18000
0.259	0.05000	0.66000	10.18000
0.329	0.06000	0.63000	10.14000
0.400	0.03000	0.68000	10.14000
0.464	0.03000	0.64000	10.12000
0.529	0.03000	0.64000	10.11000
0.590	0.01000	0.67000	10.13000

Figure 3: Accelerometer data

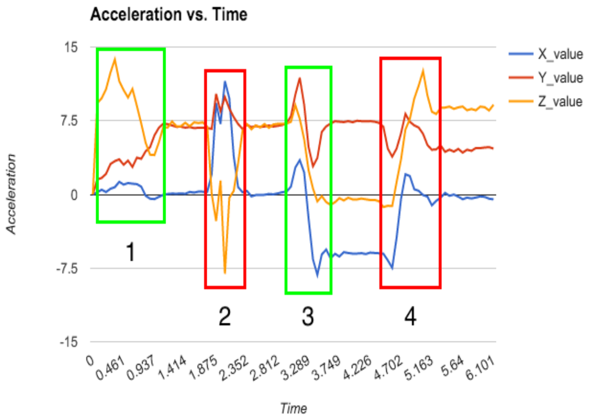


Figure 4: Simple graph of the motions

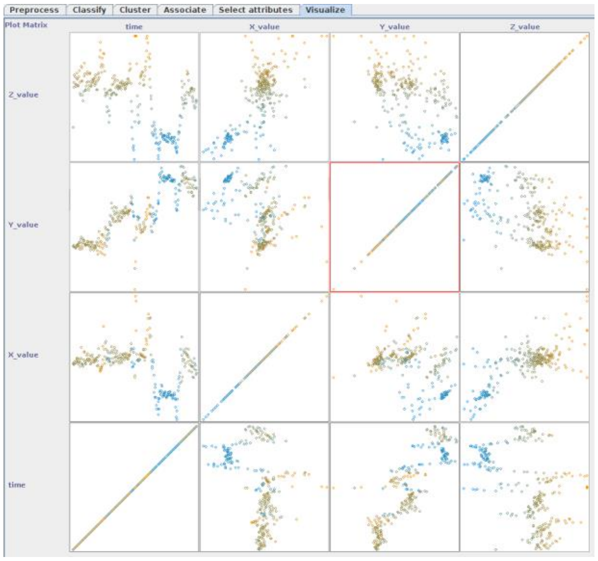


Figure 5: Weka's view of a single entry.

5. Data Processing & Machine Learning

The data was processed in the following manner: (1) Using a graph of the accelerometer data, the motions were visually inspected to isolate the particular times a given motion began and ended; (2) With the relevant "timestamps" marking the beginning and ending of a given motion, the motions were broken into quartiles; (3) Depending on whether or not the accelerometer and gyroscope captured data at the same sampling rate (the Nexus 5 did not), the time indices for the gyroscope were adjusted to match the timestamps identified for the accelerometer. More accurately, the row numbers were used, with the assumption that the sampling occurred at relatively constant rates; (4) The quartiles per motion were then averaged ({x,y,z} separately) and had a variance taken ({x,y,z} separately), so that each motion had 24 attributes for the accelerometer, and 24 attributes for the gyroscope, if present. Each trial consisted of two major motions, and thus consisted of 96 total data points (again, 48 for the Moto G 3rd Gen, lacking a gyroscope).

The use of quartiles was intended to recapture individual idiosyncrasies, while making the data amenable to the Weka library's numerous classification algorithms. Dividing up the data into smaller partitions would possibly create samples with too few data points, resulting in meaningless variances.

Initially the individual motions were manually extracted as a proof of concept to show the accuracy of the classifier algorithms with a standardized motion capture. The process was then automated using a Java program that was created process the .CSV file. This program accepts the .CSV file with user motion data, and then breaks stream of data for all four user motions into eight segments. Each of the eight segments rather than the quartiles that were used for the manual data.

For the manual data, the information was aggregated into a single .CSV file that was plugged into Weka, and each 96-data-point row was post-pended with an identifier: the test subject initials.

Once the data was appropriately captured and pre-processed, the following machine learning algorithms were applied using Weka: First, the Multilayer Perceptron was used. Second, the k-Nearest Neighbors (k-NN) algorithm was employed, for being both a benchmark, and a fast, simple algorithm to run. Finally, the Naive Bayes was selected by the research team for being well known and widely used. Finally, Support Vector Machine (SVM) was used as SVM is one of the highly used ML algorithm in the industry.

A Multilayer Perceptron is a feedforward artificial neural network. Patterns are presented via the input layer which communicates with hidden layers where the actual processing is done via the system of weighted connections. The hidden layers then link to an output layer (Fig. 7).

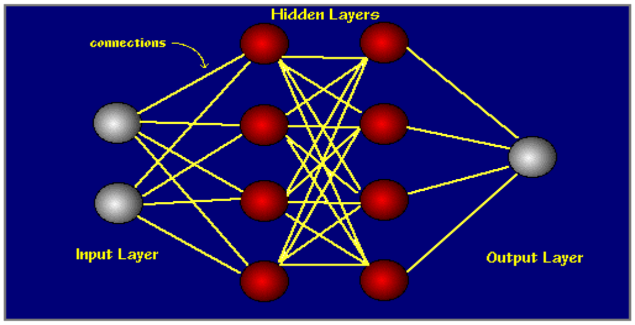


Figure 7: Diagram of a Multilayer Perceptron [13]

The network is trained using backpropagation -- a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern). When a neural network is initially presented with a pattern it makes a random 'guess' as to its classification. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights.

The k-Nearest-Neighbor algorithm assigns an unknown object to the class most common among the unknown's k nearest neighbors in feature space. A shortcoming of the k-NN algorithm is that it is sensitive to the local structure of the data.

The naive Bayes classifier is a simplification of Bayes decision method that assumes the features are independent of each other.

The Support Vector Machine (SVM) is another classification method where one plots each data item as a point in n-dimensional space (where n is number of features one has) with the value of each feature being the value of a coordinate. [101].

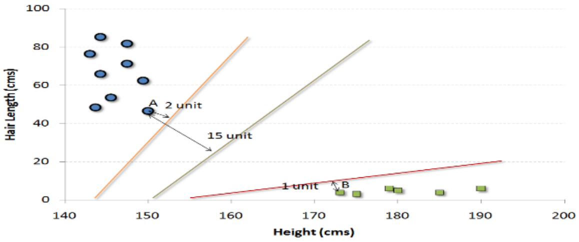
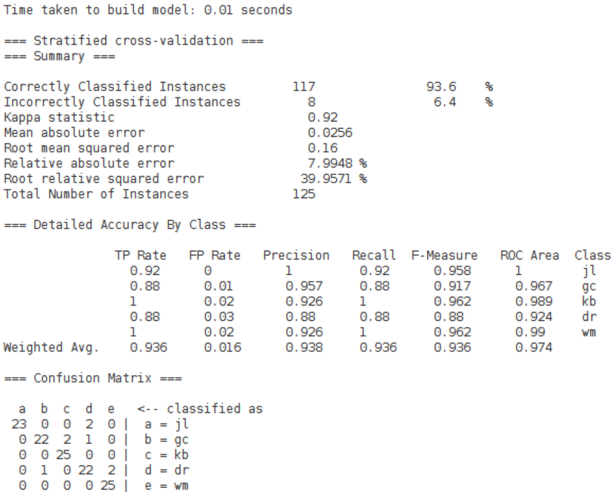


Figure 8: SVM Algorithm [101]

6. Test Results



```
Time taken to build model: 0.01 seconds

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

Correctly Classified Instances         117               93.6    %
Incorrectly Classified Instances         8                6.4    %
Kappa statistic                          0.92
Mean absolute error                      0.0256
Root mean squared error                  0.16
Relative absolute error                  7.9948 %
Root relative squared error             39.9571 %
Total Number of Instances              125

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.92      0          1         0.92      0.958      1        jl
                 0.88      0.01       0.957     0.88      0.917      0.967    gc
                 1         0.02       0.926     1         0.962      0.989    kb
                 0.88      0.03       0.88      0.88      0.88       0.924    dr
                 1         0.02       0.926     1         0.962      0.99     wm
Weighted Avg.    0.936     0.016      0.938     0.936     0.936      0.974

=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
 23  0  0  2  0 |  a = jl
  0 22  2  1  0 |  b = gc
  0  0 25  0  0 |  c = kb
  0  1  0 22  2 |  d = dr
  0  0  0  0 25 |  e = wm
```

Figure 9: Sample Weka Output

Cross validation (10 folds) was used to evaluate the learning algorithms. This means that each data set was randomly broken into 10 equal parts, and then the algorithm learned on nine of those parts, testing the algorithms' accuracies on the remaining data. This was repeated for each fold.

Test Results: **Measuring Accuracy Rate**

- **Comparing Two Sensors Instead of One:**

Table 1 shows the results of using one sensor vs. two sensors in the Fall 2015 research. That research used four (4) phones, each tested using five (5) individual users, each user performing 25 trials, using the algorithms: naive Bayes, k-nearest-neighbor (k-NN), and MultiLayer Perceptron (MLP).

TABLE 1:
Accuracy results with both accelerometer and gyroscope data from the Fall 2015 research.

	Nexus 5	HTC One M8	Moto G (3)	Xperia Z1	All Phones
Accelerometer					
Combined motions (1&2)	N-B: 96.0% **kNN:** **98.4%** **MLP:** **98.4%**	N-B: 98.4% **kNN:** **100%** **MLP:** **100%**	N-B: 86.4% kNN: 90.4% **MLP:** **94.4%**	N-B: 91.2% kNN: 89.6% **MLP:** **93.6%**	N-B: 91.4% kNN: 91.4% **MLP:** **94.2%**
Gyroscope					
Combined motions (1&2)	N-B: 89.6% kNN: 92.0% **MLP:** **96.0%**	N-B: 94.4% kNN: 96.0% **MLP:** **98.4%**	N-B: * kNN: * MLP: *	N-B: 82.4% kNN: 91.2% **MLP:** **92.8%**	N-B: * kNN: * MLP: *
Accelerometer & Gyroscope combined					
Combined motions (1&2)	N-B: 93.6% kNN: 97.6% **MLP:** **98.4%**	N-B: 98.4% **kNN:** **100%** **MLP:** **100%**	N-B: 86.4% kNN: 90.4% **MLP:** **94.4%**	N-B: 88.8% kNN: 95.2% **MLP:** **98.4%**	N-B: 91.0% kNN: 93.4% **MLP:** **96.2%**

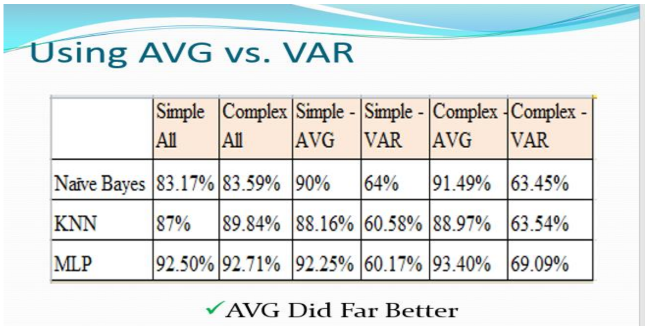
As the results show using two sensors provided higher accuracy rates instead of using either individual sensor. was used.

- **Using Average vs. Variance:**

Below is a summary of improvements when different ML algorithms were used to test Average vs. Variance statistical methods. The research compared the results between these two-separate statistical approached, AVG (Average)vs. VAR(Variance). The results showed that Average did better than Variance when three separate ML algorithms were used.

TABLE 2:

Accuracy results when either Average or Variance was used in the Spring 2016 research.



Using AVG vs. VAR

	Simple All	Complex All	Simple - AVG	Simple - VAR	Complex - AVG	Complex - VAR
Naïve Bayes	83.17%	83.59%	90%	64%	91.49%	63.45%
KNN	87%	89.84%	88.16%	60.58%	88.97%	63.54%
MLP	92.50%	92.71%	92.25%	60.17%	93.40%	69.09%

✓AVG Did Far Better

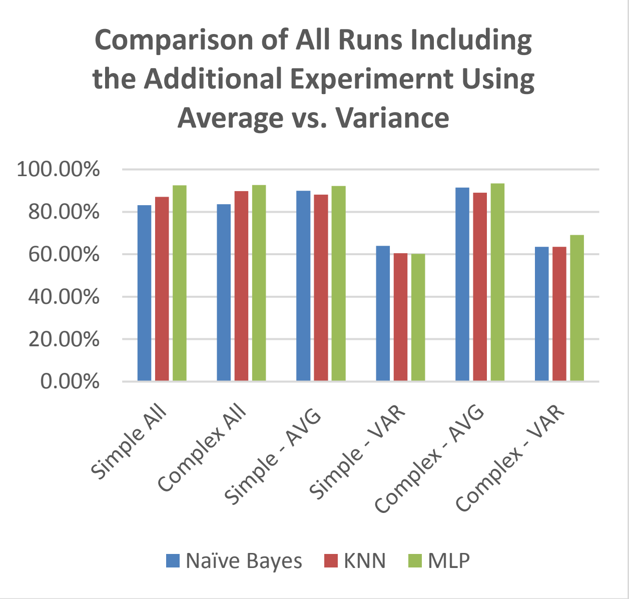


Figure 9: All Runs vs. Average or Variance

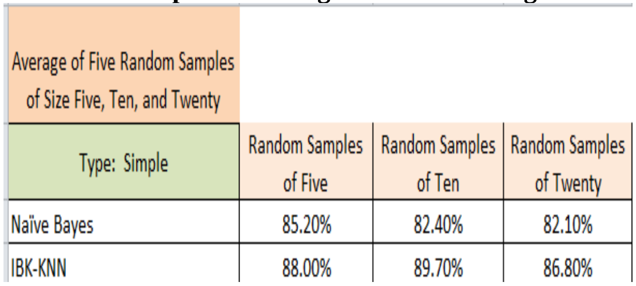
The results showed that all three algorithms had higher accuracy rates when Average statistical method was used. MLP was higher in all instances.

- **Using Different Sample Sizes & Choices of Algorithms:**

The Spring 2016 study then tried different sample sizes to measure the accuracies of the algorithms as the population sizes changed The study covered random samples sizes of five, ten and twenty. In all, as the population size grew, the accuracies changed (went down) or stayed almost the same when the same algorithms were used. However, when the algorithm was changed the results were very different. So, the choice of algorithm mattered. Here, k-Nearest Neighbor(k-NN) fared far better than Naive-Bayes in all three sample sizes.

TABLE 3:

Different Sample Size Using Different ML Algorithms



Average of Five Random Samples of Size Five, Ten, and Twenty			
Type: Simple	Random Samples of Five	Random Samples of Ten	Random Samples of Twenty
Naïve Bayes	85.20%	82.40%	82.10%
IBK-KNN	88.00%	89.70%	86.80%

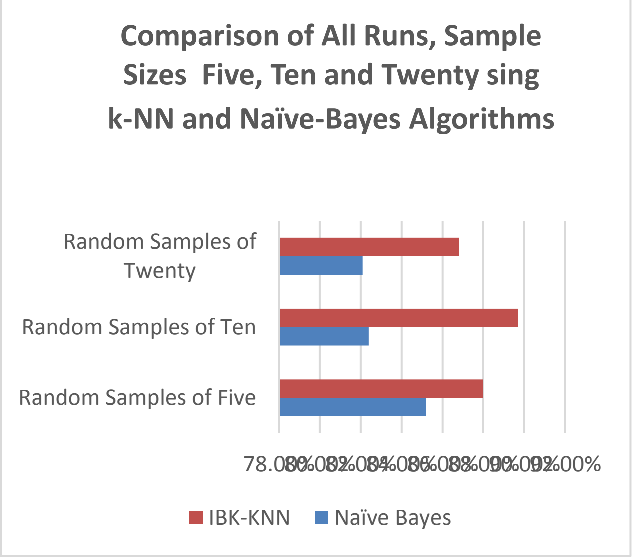


Figure 10: Using Different Sample Sizes and Different ML Algorithm

TABLE 6:

Comparison of Results from Four Algorithms

%	Simple (both)	Complex (both)
N-B	83.2	83.6
k-NN	87	89.8
MLP	92.5	92.7
SVM	92.8	92.2

Table 6.1 Approximate Values of Improvements of Research in Spring 2016

Improvement	Percent
From 1 sensor to 2 sensors (for MLP)	1.1%
From simple to complex (for K-NN)	2.1%
From NB to k-NN (For Complex)	6.2%
From k-NN to MLP (for Simple)	5.5%
From N-B Algorithm to MLP algorithm (for Complex)	9.1%

7. Conclusion

The results support the that biometric authentication could reliably be used to authenticate users. The results showed that having two sensors vs. one provided better accuracy rates. The study also supported the fact that selecting specific statistical model like Average vs. variance could significantly improve the accuracy rate. The study also showed that selecting the right algorithm matters. The study also showed that different ML algorithms behave differently as the sample sizes grow. The best performing algorithm, of the ones tested, is the SVM algorithm though. MLP carried the heaviest data-processing footprint of the four to build its model.

There are several observations for improvements. First, the drastic expansion in number of samples per person, likely played a role in better training the algorithm to recognize different individuals.

Second, breaking the motions into quartiles created much more idiosyncratic data points for everyone, rather than the overall averaging of the x, y and z components across the entire motion.,

Third, cutting out the additional processing steps such as filtering out the effects of gravity via approximation, which likely only served to introduce noise, or attempting to determine the most salient attributes and filtering out the less salient ones, likely improved the results by providing the maximum number of data points.

8. Future Development

Some areas of improvement and further study are immediately apparent following the conclusion of the analysis.

First, each user in this study supplied his or her twenty sets of data in the same session. Further testing is necessary to determine whether the classification would prove as successful under a variety of conditions. A user's gesture may vary depending on the time of day, attempting the gesture while engaged in physical activity, the user's state of mind, or other situational impairments such as holding something in the dominant hand which would otherwise be typically used to perform the gesture. It would also be useful to further investigate whether multiple gestures improve the classification accuracy.

Additionally, testing should be done to see if users can defeat the models by observing how others use the phone, either in person or by watching a recording. An attempt to replicate the motion by a machine that can
observe a visual recording of the users authenticating themselves to their device, would be a step further and very interesting to consider.

Going forward, it is also important to determine the threat model that authentication seeks to prevent. While the 'picking up the phone' gesture tested in this study has a high success rate, it is more suitable for a one-time login rather than continuous authentication. If continuous authentication is indeed required, then a gesture more suitable than the one tested in this paper would be necessary.

Future development might work on seamlessly training the device to learn the user behavioral biometrics. For example, while the user makes ordinary use of the device, such as swiping a regular pattern to unlock the screen, the device could begin training itself on the user's unique characteristics. This would involve extracting the gesture data from the applications that are already in normal use, or injecting some code that records that data. In this fashion, the training of the device could be invisible to the user.

A legal curiosity worth mentioning: how would non-physiological biometrics as unlock function, interface with law enforcement limits on legal search and seizure, or court orders to compel a user to unlock a device?

It is necessary to work through the other legal ramifications related to behavioral biometric authentication as well. The topic engenders several privacy and security concerns. For example, per one study most behavioral biometrics could reveal neurological diseases and other personal information [10].

References

[1] N. Alotaibi, et al., "Biometric System Design for Handheld Devices." Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 2014.

[2] C. Carlson, et al., "User Authentication with Android Accelerometer and Gyroscope Sensors," Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 2015.

[3] H. Crawford. "Keystroke dynamics: Characteristics and opportunities." In Privacy Security and Trust (PST), 2010 Eighth Annual International Conference 2010 (pp. 205-212): IEEE

[4] C. W. Dawson, Projects in Computing and Information Systems (second edition), Addison-Wesley, 2000, 2009

[5] R. Goodrich, "Accelerometer vs. Gyroscope: What's the Difference?" LiveScience, TechMedia Network, 01 Oct. 2013, Web, accessed 17 Apr. 2015. http://www.livescience.com/40103-accelerometer-vs-gyroscope.html

[6] D. Guse & B. Muller, "Gesture Based User Authentication for Mobile Devices using Accelerometer and Gyroscope," http://www.researchgate.net/profile/Dennis_Guse/publication/235988370_Gesture-based_User_Authentication_for_Mobile_Devices_using_Accelerometer_and_Gyroscope/links/00463515416f10e759000000.pdf, Accessed October, 2015.

[7] N. Kunnathu, "Biometric User Authentication on Smartphone Accelerometer Sensor Data," Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 2015.

[8] J. V. Monaco, et al., "Developing a keystroke biometric system for continual authentication of computer users." Intelligence and Security Informatics Conference (EISIC), 2012 European, IEEE, 2012.

[9] J. V. Monaco, et al., "Recent Advances in the Development of a Long-Text-Input Keystroke Biometric Authentication System for Arbitrary Text Input." Intelligence and Security Informatics Conference (EISIC), 2013 European, IEEE, 2013.

[10] E. Mordini & H. Ashton, "The Transparent Body - Medical Information, Physical Privacy and Respect for Body Integrity," in Mordini E, Tzovaras D (eds), Second Generation Biometrics: the Ethical and Social Context. Springer-Verlag: Berlin, 2012.

[11] M. F. Nowlan, "Human Identification via Gait Recognition Using Accelerometer Gyro Forces," http://cs-www.cs.yale.edu/homes/mfn3/pub/mfn_gait_id.pdf, Accessed October, 2015.

[12] Pace Biometrics Classifier, http://www.csis.pace.edu/~ctappert/it691-projects/biometrics-background.htm, Accessed October, 2015.

[13] Pages.cs.wisc.edu, 'A Basic Introduction To Neural Networks', 2015. [Online]. Available: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [Accessed: Oct- 2015].

[14] Sensor Kinetics Pro. 17 Apr. 2015. https://play.google.com/store/apps/details?id=com.innoventions.sensorkineticspro&hl=en; http://www.rotoview.com/sensor_kinetics_pro.htm, Web. Accessed October, 2015

[15] S. Trewin, et al., "Biometric Authentication on a Mobile Device." Proceedings of the 28th Annual Computer Security Applications Conference on - ACSAC '12 (2012), pp 157-160 Web. http://researcher.ibm.com/researcher/files/us-kapil/ACSAC12.pdf, Accessed October, 2015

[16] Weka http://www.cs.waikato.ac.nz/ml/weka/ Accessed October, 2015

[17] R. Yampolskiy, and V. Govindaraju, "Behavioral biometrics: a survey and classification," International Journal of Biometrics, Inderscience Publishers, January 2008, pp. 81-113, January 2008.

[18] Specifying an Accelerometer: Function and Applications. (2015, July 28). Retrieved from http://insights.globalspec.com/article/1263/specifying-an-accelerometer-function-and-applications