NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IDENTIFICATION OF A SMARTPHONE USER VIA KEYSTROKE ANALYSIS

by

Samuel B. Fleming

March 2014

Thesis Co-Advisors: Craig Martell
Mark Gondree

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2014	Master's Thesis

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
IDENTIFICATION OF A SMARTPHONE USER VIA KEYSTROKE ANALYSIS	
6. AUTHOR(S) Samuel B. Fleming	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Reconnaissance Office 14675 Lee Road Chantilly, VA 20151-1715	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number NPS.2013.0058-IR-EP7-A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	A

13. ABSTRACT (maximum 200 words)

Keystroke analysis has been an accepted method for user identification and authentication since the early 1980s. Most of the research in this field of biometrics has focused on traditional computer keyboards, with very few experiments performed on touchscreen keyboards found on modern smartphones. This study focused on identifying a smartphone user based on typing samples input by copying pre-written text, as well as spontaneously-authored free text. Features used for identification were duration of key press, as well as bigram and trigram transitions. User classification based on duration features proved to be successful in 70 percent of inputs to our *k*-nearest neighbors classifier.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Android, behavioral biometrics, free-text, fixed-text, *k*-nearest neighbors, keystroke analysis, smartphone, touchscreen	47
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

IDENTIFICATION OF A SMARTPHONE USER VIA KEYSTROKE ANALYSIS

Samuel B. Fleming
Lieutenant, United States Navy
B.S., North Carolina State University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2014

Author: Samuel B. Fleming

Approved by: Craig Martell
Thesis Co-Advisor

Mark Gondree
Thesis Co-Advisor

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Keystroke analysis has been an accepted method for user identification and authentication since the early 1980s. Most of the research in this field of biometrics has focused on traditional computer keyboards, with very few experiments performed on touchscreen keyboards found on modern smartphones. This study focused on identifying a smartphone user based on typing samples input by copying fixed text, as well as spontaneously-authored free text. Features used for identification were duration of key press, as well as bigram and trigram transitions. User classification based on duration features proved to be successful in 70 percent of inputs to our *k*-nearest neighbors classifier.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I owe a huge thank-you to all of the following:

My wife, Caroline, for putting up with my ridiculous work hours and non-stop blathering about (insert computer science techno-babble here) while I completed my degree. Your support means everything to me.

My brother, for being a much better programmer than I will ever be and showing me an easier way to solve Wumpus World-related problems.

Will and Vince, for support not only on thesis-related issues, but also for help during the steep learning curve of our degree program.

Dr. Mark Gondree, for your sense of humor in our computer security class and for your extreme patience as I figured out this whole research process.

And finally, Dr. Craig Martell, for your constant technical guidance, for making the thesis process a much less daunting undertaking, and for sparking my interest in AI and machine learning by being such an enthusiastic teacher.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Passive authentication of mobile devices using biometric signals has been proposed as a more secure and more convenient solution to the problem of end-user authentication. Proposed methods include gait analysis [1] and geolocation via RSSI signals [2]. We investigate the use of keystroke timing dynamics, a biometric signal previously studied for user authentication in the context of desktop clients [3] [4]. We extend this work, investigating if those results extend to the domain of software-based keyboards on mobile devices. We re-investigate the results of Tappert *et al.* [5] [6] [7] on keyboard authentication using pre-selected and free text samples for hardware keyboards with desktop devices in the new domain of software keyboards on mobile devices.

Analysis of keystroke dynamics for the purpose of identifying someone falls into a category of biometrics known as behavioral biometrics. Where physical biometrics are concerned with features of the human body that cannot be easily changed, such as fingerprints or retinal blood vessel patterns, behavioral biometrics encompass human traits that require motor skills, such as typing or walking. Yampolskiy and Govindaraju [8] observed that behavioral biometrics differ from physical biometrics in that they often incorporate a time measurement, such as how long it takes a person to transition from a press of a particular key to a press of another key.

The rapid adoption of touchscreen mobile phones has opened up a new opportunity for study in keystroke dynamics and, to date, there are very few experiments in using keystrokes entered on a virtual keyboard via a touchscreen to identify and authenticate a user. A touchscreen or soft keyboard on a smartphone offers significantly more challenges for keystroke analysis than a hardware keyboard. Since hardware keyboards have a more-or-less established shape and layout and have been used by people for most of their lives, most people are much more familiar and skilled with them than with touchscreen keyboards. Soft keyboards have only recently become more widely used as iOS and Android based phones have driven wide-spread smartphone adoption. The small form factor of a smartphone, the dramatic variation in keyboard size and layout, and some peoples' discomfort with soft keyboards due to the lack of physical

feedback they would receive from a hardware keyboard all combine to make text input on a soft keyboard a much different experience that may lead to dramatic differences from how they would type on a hardware keyboard or differences in typing style among different users.

A. RESEARCH QUESTION

Given an observation of a user's typing behavior on a smartphone with a touchscreen keyboard, can we identify the user based solely on the timing patterns associated with the previous observations?

The goal for the authentication is to answer the yes or no question: "Given a set of prior observations from X and an observation from a user, can we decide that the user is X?" Toward this goal, our research investigates a slightly different question: "Given a set of observations from a population and an observation from a user in that population, can we decide the identity of the user?" These two questions are different and require different approaches with regard to how we model the data presented to the classification algorithm. Commonly the former requires, for each user, a model of each user's timing and a "model-of-everyone-else" The "model-of-everyone-else" is commonly implemented by analyzing all of the other users' timing data in aggregate [5]. In comparison, the latter only requires a model of each user's keystroke timing. These models are then compared to the sample from the unknown user and the username of the model that looks the most like the sample is chosen to label the unknown user. We are investigating the latter in this study as a first step toward determining the feasibility of using keystroke timing data from touchscreen keyboards as an identifying feature in the authentication process.

B. RESULTS

When splitting the typing samples into 80 percent training and 20 percent testing sections, we were able to successfully identify the author of a given sample of typing 70 percent of the time using a *k*-nearest neighbors classifier. This fell to 40 percent when attempting to classify the user when based on the fourth sample, a free-text sample typed with the phone held in landscape orientation.

When splitting fixed-text typed with the phone in portrait orientation into four training and test sets by paragraph, we successfully identified the user 80 percent of the time using a *k*-nearest neighbors classifier.

C. ORGANIZATION OF THESIS

Chapter I introduces the research question, motivation for our study, and gives a summary of results. Chapter II discusses prior and related work and gives background on the algorithms and features used in this study. Chapter III discusses the structure and methodology used in this study. Chapter IV describes our data analysis procedures. Chapter V presents a discussion of our results. Chapter VI briefly summarizes our work and contains suggestions for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

In this chapter, we explore common features used in keystroke analysis, as well as prior and related work.

A. FEATURES

There are two primary categories of features explored in authentication studies based on keystroke dynamics: single character duration and n-gram timing data. Single character duration is measured for each key pressed and is the time from the press of the key to the release of the key. N-gram timing data consists of the timing between transitions between characters, with timings among bi-grams and tri-grams being most common. Following the naming convention of Tappert [5], there are two categories of transition timing one may measure: type-1 and type-2 (see Figure 3). A type-1 transition (or type-1 timing data) is the time elapsed from the release of a key to the press of the next key and can be negative. A type-2 transition (or type-2 timing data) is the time elapsed from the press of a key to the press of the next key and is always positive.

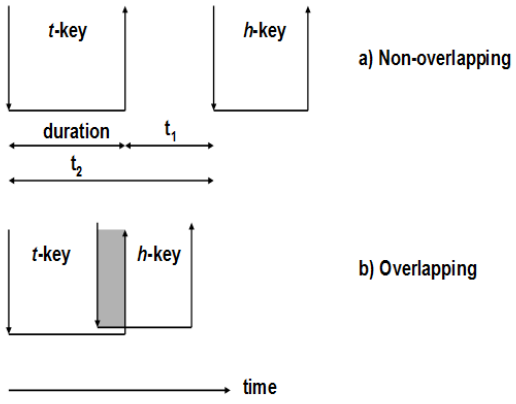

Figure 1. Type-1 (t1) and type-2 (t2) timing data and duration (from [5]).

B. PRIOR WORK

One of the first studies into keystroke dynamics as a method of user identification was done by Gaines *et al.* [9] in 1980. They asked six experienced secretaries at The

Rand Corporation to type three pre-prepared samples into a computer that measured the time it took them to transition between each successive pair of letters, also known as digraphs. They note that even before performing formal statistical analysis on the samples, it was obvious by simply comparing the timing charts on paper that the results for each secretary were unique. A summary of this and all referenced prior work can be found at the end of this section in Table 1.

Monrose and Rubin [10] later demonstrated their own system for identifying an individual, grouping test subjects into hierarchies based on similarity in typing style as they typed both pre-prepared text and free text. They then performed classification based on four progressively more complex methods that gave corresponding increasingly better results:

- Euclidian distance between timing vectors taken from training and test samples.
- Non-weighted probability that a given timing vector was from a particular subject.
- Weighted probability that a given timing vector was from a particular subject.
- Implementation of a Bayesian classifier.

Bergadano *et al*. [3] demonstrated a statistical method for authenticating users of a system based on the time between pressing the first and last key of successive series of trigraphs, or sets of three letters. They gathered their timing samples by instructing users to copy a 683-character writing sample multiple times. Gunetti and Picardi [11] later expanded on this work by experimenting with authentication of users based on samples of free text gathered over several months in whatever setting was most comfortable for the user.

Much of the prior work focused on analysis of users' keystrokes as they typed several repetitions of short, pre-defined text or numeric sequences. This allowed the users to become more and more familiar with the text as they went on and develop a consistent, distinctive pattern to how they typed the text and was a very effective way of reducing much of the variability inherent in typing due to user distraction, outside influences, etc. Variability will be much higher when dealing with free-text as opposed to short,

structured inputs, but the input will be a more realistic model of an individual to study for application to "real-world" problems. In particular, Tappert *et al*. [6] [5] demonstrate methods for authenticating users based on long-form (over 600 characters) copy and free text input and collected their data in the same manner as Gunetti, but rather than classifying purely on feature vector distance as in [11], they used *k*-nearest neighbors clustering on these features.

Clarke and Furnell [12] were among the first to investigate authenticating a mobile phone user via keystroke analysis. Users in their study entered a series of personal identification numbers (PIN), as well as short alphabetic messages into a numeric keypad-equipped handset. Following sample entry on the handset, the timing data was downloaded to a computer that processed the features using a series of neural networks for classification. While success rates around 85 percent were obtained, Maiorana *et al.* observe that neural networks are not a practical tool for mobile authentication use due to high training and processor cost. Instead, they combined a distance classifier, Bayes classifier, support vector machines and principal components analysis to build a system with much lower cost requirements both for both training and on-line classifying [13]. Using this system they were able to achieve roughly the same authentication success rate as Clarke and Furnell.

Johansen [14] used a touchscreen numeric keypad to perform user classification. Their study also explored whether or not the classifier can be "fooled" using a program written to generate imitation keystroke patterns. Trojahn and Ortmeier [15] recently obtained impressive results in their experiment using both a touchscreen numeric keypad and touchscreen QWERTY alphabetic keypad. In both cases, the input consisted of short (11-12 character length) numbers or equal length phrases. To the best of our knowledge, no one has yet performed an experiment studying classification based on touchscreen, alphabetic free-text input.

Author(s)	Keyboard Type	Input Type	Analysis Method	Results
Gaines, *et al.*	Hardware QWERTY	Fixed text	Statistical	Error free authentication
Monrose, *et al.*	Hardware QWERTY	Free and fixed text	Statistical	88–92% accuracy
Bergandano, *et al.*	Hardware QWERTY	Fixed text	Statistical	96–99% accuracy
Gunetti, *et al.*	Hardware QWERTY	Free text	Statistical	4.6% false alarm rate
Tappert, *et al.*	Hardware QWERTY	Free and fixed text	Statistical	1% equal error rate
Clarke, *et al.*	Hardware 12-key numeric	Fixed numbers and text	Neural network	12.8% equal error rate
Maiorana, *et al.*	Hardware 12-key numeric	Fixed text	Statistical	13.6% equal error rate
Johansen	Touchscreen 12-key numeric	Free text	Statistical	8.7% equal error rate
Trojahn, *et al.*	Touchscreen 12-key numeric	Fixed text	Statistical	9% false alarm rate
Trojahn, *et al.*	Touchscreen QWERTY	Fixed text	Statistical	12% false alarm rate

Table 1. Summary of prior work.

C. *K*-NEAREST NEIGHBORS ALGORITHM

The *k*-nearest neighbors algorithm (*k*NN) [16], classifies a data point *x* by looking at the *k* points closest to *x* according to some distance metric and labeling *x* based on the class of those neighbors. The parameter *k* is chosen to be odd so a simple majority vote can be employed using the classes of the neighbors. The *k*NN algorithm obtains nice results when items in a single class tend to cluster together in the feature space using an appropriate distance metric. A common distance metric employed is Euclidian distance where the distance between two points x_m and x_n is defined as:

$$D(\mathrm{x}_m, \mathrm{x}_n) = \sqrt{\sum_i \left| x_{m,i} - x_{n,i} \right|^2} \quad .$$

Since the distances of values from different categories, such as comparing the duration of a press of the letter "A" to the duration of a press of the letter "L", cannot be directly compared without skewing the results, it is standard practice to normalize all of the values for these features prior to doing any distance comparison. In the implementation of our *k*-nearest neighbors classifier, normalization was accomplished by dividing each category's data points by the category's span; however, normalization is usually done by calculating the mean μ_i and standard deviation σ_i for a point $x_{m,i}$ and using the formula

$$\frac{x_{m,i} - \mu_i}{\sigma_i} \quad .$$

k-nearest neighbors was used in [6] [5] [7] to authenticate users based on models created specifically for each user. If user one (or someone claiming to be user one) tried to log-in, the classifier would load a two-class model, consisting of a feature space for user one, along with a feature space created from all other users' keystroke data. The classifier would place the data from the log-in attempt in this model and compare distances to the *k*-nearest neighbors. If a majority of the closest data points were in class "user one", the log-in attempt would be valid, else the system would reject the imposter. The studies also tested a model using a weighting system, where the contribution of nearest neighbors to

the voting were weighted based on their distance from the point being classified on the theory that closer points were more likely to represent the true class of the point in question.

III. EXPERIMENT DESIGN

This chapter will discuss the methodology and design of our study.

A. DATA COLLECTION

Data collection was performed on Nexus 4 smartphones, manufactured by LG Electronics and running the Android 4.2 operating system. Aside from very early versions, Android has built-in security designed to prevent the collection of keystroke data from users. We explored several options for bypassing this security in order to collect the data we needed, including altering the kernel and using a browser-based collection application, but found that we could simply load a custom keyboard application that contained our collection code onto the phone and give the application explicit permission to collect the data we needed.

Each subject was asked to create four typing samples. Two fixed-text samples were based on a pre-written business email (see Appendix B) and two free-text samples were authored spontaneously by the subject. Subjects were provided instructions a few days before data collection in order to allow to prepare topics or themes to guide their free-text generation (e.g., to avoid writer's-block during data collection) but they were not permitted to bring pre-written samples to copy for their free text.

Two versions of each fixed-text and free-text samples were collected, one typed with the phone in the vertical (portrait) orientation and the other typed with the phone in the horizontal (landscape) orientation.

No time limit was placed on data collection for any text sample. Upon completion of each sample, the data was saved in tab-delimited format to the phone's internal memory and later collected for processing.

B. RAW DATA

Tappert *et al*. [5] build a feature vector for each user measuring average duration of key press and standard deviation for each letter in the alphabet and numbers 0–9, as well as special keys such as space and delete. The feature vector also contained average

type-1 and type-2 transition times and standard deviations for several common digraphs and trigraphs of alphabetic characters. We closely followed this approach, but used a list of the most common digraphs and trigraphs in the English language mined from numerous top classic works of literature [17] to study whether using more *n*-grams would produce better classification results. Our full list of features can be found in Appendix A.

Raw data consisted of:

- Key pressed
- Timestamp of key press
- Timestamp of key release

Table 2 shows an example of the raw data collected. From this data, all duration and type-1 and type-2 timing data can be generated for any character or set of n-grams.

Key	Time pressed	Time Released
s	171817498919	171948828083
i	172160120620	172284277495
r	172747668307	172820856160
,	176803520588	176891388741
	178524533865	178634468247
c	180236421550	180334178302
a	180658548819	180761219352
n	181344707831	181424213400
	181681439099	181757923151
y	182168239038	182272923915
o	182785513568	182896272000
u	183058304656	183184994722

Table 2. Raw data example.

IV. ANALYSIS

This chapter will describe our feature construction and data analysis methodology.

A. FEATURE EXTRACTION

Our raw data was processed with Python scripts to generate samples consisting of features used during training and testing. The features used in our analysis consist of:

- Duration
- Type-1 and type-2 transitions between bigrams
- Type-2 transitions between trigrams

Data was split into training and test data. For each, features were extracted containing three columns of data. The first contained one entry for each key on the keyboard, including letters, numbers, punctuation, and special characters such as delete. The second column contained the average duration time in milliseconds of the press of each key. The third column contained the standard deviation of the duration time for each key. Table 3 shows an example of a duration feature file. The files for *n*-gram transitions followed the same format, with the first column containing each of the two- or three-character sets, the second column containing the average transition time in milliseconds, and the third column containing the standard deviation of the transition time. Table 4 shows an example of an *n*-gram feature file. A global entry was calculated for each sample, consisting of the average duration time and standard deviation or average transition time and standard deviation for the entire file. When calculating the times and standard deviations, a value of 0.05 was inserted as a default value if a particular key or *n*-gram was not seen in the typing sample being processed in order to ensure each key had some small, non-zero probability of being observed under any classifier.

Key Pressed	Duration Time (ms)	Std. Deviation (ms)
Global	125359534.4	26037704.04
a	144162979	18996714.94
b	132633908	19477968.23
c	104753548	17206444.97
d	114459026	21942023.64
e	121318480	21847877.46
f	116461926	33506453.14
g	126501856.3	24237631.21
h	138550281.6	9141359.669
i	114632912.2	18309662.65
j	0.05	0.05
k	118065007.8	30226373.56

Table 3. Example duration training file

Bigram	Duration Time (ms)	Std. Deviation (ms)
Global	566001705.2	635577986.5
OF	131298643	671450
EL	528948573	0.05
ED	116374180	0.05
VE	266290249	123105555.5
AD	0.05	0.05
SO	140683657	686708
SI	123927973	24156875
MO	209583396	0.05
GO	0.05	0.05
MY	0.05	0.05
WE	222056055.3	126685731.5

Table 4. Example bigram type-1 transition test file

Since typing is an activity that is prone to multiple interruptions to review text being copied, take sips of coffee, answer questions from co-workers, etc., we set

thresholds for transition times in order to avoid data being skewed by outliers caused by these types of routine activities. We set the thresholds by making one pass through the raw data to calculate a global standard deviation, then experimenting with various multiples of that global standard deviation as a cutoff for inclusion in the processed training or test file.

B. TRAINING AND TEST SETS

In order to ensure valid classification results, the raw data for each user's typing samples was split into separate training and test sets prior to processing as described above. Two different splits were used in order to compare classification performance for each.

The first method was an 80/20 split where the first 80 percent of the sample was processed for a training set and the last 20 percent of the sample processed for a test set. For further validation of this method, we flipped this and made an additional set, using the last 80 percent of a sample for training and the first 20 percent for testing.

The second method was only able to be performed on the first group of typing samples, portrait orientation copy text, due to both time constraints and the nature of the samples. Unlike the participant-authored free text samples, the copy text samples all contained the same content, broken into four different paragraphs. We constructed four corresponding training and test sets for each user by using each paragraph individually as a test set, with the remaining three paragraphs serving as training data.

C. CLASSIFICATION

We used the Orange toolset with Python for *k*NN classification. Continuous values were normalized within the classifier, Euclidian distance was used for measurement, and *k* was set to 5. Classifier performance was compared with a Naïve Bayes classifier and a Random Forest classifier, but these algorithms almost never delivered a correct classification and we do not report these negative results here.

Used naively, the initial classification via *k*NN returned results based on what user the classifier thinks each individual feature belonged to. For example, it would tell

us what user it thought the values for a press of the "a" key belonged, what user the values for a press of the "b" key belonged to etc. While this is useful, it does not get us to our goal, which is to input the entire typing sample into the classifier and return single user identification based on *all* the features together. Instead, the individual classifications were combined using majority vote. An example of the results can be seen in Table 5.

	122	**301**	**323**	**336**	**347**	**362**	**372**	**381**	**388**	**392**
122	7	3	10	6	3	5	9	1	4	3
301	1	19	5	2	3	2	6	0	11	2
323	5	3	9	6	5	5	9	3	4	2
336	3	4	5	3	8	7	8	7	4	2
347	2	9	5	3	10	3	5	2	9	3
362	1	4	8	3	5	9	10	3	4	4
372	5	3	9	3	5	5	12	2	4	3
381	1	10	5	3	6	2	7	5	4	8
388	1	9	5	2	4	3	5	0	17	5
392	1	9	5	2	8	3	7	1	8	7

Table 5. Example *k*-nearest neighbor classifier results. Rows are true users and columns are predictions for each individual feature.

For example, using the results from Table 5, the classifier correctly classified the test sets for users 301, 323, 347, 372, and 388 since the number of features identified as belonging to their class was more than any other individual user identified with their data.

Several runs were done for each user's data with each feature set in order to find an ideal threshold to use to trim outlier values during pre-processing. We settled on cutting outlier values in our feature vectors that varied more than two standard deviations from the mean and that threshold is what the results reported in this thesis are based on.

V. RESULTS

Participation for the study was solicited via email to students and faculty of the Computer Science Department and Cyber Academic Group at the Naval Postgraduate School. All subject participation was voluntary. Before data collection, all subjects were asked to complete a small demographic questionnaire covering their smartphone use and handed-ness. Subject data was tracked using a randomly assigned subject identifier to anonymize their data.

Classification results for training/test data splits for the fixed-text samples can be found in Table 6 and results for the same data splits for the free-text samples can be found in Table 7. 80/20 means the first 80 percent of the sample was used for training and the last 20 percent was used for testing. 20/80 means the last 80 percent of the sample was used for training and the first 20 percent was used for testing. Portrait and landscape refers to the orientation the phone was held in during typing.

Duration proved to be the most accurate feature for identification for most of these typing samples, correctly identifying participants 70 percent of the time in portrait fixed-text, portrait free-text and landscape fixed-text samples. However, it did not perform well with landscape free-text samples, as performance fell to a 35 percent correct identification rate.

	Portrait 80/20	Portrait 20/80	Landscape 80/20	Landscape 20/80
Duration	0.7	0.7	0.7	0.7
Bigram Type 1 Transitions	0.3	0.4	0.6	0.7
Bigram Type 2 Transitions	0.4	0.4	0.2	0.5
Trigram Type 2 Transitions	0.1	0.3	0.1	0.4
All Features Combined	0.1	0.5	0.5	0.5

Table 6. User identification on fixed-text samples (80/20 split).

	Portrait 80/20	Portrait 20/80	Landscape 80/20	Landscape 20/80
Duration	0.7	0.7	0.3	0.4
Bigram Type 1 Transitions	0.4	0.4	0.1	0.4
Bigram Type 2 Transitions	0.5	0.5	0.2	0.8
Trigram Type 2 Transitions	0.4	0.6	0.2	0.3
All Features Combined	0.3	0.7	0.2	0.4

Table 7. User identification on free-text samples (80/20 split).

Classification results for the training/test splits by paragraph can be found in Table 8. The only typing sample used to test this data split was the portrait orientation copy text. The paragraph number refers to the paragraph used as the test paragraph. Duration was once again the most reliable feature for classification, with performance rising from 50 percent at the beginning of the typing sample to 80 percent by the last paragraph tested.

	Paragraph 1	Paragraph 2	Paragraph 3	Paragraph 4
Duration	0.5	0.6	0.7	0.8
Bigram Type 1 Transitions	0.2	0.1	0.2	0.5
Bigram Type 2 Transitions	0.2	0.4	0.5	0.6
Trigram Type 2 Transitions	0.1	0.1	0.2	0.2
All Features Combined	0.1	0.3	0.4	0.5

Table 8. User identification on fixed-text samples (paragraph split).

A. DISCUSSION

Given that most prior work in keystroke-based classification found *n*-gram transition feature vectors worked much better than duration feature vectors for identification and authentication, it was surprising to see the opposite observed in this study. While future study is warranted, we propose some hypotheses for why this may be so.

The first is that the feature space we chose for *n*-gram transitions may be too big. Russell and Norvig point out that nearest neighbors' algorithms are well suited for situations with lower feature counts and robust data sets, but as the dimensionality of the feature set starts to rise, the nearest neighbors begin to fall farther and farther away from the data point in question [16]. With 300 bigrams and 150 trigrams, the size of the "neighborhood" in which we are looking for neighbors to poll to answer our classification question becomes very large and the probability that the closest neighbors are actually representative of the user class is small.

Another contributor to duration being a better identifying marker may be the vastly different mechanics of typing on a smartphone touchscreen as opposed to typing on a keyboard laying on a desk or table. Most of the study participants typed in what has become the most widely used way of using a smartphone for text entry, which is to hold

the phone in both hands and use their thumbs to type, however one participant did lay the phone flat on the table in front of him, using his index fingers to type. The smaller size of touchscreen keyboard keys, along with the relatively large surface area of a human thumb tip or fingertip often leads to multiple typing errors in any given text entry session. The increase in error rate over using a conventional keyboard, along with the different biomechanics may lead to a natural tendency to produce a more pronounced duration model. Studying the differences in keystroke rhythm models produced by individuals using both traditional and touchscreen keyboards would be an interesting avenue for further study.

The small size and relative homogeneity of the participant group also may have played a part in weighting classification success toward the duration features. A larger and more diverse mix of participants, including people who used touchscreen smartphones less frequently than our group or used this particular brand of smartphone regularly may have led to different results.

The common trend across the data for both duration and transitions was an improvement in identification success as we moved further into the document. This was particularly evident when splitting the training and test sets by paragraph in the portrait orientation copy text sample. Using the first paragraph as a test set yielded only a 50 percent success rate, but by the last paragraph, we identified the user correctly 90 percent of the time. Of course, using this method of splitting the training and test sets to compare classification results among several users only works when each user is typing exactly the same thing. This makes it less useful for identification in true free text entry situations, but does indicate that it is important to give the user time to become familiar with the equipment being used before creating a model or profile for classification, as well as periodically updating and refreshing the user model to account for possible changes in the user's typing habits.

VI. CONCLUSION

Our goal was to identify the user of a touchscreen smartphone based solely on the analysis of the user's keystroke timing data created as they both copied pre-written text and typed unscripted free text into a text editor on the phone. We gathered data from ten users who typed two copy-text samples and two free-text samples into an LG Nexus 4 smartphone, creating raw timing data based on the press and release times of each key the users pressed. We converted the raw timing data into feature vectors based on duration of key press and length of bigram and trigram transitions. These features were used in the Orange toolset implementation [18] of a *k*-nearest neighbors algorithm to identify the users. We obtained a 70 percent success rate identifying the user in three out of the four typing samples provided by each user; however that rate fell to 40 percent in the fourth typing sample. We learned that the *n*-gram transition feature vectors were not as successful as the duration feature vectors in classification, possibly due to an overly-large feature space.

A. FUTURE WORK

Several opportunities exist for future work based on this study.

- Using 10-fold validation in order to confirm test results.
- Pruning the *n*-gram transition space in order to test whether or not duration is actually a better feature to base user identification on when using smartphone text input or if the feature space was simply too large to allow for accurate classification.
- Using the raw data collected in this study, create authentication models for each user as described in part B of the introduction chapter and in Tappert [5] in order to ask the question "Is this user X or not?" as opposed to the question asked in this study, "Who is this user?".
- Recruit a significantly larger and more diverse pool of participants for this study in order to determine the effect such a change on the user count and smartphone usage level would have on identification.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. FEATURES

A. KEYS MONITORED FOR DURATION OF PRESS

- a-z
- 0-9
- ? ! @ # $ % ^ & * ( ) \ /
- <space>
- <enter>
- <delete>

B. *N*-GRAMS MONITORED FOR TRANSITION TIMES

Most common bigrams including space (sample includes 6442495 bigrams)

E - 245521 3.81%	M - 54707 0.85%	H - 34308 0.53%
T - 188459 2.93%	AT - 54679 0.85%	ME - 33498 0.52%
HE - 158681 2.46%	ON - 54317 0.84%	P - 33488 0.52%
TH - 155382 2.41%	B - 52647 0.82%	NT - 33309 0.52%
D - 151912 2.36%	HI - 51487 0.80%	EA - 33115 0.51%
A - 137885 2.14%	EN - 50680 0.79%	AL - 31638 0.49%
T - 131548 2.04%	TO - 48934 0.76%	L - 31413 0.49%
S - 127468 1.98%	NG - 48452 0.75%	L - 31271 0.49%
H - 103608 1.61%	C - 46867 0.73%	A - 31181 0.48%
S - 97862 1.52%	IS - 46795 0.73%	LL - 30942 0.48%
IN - 94900 1.47%	IT - 46750 0.73%	NE - 29606 0.46%
N - 90466 1.40%	F - 44074 0.68%	N - 28561 0.44%
AN - 89239 1.39%	OR - 43306 0.67%	TI - 27954 0.43%
W - 87123 1.35%	F - 42456 0.66%	DE - 27149 0.42%
ER - 84372 1.31%	AS - 41550 0.64%	NO - 27144 0.42%
I - 78395 1.22%	G - 40856 0.63%	BE - 25716 0.40%
R - 71433 1.11%	TE - 40346 0.63%	RO - 25665 0.40%
RE - 69581 1.08%	ES - 40152 0.62%	R - 25511 0.40%
O - 69365 1.08%	D - 39144 0.61%	WA - 25409 0.39%
Y - 69357 1.08%	AR - 38194 0.59%	WH - 25352 0.39%
ND - 64917 1.01%	ST - 38056 0.59%	M - 24953 0.39%
O - 61336 0.95%	LE - 37620 0.58%	HO - 24900 0.39%
OU - 59917 0.93%	SE - 36629 0.57%	Y - 24563 0.38%
HA - 58931 0.91%	OF - 35593 0.55%	EL - 24556 0.38%
ED - 56774 0.88%	VE - 35534 0.55%	AD - 24154 0.37%

Most common bigrams in the beginning of words (sample includes 1226563 trigrams)

TH - 125714 10.25%	SO - 12480 1.02%	SI - 6781 0.55%
AN - 50095 4.08%	MO - 12065 0.98%	GO - 6575 0.54%
TO - 40128 3.27%	AS - 12000 0.98%	MY - 6421 0.52%
HE - 39426 3.21%	WE - 11936 0.97%	SU - 6383 0.52%
OF - 34439 2.81%	SE - 11028 0.90%	DA - 6012 0.49%
IN - 28313 2.31%	CA - 10927 0.89%	FI - 5343 0.44%

HI - 26851 2.19% | BU - 10719 0.87% | CH - 5325 0.43%
HA - 26660 2.17% | ME - 10697 0.87% | LA - 5276 0.43%
WH - 24883 2.03% | ST - 10569 0.86% | PE - 5042 0.41%
A - 23513 1.92% | DO - 10360 0.84% | EX - 4975 0.41%
BE - 22023 1.80% | AT - 9867 0.80% | FE - 4805 0.39%
WA - 20721 1.69% | LI - 9455 0.77% | PO - 4757 0.39%
YO - 20708 1.69% | DE - 9078 0.74% | BY - 4756 0.39%
NO - 19878 1.62% | PR - 9064 0.74% | MI - 4720 0.38%
CO - 19722 1.61% | WO - 9033 0.74% | UP - 4719 0.38%
WI - 19434 1.58% | IS - 8833 0.72% | GR - 4691 0.38%
I - 18192 1.48% | FR - 8512 0.69% | NE - 4654 0.38%
SH - 16490 1.34% | HO - 8188 0.67% | OU - 4632 0.38%
SA - 15659 1.28% | DI - 8171 0.67% | UN - 4629 0.38%
IT - 15521 1.27% | LO - 7779 0.63% | CR - 4578 0.37%
FO - 15241 1.24% | LE - 7583 0.62% | EV - 4517 0.37%
RE - 15029 1.23% | AR - 7413 0.60% | TR - 4428 0.36%
ON - 14957 1.22% | S - 7372 0.60% | BR - 4323 0.35%
MA - 14752 1.20% | FA - 7149 0.58% | BA - 4295 0.35%
AL - 12594 1.03% | PA - 6801 0.55% | TA - 4134 0.34%

Most common bigrams in the end of words (sample includes 1226563 trigrams)
HE - 101821 8.30% | TH - 14891 1.21% | UR - 5982 0.49%
ED - 53080 4.33% | AD - 14338 1.17% | MY - 5978 0.49%
ND - 51591 4.21% | VE - 14022 1.14% | TY - 5944 0.48%
NG - 39647 3.23% | ST - 13369 1.09% | TS - 5844 0.48%
ER - 38873 3.17% | NT - 13130 1.07% | ET - 5778 0.47%
TO - 37868 3.09% | LE - 13047 1.06% | SO - 5498 0.45%
AT - 33811 2.76% | LD - 12476 1.02% | RT - 5286 0.43%
OF - 32699 2.67% | ID - 12256 1.00% | KE - 5192 0.42%
IS - 29806 2.43% | CH - 12086 0.99% | DE - 5097 0.42%
AS - 26232 2.14% | CE - 11760 0.96% | AL - 5047 0.41%
IN - 25271 2.06% | OT - 11697 0.95% | BY - 4857 0.40%
RE - 24297 1.98% | SE - 11433 0.93% | IR - 4769 0.39%
A - 23513 1.92% | NE - 10613 0.87% | LF - 4555 0.37%
ON - 22656 1.85% | OW - 9434 0.77% | US - 4472 0.36%
EN - 19830 1.62% | AY - 8627 0.70% | DS - 4406 0.36%
LL - 19094 1.56% | IM - 8566 0.70% | HO - 4228 0.34%
ES - 18196 1.48% | RY - 7904 0.64% | AR - 4211 0.34%
I - 18192 1.48% | S - 7372 0.60% | NS - 4183 0.34%
LY - 17917 1.46% | HT - 7283 0.59% | EE - 4178 0.34%
OR - 17357 1.42% | RS - 7167 0.58% | NO - 4178 0.34%
ME - 17309 1.41% | SS - 7124 0.58% | RD - 3814 0.31%
UT - 16237 1.32% | OM - 7054 0.58% | WN - 3793 0.31%
IT - 15953 1.30% | TE - 7045 0.57% | GE - 3681 0.30%
OU - 15459 1.26% | EY - 6965 0.57% | CK - 3635 0.30%
AN - 15178 1.24% | BE - 6501 0.53% | DO - 3421 0.28%

Most common bigrams not including space (sample includes 5215931 bigrams)
TH - 167258 3.21% | TE - 42514 0.82% | SI - 26473 0.51%
HE - 159235 3.05% | TI - 40982 0.79% | SO - 26287 0.50%
IN - 95194 1.83% | SE - 39804 0.76% | RA - 26255 0.50%
ER - 90930 1.74% | AR - 39143 0.75% | EC - 26225 0.50%

```
 AN - 90006  1.73% | LE - 38271  0.73% | YO - 25772  0.49%
 RE - 71383  1.37% | OF - 37388  0.72% | BE - 25717  0.49%
 ND - 66692  1.28% | SA - 36088  0.69% | AD - 25681  0.49%
 ED - 66683  1.28% | VE - 35538  0.68% | SS - 25358  0.49%
 HA - 64086  1.23% | ME - 33804  0.65% | DA - 25316  0.49%
 ES - 63216  1.21% | AL - 33710  0.65% | LI - 24618  0.47%
 OU - 60474  1.16% | NO - 32644  0.63% | OM - 24394  0.47%
 TO - 58346  1.12% | NE - 31669  0.61% | RT - 24148  0.46%
 AT - 56683  1.09% | LL - 31649  0.61% | EW - 24054  0.46%
 EN - 55832  1.07% | EL - 31405  0.60% | DI - 24030  0.46%
 ON - 55755  1.07% | SH - 30650  0.59% | CO - 23975  0.46%
 EA - 55459  1.06% | OT - 30566  0.59% | EE - 23940  0.46%
 NT - 54694  1.05% | TT - 30218  0.58% | MA - 23817  0.46%
 ST - 54195  1.04% | RO - 29790  0.57% | EM - 23453  0.45%
 HI - 53885  1.03% | DE - 29619  0.57% | AI - 22856  0.44%
 NG - 49388  0.95% | TA - 28744  0.55% | UT - 22840  0.44%
 IS - 49156  0.94% | DT - 28373  0.54% | WI - 22502  0.43%
 IT - 48057  0.92% | RI - 28017  0.54% | CE - 22365  0.43%
 AS - 45974  0.88% | WA - 26889  0.52% | OW - 22174  0.43%
 OR - 45043  0.86% | WH - 26749  0.51% | CH - 22152  0.42%
 ET - 42573  0.82% | HO - 26702  0.51% | RS - 21231  0.41%

Most common trigrams including space (sample includes 6442494 trigrams)
 TH - 125714  1.95% | WH - 24883  0.39% | OR - 17357  0.27%
HE - 101821  1.58% | RE - 24297  0.38% | ME - 17309  0.27%
THE - 98530  1.53% | A - 23513  0.36% | E H - 17282  0.27%
ED - 53080  0.82% | E S - 23064  0.36% | D A - 16997  0.26%
ND - 51591  0.80% | HAT - 22861  0.35% | SH - 16490  0.26%
 AN - 50095  0.78% | ON - 22656  0.35% | FOR - 16426  0.25%
AND - 48312  0.75% | E A - 22344  0.35% | UT - 16237  0.25%
 TO - 40128  0.62% | BE - 22023  0.34% | S T - 16139  0.25%
NG - 39647  0.62% | N T - 21385  0.33% | IT - 15953  0.25%
 HE - 39426  0.61% | HIS - 20975  0.33% | ERE - 15807  0.25%
ER - 38873  0.60% | T T - 20809  0.32% | SA - 15659  0.24%
ING - 38182  0.59% | WA - 20721  0.32% | IT - 15521  0.24%
TO - 37868  0.59% | YO - 20708  0.32% | OU - 15459  0.24%
 OF - 34439  0.53% | YOU - 20678  0.32% | FO - 15241  0.24%
AT - 33811  0.52% | E W - 19929  0.31% | AN - 15178  0.24%
OF - 32699  0.51% | NO - 19878  0.31% | WAS - 15122  0.23%
IS - 29806  0.46% | EN - 19830  0.31% | RE - 15029  0.23%
D T - 28343  0.44% | CO - 19722  0.31% | E C - 15001  0.23%
 IN - 28313  0.44% | WI - 19434  0.30% | ON - 14957  0.23%
 HI - 26851  0.42% | THA - 19227  0.30% | TH - 14891  0.23%
 HA - 26660  0.41% | LL - 19094  0.30% | MA - 14752  0.23%
E T - 26459  0.41% | ES - 18196  0.28% | AD - 14338  0.22%
AS - 26232  0.41% | I - 18192  0.28% | D H - 14309  0.22%
HER - 26208  0.41% | LY - 17917  0.28% | E O - 14113  0.22%
IN - 25271  0.39% | S A - 17434  0.27% | VE - 14022  0.22%

Most common trigrams not including space (sample includes 5215930
trigrams)
THE - 104376  2.00% | VER - 12279  0.24% | ESA -  9302  0.18%
AND - 48638  0.93% | TER - 12274  0.24% | EVE -  9271  0.18%
ING - 38500  0.74% | ALL - 12021  0.23% | NCE -  9249  0.18%
HER - 30219  0.58% | ION - 11289  0.22% | EDA -  9239  0.18%
```

THA - 24760 0.47%	FTH - 11247 0.22%	AID - 9213 0.18%
HAT - 23177 0.44%	STH - 11210 0.21%	HIN - 9203 0.18%
HIS - 21322 0.41%	OFT - 11144 0.21%	NDT - 9190 0.18%
YOU - 20873 0.40%	HAD - 11113 0.21%	HEN - 9184 0.18%
ERE - 20173 0.39%	REA - 11110 0.21%	BUT - 9178 0.18%
DTH - 18382 0.35%	EST - 10757 0.21%	OME - 9149 0.18%
ENT - 17684 0.34%	ERS - 10698 0.21%	ILL - 9120 0.17%
ETH - 16638 0.32%	GHT - 10475 0.20%	AST - 9111 0.17%
FOR - 16484 0.32%	ESS - 10280 0.20%	RTH - 9067 0.17%
NTH - 16221 0.31%	HIM - 10191 0.20%	OUL - 8901 0.17%
THI - 15782 0.30%	EAR - 10173 0.20%	ATT - 8848 0.17%
SHE - 15440 0.30%	EAN - 9983 0.19%	STO - 8836 0.17%
WAS - 15277 0.29%	AVE - 9720 0.19%	SAI - 8753 0.17%
HES - 14937 0.29%	ONE - 9672 0.19%	ATH - 8683 0.17%
ITH - 14829 0.28%	HEC - 9606 0.18%	OUN - 8664 0.17%
TTH - 14454 0.28%	TIN - 9590 0.18%	ERT - 8579 0.16%
OTH - 14352 0.28%	RES - 9485 0.18%	SAN - 8556 0.16%
INT - 13802 0.26%	HEW - 9480 0.18%	HOU - 8465 0.16%
NOT - 13411 0.26%	ONT - 9445 0.18%	OUR - 8460 0.16%
WIT - 13084 0.25%	ATI - 9437 0.18%	OUT - 8436 0.16%
EDT - 12922 0.25%	HEM - 9363 0.18%	HEA - 8393 0.16%

List from [13].

APPENDIX B. FIXED-TEXT SAMPLE

Below is the fixed-text sample used in our experimentation.

Sir,

Can you clear up some confusion on a few issues before next week's budget meeting, please? We have a difference of opinion among our team about which direction to take on some of the talking points we discussed earlier. Any guidance you can give us would be very helpful.

First, is the major project in California going ahead as scheduled? There have been several different dates thrown out by various team leads, with one quoting tomorrow as the start date, and we just need clarification about the actual starting date for that project. The hard deadline is approaching fast, so this is time sensitive.

Second, will the renovation of the main office building be funded under this year's budget or next year's budget? We were operating under the assumption that this project was already fully funded, but the accounting department has given us some pushback about starting, saying we don't have the money. The quotes we got were reasonable, but maybe you got a different quote. Of course, its possible accounting is just using fuzzy math, too!

Finally, will vacation be allowed before the end of the uptown project? I definitely understand that we have been building toward our goal for the better part of ten years now and that organizing so many state and local government agencies along with all the associated neighborhood committees into a group able to come to a consensus has taken a monumental effort, but some of our team members are on the verge of burnout. I believe a short break would work wonders for our technical team.

Thanks for your assistance clarifying these matters.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Advanced Micro Devices, Inc.,"AMD face login," 2014. [Online]. Available: http://www.amd.com/us/consumer/software/pages/face-login.aspx. [Accessed 9 January 2014]

[2] C. Tappert, M. Villani, S.-H. Cha, G. Ngo, J. Simone and H. St. Fort, "Keystroke biometric recognition studies on long-text input," in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, New York, 2006.

[3] W. Parker, "Authentication of a smartphone user via gait analysis," M.S. thesis, Dept. of Comp. Sci., Naval Postgraduate School, Monterey, CA, 2014.

[4] V. Nguyen, "Authentication of a smartphone user via RSSI geolocation," M.S. thesis, Dept. of Comp. Sci., Naval Postgraduate School, Monterey, CA, 2014.

[5] R. V. Yampolskiy and V. Govindaraju, "Behavioral biometrics: a survey and classification," *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[6] R. S. Gaines, W. Lisowski, S. J. Press and N. Shapiro, "Authentication by keystroke timing: Some preliminary results," Rand Corporation, Santa Monica, CA, 1980.

[7] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351–359, 2000.

[8] F. Bergadano, D. Gunetti and C. Picardi, "User authentication through keystroke dynamics," *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 367–397, 2002.

[9] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312–347, 2005.

[10] C. C. Tappert, M. Villani and S.-H. Cha, "Keystroke biometric identification and authentication on long-text input," in *Behavioral Biometrics for Human Identification*. Hershey, PA: Medical Information Science Reference, 2009, pp. 342–367.

[11] N. Clarke and S. Furnell, "Authenticating mobile phone users using keystroke analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 1–14, 2007.

[12] E. Maiorana *et al*., "Keystroke dynamics authentication for mobile phones," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, TaiChung, 2011.

[13] U. A. Johansen, "Keystroke dynamics on a device with touchscreen," M.S. thesis, Dept. of Comp. Sci. and Media Tech., G. U. College, Gjovik, Norway, 2012.

[14] M. Trojahn and F. Ortmeier, "Biometric Authentication Through a Virtual Keyboard for Smartphones," *International Journal of Computer Science & Information Technology*, vol. 4, no. 5, pp. 1–6, 2012.

[15] Chalmers Department of Computer Science and Engineering, "Mono-, bi and trigram frequency for English." [Online]. Available: http://www.cse.chalmers.se/edu/year/2010/course/TDA351/ass1/en_stat.html. [Accessed 17 December 2013].

[16] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education, Inc., 2010.

[17] R. S. Zack, C. C. Tappert and S.-H. Cha, "Performance of a Long-Text-Input Keystroke Biometric Authentication," in *2010 Fourth IEEE International Conference on Biometrics Compendium*, Washington DC, 2010.

[18] J. Demšar, T. Curk and A. Erjavec, "Orange: Data Mining Toolbox in Python," *Journal of Machine Learning Research*, vol. 14, no. Aug, pp. 2349–2353, 2013.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California