Numeric-Passcode Keystroke Biometric Studies on Smartphones

Michael J. Coakley, John V. Monaco, and Charles C. Tappert
Seidenberg School of CSIS, Pace University, White Plains, New York

Abstract—A keystroke biometric classification system traditionally used on data captured from physical keyboards associated with laptops and personal computers, was extended to evaluate biometric data extracted from mobile devices. The results were compared to the results of similar studies utilizing the same data inputs. Additional results were extracted; including results tied to features native to mobile devices, as well as results of the combined keystroke and mobile (touchscreen) feature file. The best results observed in this study were associated with the touchscreen biometric features, which achieved a top Equal Error Rate (EER) of 4.9%. As a result of the combined feature results being uncharacteristically higher than the touchscreen features alone, further study will include identifying other distance metrics that would better suit the feature data associated with this study.

Keywords—biometrics; pattern recognition; machine learning; keystroke dynamics; mobile devices; user authentication

I. INTRODUCTION

A. Biometrics Overview

The proliferation of passcodes and Personal Identification Numbers (PIN) associated with computer accounts, ATM access, building and office accessibility and now mobile devices, demands that an additional layer of user-identity information be incorporated in order to properly securitize access to the device or entity associated with its corresponding keypad. Whereas traditional keystroke biometric data associated with physical keyboards has been observed to provide adequate results for authenticating users, touchscreen biometric data associated with mobile devices such as mobile phones and tablets has the potential to supplement the results to an even greater level of acceptability. This study aims to identify the biometric performance achievable from features historically associated with a physical keyboard, the performance of features associated with touchscreen keyboards, and the performance of the combined set of features, in an attempt to compare the results and identify a preferred approach for biometric authentication on a mobile device.

In its simplest terms, keystroke biometrics are the unique individual typing characteristics each user has when typing on a keyboard. A user’s typing rhythm or cadence can be extracted, stored and then compared to subsequent samples in an attempt to authenticate or identify a user utilizing the device associated with the keyboard. The extraction and storage of this biometric data is done via a keystroke biometric system.

Once the feature data is stored, it can then be run through a number of classification algorithms, distance based classifiers and/or machine learning technologies in order to authenticate and identify participating users [1]. The effectiveness of the biometric system and its corresponding algorithmic and/or distance classifier is expressed via the commonly utilized Equal Error Rate (EER). The EER is the point on the Receiver Operating Characteristic (ROC) graph where False Acceptance Rate (FAR) and False Rejection Rate (FRR) intersect. In other words, it is where the rate of False Acceptance is equal to the rate of False Rejection.

B. Relevancy of Study

Keystroke biometric as a means for authentication is appealing due to its non-intrusive and transparent approach. Indeed, with the advancement of mobile phone technology, consisting of enhanced computing power as well as storage capacity, a keystroke biometric authentication system could run on today’s devices without impeding the performance of the device in the slightest. Additionally, use of these mobile devices has exploded in recent years, with some studies estimating that there are now more smartphones than there are people on the earth [12].

As these devices become more pervasive, the amount of sensitive data stored on them grows exponentially. With mobile applications that include online banking, e-commerce and social media, the amount of personal information that is being channeled through and stored on these devices continues to grow. Yet, for most smartphone users, the only security measures tied to their device is a four digit pin code at best, with some choosing to even forego that small step towards device security.

Finding a way to securitize these devices has become a pressing issues for organizations around the world. In 2013, DARPA released a Request for Proposal (RFP) soliciting biometric solutions for authenticating users utilizing Department of Defense (DoD) Information Technology devices [13]. Other organizations, such as the National Institute of Standards and Technology (NIST) and the National Science Foundation (NSF), have also solicited research proposals in this area [14,15].
II. BACKGROUND & RELATED WORK

A. Keystroke Biometrics

Research relating to keystroke biometrics on traditional hardware keyboards has been increasing in recent years. In 2013, Bakelman et al [2] utilized the Pace Classification System to evaluate both short-string input: passwords and numeric keypad input. The Pace Classifier outperformed 14 other classification systems, with an Equal Error Rate (EER) as low as 8.7% on the password input data, and 6.1% on the numeric input data.

Maxion & Killourhy [3] added the constraint of mandating the use of a single finger (right hand, index) to enter a 10 digit number on a numeric keypad. Utilizing the Random Forest machine learning system, they were able to achieve a correct-detection rate of 99.97%.

Leveraging traditional keystroke biometric features on mobile devices has been researched with increased frequency. Maiorana et al [4] achieved an EER of 13.59% while using traditional keystroke biometric features run through a classification algorithm of their own design.

Additional studies have been done comparing the performance of physical keyboards associated with personal computers, with virtual keyboards associated with mobile phones. In their paper from 2012, Trojahn and Ortmie concluded that keystroke biometric data extracted from the virtual keyboards provided adequate features for authentication.

B. Touch Screen Biometrics

Several researchers have begun incorporating touchscreen features native to mobile devices in order to supplement keystroke biometric data and possibly improve authentication performance. Zheng et al.[6] utilized timing, pressure and size & orientation of the finger used to press – all data easily extracted from today’s smartphones – to achieve an EER of 3.65% on users entering short-string data input. Others, such as Kambourakis et al.[7] supplemented traditional keystroke biometric data with distance covered (in pixels) between successive key presses, which resulted in an EER of 26% on a ten digit passcode as well as an EER of 13.6% on short passphrase input.

C. Additional Biometric Data

There are several other types of biometric data that can be captured on today’s smartphones which we chose to exclude from this study. This biometric data is in no way less valuable or useful than the data we chose to focus on. It simply wasn’t included in this study so that focus could be applied to the biometric data associated with the virtual keyboards and touch screens native to mobile phones.

Feng et al. [17] utilized motion sensors such as the accelerometer to study whether or not the action associated with picking up the phone could provide adequate biometric data for authentication.

A study by Alariki et al. [18] leveraged touch gesture behavior as the biometric data used for authentication.

III. PACE UNIVERSITY BIOMETRIC SYSTEM

A. Android BioKeyboard

For our study, a 10-digit numeric passcode was utilized for data input. This passcode was similar to one used in a previous study [3] and identical to the passcode used in [2, 8]. This passcode (914 193 7761) was selected not only because 914 is the local telephone area code, but also because the sequence of numbers adequately spans the entire range of the keyboard.

Data was collected for this study using 5 identical Android LG-D820 Nexus 5 mobile phones. A virtual keypad was developed for the Android platform and was loaded on each phone [8]. This keypad was then identified as the default keyboard on each of the 5 Android devices. With the new virtual keyboard running on each device, keystrokes are captured on any application being used on the device that utilizes a keyboard. A SQLite database is used to capture key press and key release events. Data can then be transmitted from each of the 5 devices to a server where all of the data is then combined. The data associated with these key press and key release events include the action itself (whether it was a press or a release), the timestamp, the screen coordinates, and the pressure associated with each key press. In order to achieve as much precision as possible, timestamps were captured in milliseconds. Pressure results are typically between 0 and 1, where 1 equates to what could be considered as a normal amount of pressure. Finally, screen coordinate data is calculated in conjunction with the screen resolution of each device.

B. Data Collection

Three different sets of participants were utilized for this study, totaling 52 distinct subjects. The sets of users included:

- City of White Plains, NY Employees
- Pace University (NYC) Students
- Pace University (PLV) Students

On two separate occasions, separated by several weeks, each set of users was tasked with entering the 10 digit string (914 193 7761) at least 30 times. If any of the users felt they had made an error, they were instructed to continue typing, as the system would only register successfully entered strings. After each data collection session, data on each Android device was saved locally to the device, and then additionally transmitted to the centralized server. Fig. 1 provides an overview of how the data is collected and then transmitted to the centralized database.
Upon completion of both rounds of data collection, we had collected almost 59,000 records of data, from 52 distinct participants. Combined, 190 keystroke and Touch Screen features were collected. Of the 52 participants, 44.3% were male (55.7% were female), the average age was 23 (the range of ages was between 17 and 51), and the vast majority were right hand dominant (86%).

With the raw data collected, the next step was to run the data through the Pace University Classifier. In order to thoroughly vet the success at authentication with each feature type, the data would be run through the classifier in three ways:

1. Keystroke features only (Features traditionally associated with hardware keyboards)
2. Touch Screen features only (Features unique to mobile devices)
3. Combined (Keystroke and Touch Screen) features

IV. PACE UNIVERSITY DICHTOMY CLASSIFIER

The Pace University Dichotomy Classifier (henceforth to be referred to as the “Pace Classifier”) is a system that takes a multi-class problem and transforms it into a two class problem [9]. It achieves this via a vector-difference authentication model. The two classes that are then formed (“Within-person” and “between person”) are used to authenticate the users based upon the feature data collected. The “Within-person” class is associated with being authenticated. The “between-person” class is associated with not being authenticated.

V. RESULTS

The feature data associated with our study was run through the Pace Classifier twice with varying results. The first session utilized Repeated Random Subsampling (RSS) as our procedure for validation. This method uses fewer samples than other validation procedures. Thus, it can run quickly and efficiently through datasets with a very large number of records. For the second pass through the Pace Classifier, Leave One Out Cross Validation (LOOCV) was utilized to evaluate the validation performance. This procedure utilizes much more of the data than RSS, and as a result, takes significantly longer to run through the Pace Classifier.

A. Repeated Random Subsampling Results

Each of the three feature files was run through the Pace Classifier using RSS as the validation procedure. The Keystroke Features file, consisting of the features that are associated with physical hardware keyboard, returned an Equal Error Rate (EER) of 23%. This finding was higher than a similar study done on physical hardware keyboard associated with a personal computer (PC)[10].

The feature file containing the data associated with Touch Screens fared better. Running this feature data through the Pace Classifier resulted in an Equal Error Rate (EER) of 13.8%.

Lastly, the combined feature file was run through the Pace Classifier, returning an Equal Error Rate (EER) of 14.9%.

B. Leave One Out Cross Validation (LOOCV)

Running the data through the Pace Classifier using Leave One Out Cross Validation (LOOCV) returned better results. The data set containing the Keystroke feature data returned an EER of 20.2%, slightly better than the 23% returned when using RSS.

The improved performance of LOOCV was more pronounced when running the other two datasets through the Pace Classifier. The Touch Screen feature file returned an EER of 4.9%.

Finally, the combined file consisting of both keystroke and touch screen features returned an EER of 7.1%.

C. Receiver Operating Characteristic (ROC) Curves

Receiver Operating Characteristic (ROC) curves are a commonly used tool used to evaluate the performance of classifiers as well as matching algorithms associated with Machine Learning [11]. ROC curves are often used to illustrate where the False Acceptance Rate (FAR) and False Rejection Rates (FRR) intersect. That point of intersection on the ROC curve is a visual representation of the Equal Error Rate. The ROC Curves for this study are below:
ROC Curve for Keystroke Feature Data – Repeated Random Subsampling (RSS)

ROC Curve for Keystroke Feature Data – Leave One Out Cross Validation (LOOCV)

ROC Curve for Touch Screen Feature Data – Repeated Random Subsampling (RSS)

ROC Curve for Touch Screen Feature Data – Leave One Out Cross Validation (LOOCV)

ROC Curve for Combined Feature Data – Repeated Random Subsampling (RSS)

ROC Curve for Combined Feature Data – Leave One Out Cross Validation
VI. LIMITATIONS

As a result of the data inputs associated with this study, there are several limitations associated with this research which should be addressed.

Firstly, the brevity of the fixed 10 digit string (914 193 7761) limits the ability for the system to thoroughly “learn” the biometric data being returned by each user. Additionally, since the only characters that were captured to be incorporated into this study were associated to the 10 digit string, additional data related with each user’s keystrokes were completely ignored. Since the key logger ignored incorrect strings altogether, as well as backspaging, a significant amount of biometric data was forgone in order to focus on the dedicated 10 digit numeric string.

Secondly, this research was conducted on a singular type of mobile device, the Android LG-D820 Nexus 5 mobile phone. Recent studies have indicated that iOS phones associated with the Apple iPhone mobile device have captured 47.7% of smartphone sales in 2015, with Android devices capturing approximately 47.6% [16]. By not running the experiments associated with this study on the Apple iOS, we cannot assertively state that any results currently extracted via the Android device would be better, worse or comparable to results associated with iOS devices. That said, it is quite clear why the Android platform was selected for this study as opposed to utilizing the Apple iOS. The Open Source approach associated with Google, and Android specifically, lends itself to the type of experimentation and research associated with that of this study.

Lastly, this study was limited to keystroke biometric authentication using the Pace Classifier, using a standard Euclidean distance metric. It is fair to say that results would vary if additional studies were conducted using various other classification algorithms, as well as other distance metrics.

VII. NEXT STEPS

It was our assumption that the combined features files would perform better than their individual source files, regardless of whether or not we utilized RRS or LOOCV validation procedures. It can be clearly seen, however, that the combined file in both instances, performed considerably worse than the Touch Screen feature files. It is our assumption that this is due to keystroke features overwhelming the touchscreen features, resulting in a higher combined EER. Adjusting the distance metric that the Pace Classifier utilized may remedy this. As a result, our next step would be experimenting using different distance metrics, such as Mahalanobis, to see if the performance improves not only on the combined files, but on the individual Keystroke and Touch Screen features files as well.

Additionally, comparing the performance of other classification algorithms with the results we reported using the Pace Classifier, would not only help us identify how well the Pace Classifier performs, but could possibly provide additional insights as to how to modify the Classifier for improved performance associated with future research. Classification algorithms and/or distance metrics that may be leveraged for future research include:

- Manhattan (scaled)
- Nearest Neighbor (Mahalanobis)
- Outlier Count (z-score)
- SVM (one-class)
- Mahalanobis
- Mahalanobis (normed)
- Manhattan (filter)
- Manhattan
- Neural Network (auto-assoc)
- Euclidean
- Euclidean (normed)
- Fuzzy Logic
- K Means
- Neural Network (standard)

VIII. CONCLUSION

The main goal of this paper was to identify whether or not the Pace University classifier could be adequately extended to authenticate data associated with and extracted from mobile devices. Indeed, the research associated with this paper is essentially an extension of previous research studies that leveraged the Pace Classifier [2,9,10]. The EER of 4.9% associated with the Touch Screen features indicates that the classifier works exceedingly well on the new feature sets. However, some modification to the distance metrics utilized by the Pace Classifier may be required in order to improve results associated with the combined feature files. That research is currently underway.

REFERENCES


