INVISIBLE SHIELD: GESTURE-BASED MOBILE AUTHENTICATION

by

Kent W. Nixon

B.S., University of Pittsburgh, 2013

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Kent W. Nixon

It was defended on

March 21, 2016

and approved by

Yiran Chen, PhD, Professor, Department of Electrical and Computer Engineering

Ervin Sejdić, PhD, Assistant Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, PhD, Associate Professor, Department of Bioengineering

Thesis Advisor: Yiran Chen, PhD, Professor, Department of Electrical and Computer Engineering

INVISIBLE SHIELD: GESTURE-BASED MOBILE AUTHENTICATION

Kent W. Nixon, M.S.

University of Pittsburgh, 2016

Intelligent mobile devices have become the focus of the electronics industry in recent years. These devices, e.g., smartphones and internet connected handheld devices, enable quick and efficient access of users to both business and personal data, but also allow the same data to be easily accessed by an intruder if the device is lost or stolen. Existing mobile security solutions attempt to solve this problem by forcing a user to authenticate to a device before being granted access to any data. However, such checks are often easily bypassed or hacked due to their simplistic nature. In this work, we demonstrate *Invisible Shield*, a gesture-based authentication scheme for mobile devices that is far more resilient to attack than existing security solutions and requires neither additional nor visible effort from user perspective. In this work, we design methods that efficiently record and preprocess gesture data. Two classification problems, "one vs. many" and "one vs. all," are then mathematically formulated and examined using the gesture data collected from 20 individuals. Classification algorithms specialized for each case are developed, achieving a classification accuracy as high as 90.7% in the former case, and an equal error rate as low as 7.7% in the latter using real Android systems. Finally, the system resource requirements of different classification algorithms are compared.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

During my time in the University of Pittsburgh's graduate program over the past three years, I have learned a significant amount about my field, about the research process, and about myself. As any research-oriented graduate student will tell you, only a small fraction of this has occurred in the classroom setting, with the rest occurring over the course of many long days and nights spent with my advisor, collaborators, and lab-mates laboring over our next paper submission. As such, I would like to take this opportunity to thank the various members of these respective groups for their support and encouragement over the years. Individual's bearing specific mention and thanks are as follows: My advisor, Dr. Yiran Chen, for first recruiting me into the Evolutionary Intelligence lab, and for his constant encouragement and presentation of exciting opportunities since. My professor and collaborator, Dr. Zhi-Hong Mao, for inspiring me to appreciate the beauty of mathematics and for lending his insight when mine inevitably fell short. Finally, my friend and lab-mate Xiang Chen, for providing invaluable (even if sometimes terse) feedback on my research methods and writing style.

Also, mom.

1.0 INTRODUCTION

Intelligent mobile devices have become the major horsepower behind the evolution of the modern electronics industry: smartphone and tablet shipments triple every six years, and are anticipated to reach a volume of 1.7B in 2017 [18]. Besides communication, smart mobile devices are broadly utilized in many aspects of everyday life, such as web surfing, entertainment viewing, fitness tracking, etc. With 84% of smartphone owners now utilizing their device to store both business and personal data [1], protection of this information has become a major concern. Although a number of solutions for mobile device security and authentication have been proposed, the most popular schemes like PIN and pattern lock remain easily bypassed [2][3].

Mobile phone manufacturers have begun to include biometric identification systems in their products to address this security weakness. As an example, a fingerprint scanner was introduced in both the Apple iPhone 5S and HTC One. However, such biometric authentication methods suffer from two intrinsic drawbacks: First, they often incur extra cost, e.g., bespoke sensors like fingerprint scanners. Second, the authentication procedures of these solutions still require extra time and effort before a user can access their data, no different from conventional methods. This delay continues to be the largest contributor to reduced user adoption rates of mobile security solutions in general [12].

In this work, we propose *Invisible Shield*, a gesture-based mobile authentication scheme which overcomes the drawbacks of existing security solutions. Our scheme is based on data

recorded during regular operation of the device using only existing, internal sensors. Specific contributions of our work to the field include:

- Mathematical definition of two authentication scenarios, "one versus many" and "one versus all," and application of gesture-based classification algorithms for each case;
- Demonstrating that normal device interaction contains sufficient information to be utilized as a highly reliable authentication key, achieving an identification accuracy as high as 90.7% and allowing for an entirely transparent security scheme for mobile devices;
- Identification of the most efficient form of data representation, sample rate, and training set size so as to achieve maximized classification accuracy and minimized system resource requirements;
- Detailed analysis of the utilized classification algorithms in terms of feasibility, accuracy, and demand on mobile device resources.

The remainder of this thesis is organized as follows: Section 2 provides a background on gesture-based mobile security solutions and an overview of our *Invisible Shield* system; Section 3 presents our data collection and processing techniques; Section 4 details the two target mobile security scenarios, and our developed approaches to each; Section 5 illustrates the experimental results; Section 6 provides discussion regarding our experimental results; Section 7 reviews related works; and Section 8 concludes the paper.

2.0 BACKGROUND AND OVERVIEW

Without loss of generality, we define "gesture" as the way in which a user interacts with their device. Because the majority of interaction with modern mobile devices involves interfacing with a touchscreen, we limit our examination to only gestures performed during these interactions.

The human hand contains 29 pieces of bone, 35 pieces of muscle, 48 strips of nerves, and 123 pieces of ligament [19]. Such biometric features cause an individual's gesture to be entirely unique, with this uniqueness further emphasized by varying flexibility of joints and personal comfort preferences [6]. Hence, the unique nature of gesture defines it as an ideal vehicle for user identification. Further, gesture patterns are not only unique but also change over time [15], allowing authentication credentials to evolve with the user.

2.1 PROPOSED SOLUTION

Invisible Shield functions as follows: First, gesture information is captured as a user interacts with their device. The accelerometer, gyroscope, and touchscreen are polled for information while a user is in direct contact with the touchscreen. Once the user is no longer in contact with the touchscreen, the gesture is considered complete and all collected data is stored to a database. We note that the three selected sensors are utilized as they are almost universally available on

modern mobile devices, allowing *Invisible Shield* to theoretically function regardless of the specific device on which it is being used.

The recorded data is standardized and normalized into a uniform format for later comparison. Specifically, standardization transforms gesture data into equally sized data vectors with constant sample rates. As different sensors on a device likely report raw data values with varying magnitudes, normalization is performed to remove this artificial weighting from the recorded features. The set of newly-formatted data then becomes the basis of a user profile utilized to uniquely identify an individual.

As new input gestures are received, they are compared to existing user profiles utilizing machine-learning and pattern-matching techniques, which are selected based on the authentication scenario. Two distinct scenarios are defined: "one versus many" and "one versus all." "One versus many" (OVM) denotes the case that an unknown user is identified based on his/her gestures as one out of a finite group of known users. Alternatively, "one versus all" (OVA) denotes the case that an unknown user is identified either as the owner of a device or as an attacker.

In either case, once an input gesture passes the authentication check, the user is allowed continued access to the device, and the newly recorded gesture is added to the database of gestures for that user. The oldest gesture is then removed from the same database and the user profile is updated, allowing the *Invisible Shield* system to evolve with the user. If the input does not pass an authentication check, it likely originates from an attacker, so the device is locked to protect the user’s data.

An overview of *Invisible Shield* is shown in Figure 1.

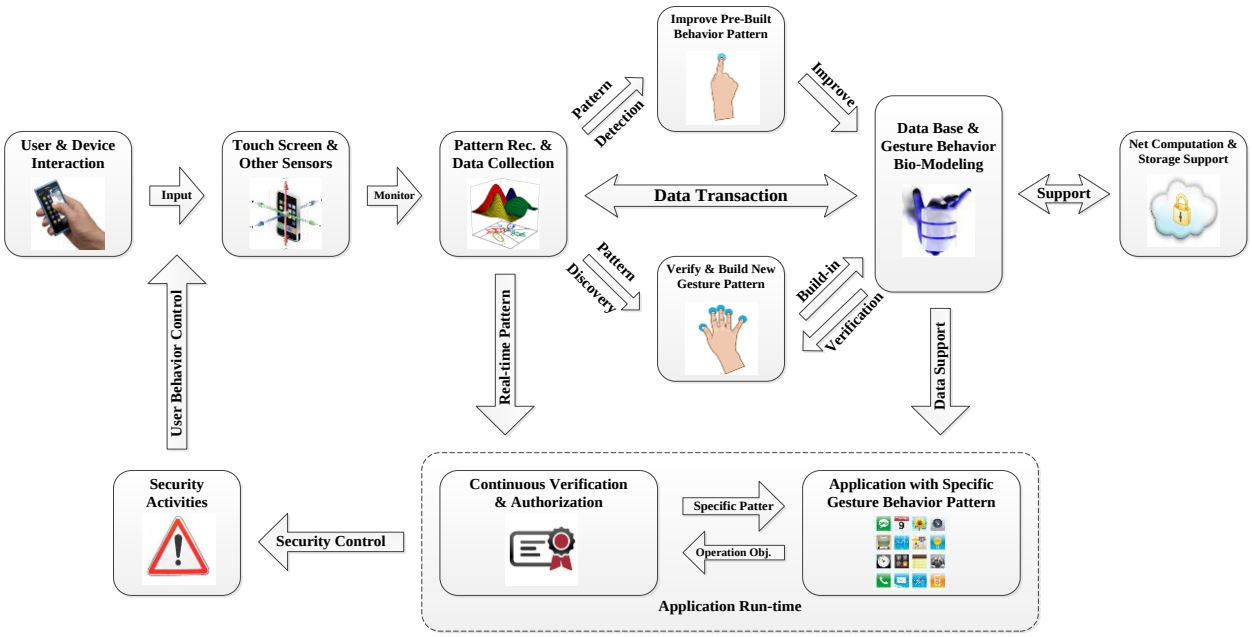


Figure 1. Data flow of *Invisible Shield* system

2.2 CHALLENGES

The first challenge in *Invisible Shield* system design is the identification of features which represent a user's gesture quantitatively. Our design utilizes sensors already embedded on the device to record information related to gesture while a user interfaces with the device. In addition to being available on most mobile devices, the specific sensors we utilize are selected as they provide data closely associated with the way a user is currently interacting with the phone or tablet. Through the classification accuracy of our algorithms, we show that the data returned from these sensors sufficiently defines a user's unique gesture.

The second challenge is to convert the raw data recorded from device sensors into a standardized format such that it can be used in machine-learning algorithms. In our data collection phase, gestures are recorded which occur over a period ranging from 0.3 to 2.2 seconds in length, with a timing distribution as seen in Figure 2. The varying duration of the gestures, combined with the fluctuating rate at which the Android OS returns data from sensors, results in gestures containing differing amounts of raw data. In our work, we standardize and normalize gesture data before attempting to utilize it for classification. We consider two standardization and two normalization methods, and determine which yields the highest consistency for user identification while maintaining differentiability between users.

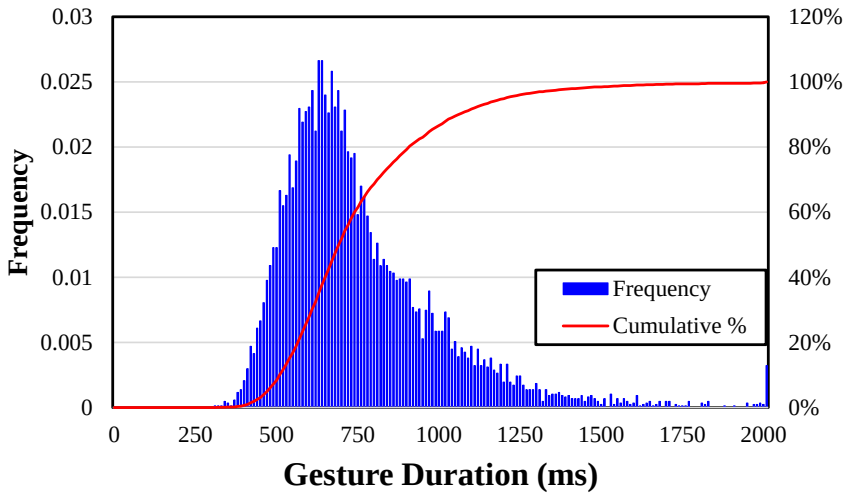


Figure 2. Timing distribution for recorded gestures

The third challenge is to determine which algorithms function best to correctly classify users based on their gestures. As the targeted classification problem is partitioned into OVM and OVA, we are able to focus our examination on algorithms specifically suited to each situation. In Section 4, we discuss the various approaches we attempt.

Finally, the remaining challenge is to design an authentication system that will evolve with the user. It has been suggested in other works that user gestures change over time [15]. Utilizing the collected data, examine methods to account for this. Based on a rolling database of training data, we are able to effectively compensate for the shift of the gesture data over time.

Besides the above theoretical challenges, feasibility requirements need also be fulfilled. In a real world scenario, training time required by *Invisible Shield* must be minimized to encourage users to adopt it. However, most pattern-recognition algorithms achieve higher classification accuracy with a larger training set. In order to find the most efficient solution to this tradeoff, accuracy is investigated utilizing a range of previous inputs as training data.

Furthermore, the sampling resolution required to achieve optimal classification accuracy must be evaluated. Sampling from sensors at higher rates enhances observation granularity and will likely improve classification accuracy. However, this requires additional power and greater utilization of system resources during the recording stage as well as an increase in processing time in the classification stage. We analyze recorded gesture data at different sampling rates via down sampling from an original raw recording in order to evaluate the impact on classification accuracy.

3.0 DATA COLLECTION AND ANALYSIS

This section describes the data collection methods and sample group utilized as part of *Invisible Shield*. Also discussed is the mathematical description of gesture data referenced throughout the rest of the paper, as well as the various techniques used to standardize and normalize raw data.

3.1 DATA COLLECTION

To collect gesture data suitable for analysis with the *Invisible Shield* system, a gesture recording application is developed based on the Android lock screen. In it, a 3×3 grid of dots is displayed to the user, who is then tasked with connecting the correct dots in the correct order to form an unlock pattern. In our experiments, the unlock pattern is set as a "fish" shape – a gesture which utilizes the top six dots in the 3×3 grid, as displayed in Figure 3. This particular recording method and pattern are selected as they are easy for users to understand and perform. All subjects are asked to draw the same "fish" in order to guarantee that any differences between users will result from the way the gesture is performed and not the gesture itself.

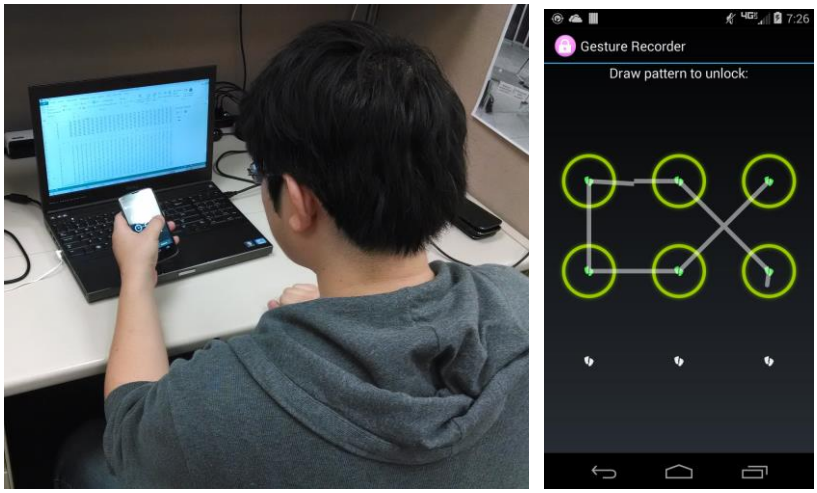
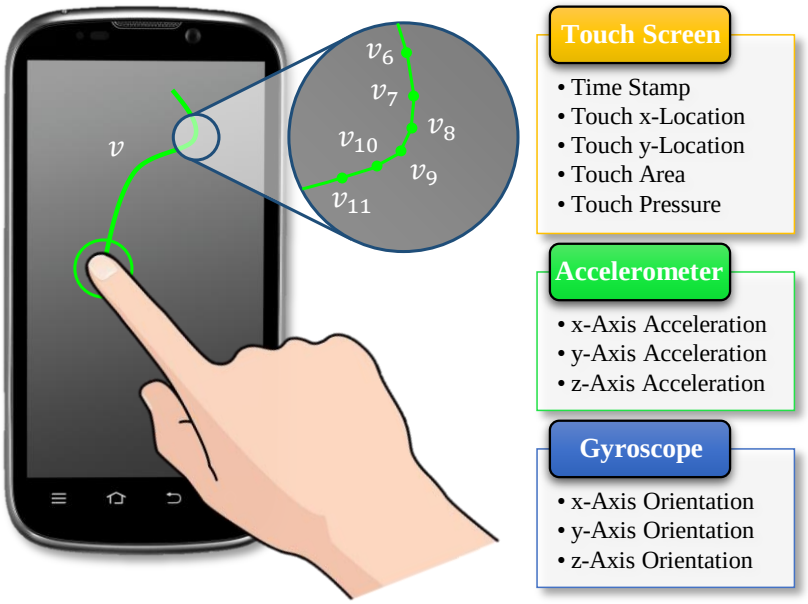


Figure 3. User and recording app displaying fish pattern

Our gesture recording application is deployed onto several LG Nexus 4 devices running the Android 4.3 operating system. The application samples and records data from the device touchscreen, accelerometer, and gyroscope while the user is in direct contact with the touchscreen. During each gesture, data is first sampled from sensors at the maximum possible rate allowed by the device. At each sample point a total of 11 values are recorded, a summary of which is shown in Figure 4, with sample values from the sensors displayed in Table I.



(a) (b)

Figure 4. (a) A user gesture is composed of many discrete samples (b) Each sample contains F=11 raw features

Table 1. Sample Feature Data

Parameters	Mean	Variance
Time Stamp	82.10	7.01
Touch x-Location	631.00	22.80
Touch y-Location	158.00	23.00
Touch Area	0.29	0.03
Touch Pressure	0.71	0.03
x-Axis Acceleration	2.52	1.01
y-Axis Acceleration	8.29	0.57
z-Axis Acceleration	5.48	0.68
x-Axis Orientation	0.05	0.17
y-Axis Orientation	-0.04	0.17
z-Axis Orientation	0.13	0.10

Gesture samples are collected from 20 subjects between the ages of 21 and 26, including 7 female users and 13 male users. Each subject is given a device preloaded with the gesture recording application for a 1-2 week period. The gesture recording application reminds the subject to interact with the device once an hour by vibrating and displaying an onscreen notification. At that time, the user repeats the "fish" gesture on the device ~15 times, resulting in a total of between 178 and 2036 recorded gestures per user. By collecting samples gradually over a long period of time, variance induced by muscle exhaustion and boredom are minimized. In the case where a user enters the wrong pattern into the device, e.g. missing one of the dots or connecting them in the wrong order, the data from that gesture is not recorded in the gesture database.

3.2 STANDARDIZATION

The Android OS does not permit a constant data sampling rate from attached sensors. Instead, sensor data is supplied at a coarsely-grained, system relative rate, i.e. slow, medium, fast. The recording application requests data at the highest rate possible, resulting in an average (but not constant) sample rate of approximately 200Hz. The inconsistencies in sampling rate, in addition to varying gesture durations, cause each data vector to contain differing amounts of sample points.

In order to reliably compare user gestures, all samples are reformatted to a uniform structure, or standardized. In cases where sample duration is variable, a common method of data standardization defines a maximum time window within which each time series is expected to

fit. This method is further extended to extrapolate sample points from the raw data which occur at regular intervals within the time window, approximating a constant sample rate. We refer to this process as "temporal" standardization.

Mathematically, each recorded gesture is defined as originating from one out of *N* users. For user n, data vector $x_n^{(i)}$ represents a single gesture where $i \in \{1, \dots, r_n\}$ and r_n is the total number of gesture samples recorded for that user. Within vector $x_n^{(i)}$ there are $s_{n,i}$ subvectors denoted as $x_{n_j}^{(i)}$, each representing a single sample point where $j \in \{1, \dots, s_{n,i}\}$. Each $x_{n_j}^{(i)}$ contains $F = 11$ raw features, addressed individually as $x_{n_{j,f}}^{(i)}$ with $f \in \{1, \dots, F\}$. The dimensionality of each $x_n^{(i)}$ is then equal to $s_{n,i} * F$. Table II contains a summary of these variables and definitions.

Temporally standardized data vector $y_n^{(i)}$ and its subvectors $y_{n_m}^{(i)}$ are defined using

$$y_{n_m}^{(i)} = \frac{t_{x,n_v}^{(i)} - t_{y,n_m}^{(i)}}{t_{x,n_v}^{(i)} - t_{x,n_u}^{(i)}} x_{n_u}^{(i)} + \frac{t_{y,n_m}^{(i)} - t_{x,n_u}^{(i)}}{t_{x,n_v}^{(i)} - t_{x,n_u}^{(i)}} x_{n_v}^{(i)}. \quad (1)$$

Here $m \in \{1, \dots, q\}$, where q is the total number of temporally equidistant samples in the standardized data vector. q is determined as

$$q = lh + 1, \quad (2)$$

where l is duration of the standardized time window in seconds and h is the desired sample rate within the window. Each sample point $y_{n_m}^{(i)}$ in the standardized data set is taken to occur at time $t_{y,n_m}^{(i)}$, which is calculated by

$$t_{y,n_m}^{(i)} = (m - 1) * h. \quad (3)$$

Raw data samples $x_{n_u}^{(i)}$ and $x_{n_v}^{(i)}$ are the recorded samples that occur at time $t_{x,n_u}^{(i)}$ and $t_{x,n_v}^{(i)}$, immediately before and after time $t_{y,n_m}^{(i)}$, respectively. For gestures that do not last the full duration l there may not exist $t_{x,n_v}^{(i)} \geq t_{n,y_m}^{(i)}$ for every m; in such cases, $y_{n_m}^{(i)} = 0$. In our work, we

examine data standardized with $h \in \{1, \ldots, 25\}$ and $l = 2$, which allows a time window long enough to entirely contain almost every recorded gesture.

While temporal standardization is the traditional method of handling data time series, very few of the recorded input gestures exist for the full duration l. The result of this is that standardized data vectors contain a large number of trailing zeroes which likely contribute little to classification accuracy. This has a further effect of causing the machine-learning algorithms explored in Section 4 to utilize valuable system resources processing data that is insignificant.

In response to this perceived inefficiency, we propose "spatial" standardization to process the data in our design. Spatially standardized data vector $y_n^{(i)}$ and its sample points $y_{n_m}^{(i)}$ are calculated using Eq. (1), but with q redefined as

$$q = p + 1, \tag{4}$$

where p is the number of temporally equidistant samples of gesture $x_n^{(i)}$ that occur between $t_{x,n_1}^{(i)}$ and $\mathrm{t}_{x,n_{s_{n,i}}}^{(i)}$ $\mathrm{t}_{x,n_{s_{n,i}}}^{(i)}$. While every temporally standardized $y_n^{(i)}$ shares the sample rate h, each spatially standardized $y_n^{(i)}$ has its own unique sample rate $h_{n,i}$, defined based on

$$h_{n,i} = \frac{p}{\mathrm{t}_{x,n_{s_{n,i}}}^{(i)}}. \tag{5}$$

By having a unique sample rate for each gesture, it is guaranteed that each standardized data vector is fully utilized to store unique information from the original gesture, eliminating trailing zeroes.

Classification accuracy when using both temporal and spatial standardization is investigated in Section 5.

3.3 NORMALIZATION

Individual feature values reported by a device's sensors have different magnitudes. In order to avoid individual features being artificially weighted by their size relative to other components, the magnitude of each feature score must be normalized. Table I contains measurement values taken from a random set of sample points recorded using an LG Nexus 4 smartphone as an example of the raw values which are returned from device sensors. Although a normalization method can be easily determined for this specific device, the goal of *Invisible Shield* is to function on all existing devices, regardless of specific hardware components or returned measurement values. Hence, a normalization technique that functions regardless of the actual values returned by sensors is developed.

One possible normalization strategy is to adjust feature values such that each feature is restricted to the range 0-1 [20]. Each feature $z_{n_{m,f}}^{(i)}$ of normalized vector $z_n^{(i)}$ is calculated from standardized vector $y_n^{(i)}$ using the G previous inputs from user n via

$$z_{n_{m,f}}^{(i)} = \frac{y_{n_{m,f}}^{(i)} - \min_{a\in\{1,\dots,G\}}\left\{y_{n_{m,f}}^{(a)}\right\}}{\max_{a\in\{1,\dots,G\}}\left\{y_{n_{m,f}}^{(a)}\right\} - \min_{a\in\{1,\dots,G\}}\left\{y_{n_{m,f}}^{(a)}\right\}}. \quad (6)$$

This method is referred to as "naïve" normalization, as it has the easily recognizable drawback that extreme outliers can have disproportionate effect on normalized feature scores. In order to compensate for this, a z-score normalization method [21] is also investigated as a potentially superior method, and the impact of both normalization methods on classification accuracy is examined in Section 5.

Table 2. Symbols definition and description

Symbol	Description
N	Total number of users
F	Total features per sample
x_n	Raw gesture data from user n
$x_n^{(i)}$	The i^{th} gesture in data set x_n
$x_{n_j}^{(i)}$	The j^{th} sample point in gesture $x_n^{(i)}$
$x_{n_{j,f}}^{(i)}$	The f^{th} feature of sample point $x_{n_j}^{(i)}$
r_n	Total number of recorded gestures for user n
$s_{n,i}$	Total sample points in gesture $x_n{}^{(i)}$
y_n	Standardized gesture data from user n
z_n	Normalized gesture data from user n
q	Number samples in standardized data
l	Number seconds input gesture may last
h	Sample rate of standardized data
p	Number of gesture partitions
$t_{x,n_u}^{(i)}$	Time of u^{th} sample in $x_n^{(i)}$
G	Number of training gestures

4.0 AUTHENTICATION ALGORITHMS

In this section, the authentication algorithms developed for the targeted applications "one versus many" (OVM) and "one versus all" (OVA) are described in detail.

4.1 ONE VERSUS MANY (OVM)

In the case of OVM, an algorithm must identify which user is interfacing with the device out of a set of known users. An example of such a situation would be a family tablet that is able to adjust its interface to suit the interests of the family member currently using it. Two types of classification algorithms, *k-nearest neighbor* and generative models, are examined and applied to this scenario.

4.1.1 K-Nearest Neighbor

k-Nearest Neighbor (*k*NN) algorithms are commonly utilized when attempting to classify input data based on one of N known classes [17]. A *k*NN classification algorithm is desirable in that it is accurate and computationally lightweight when the number of users and gestures are small.

The algorithm functions by maintaining a database of G previous inputs for each user n. When a new input φ_w is received from user w, it is compared to all previous inputs in the

database using a selected distance metric. Input φ_w is then determined as originating from user c_φ such that the majority of its *k*-nearest neighbors also originate from user c_φ. When $c_\varphi = w$, classification is considered correct.

Classification accuracy is investigated for all $G \in \{1, \dots, 100\}$, with the upper limit of G being selected as it represents a realistic value for the number of gestures a user can be expected to perform in a single day [4]. The value of k remains constant at $k = 3$ for all classification attempts, as it has been shown that the specific value of k exerts little influence over classification accuracy [17]. Lastly, we select a Euclidian distance metric to compare gestures.

4.1.2 Generative Model

Generative models function by first creating a simplified mathematical representation of each user. Each representation is constructed by maximizing the probability that the model will generate the data already recorded from that user. When a new input is received, the probability of it being generated by each user's model is determined. The input is then classified as belonging to the user whose model has the highest probability of generating it.

In this work, we assume that all recorded features from a gesture exhibit Gaussian distributions, leading us to select a multivariate Gaussian generative model as our basis for each user. The parameters of this model include the class mean μ_n and class covariance Σ_n, which can be estimated using the sample mean and sample covariance of the G previous inputs (training data) from each user. The originator of input φ_w is determined to be user c_φ where the model for user c_φ has the highest calculated probability of generating φ_w using

$$c_\varphi = \underset{n=1,\dots,N}{\arg\max}\{\log(P(\varphi_w|\mu_n, \Sigma_n))\}. \quad (7)$$

Of note is that the algorithm utilizes the logarithm of the probability density function (PDF) instead of the PDF itself. The reason for this is that in cases of high-dimensional data (high sample rate), the magnitude of calculated probabilities becomes too small to represent reliably using double-precision floating point values. By instead utilizing the logarithm of the PDF, generated probabilities exist entirely within a representable range.

A multivariate Gaussian distribution model is potentially beneficial as it is able to weight features individually and detect relationships between features through the use of the sample covariance, Σ_n. However, generative models place a heavy reliance on having access to an abundance of training data, as the additional information allows for the closer approximation of class models. In addition to this, Σ_n is required to be nonsingular for the PDF to be defined, in turn requiring G to be greater than the dimensionality of $y_n^{(i)}$. At higher sampling rates, gesture data vectors contain over 1,000 features, which translate to more than 10 days' worth of gestures from the average user. By our definitions, this requirement causes *Invisible Shield* to become unlikely feasible in terms of training time.

To overcome this drawback, a dimensionality reduction algorithm is utilized to project recorded data vectors onto a $G - 1$ dimensional subspace. Each user's generative model is then constructed within this subspace, whose reduced dimensionality greatly increases the likelihood that Σ_n will be full rank even for small values of G. Principal component analysis (PCA) [16] is utilized as it creates a low-dimensional subspace for data while maintaining as much of the original sources of variance as possible.

While this allows the generative model to function feasibly, there remains a reliance on large values of G, as PCA introduces a loss of characteristic variance during dimensionality reduction, even though it is minimized.

4.2 ONE VERSUS ALL (OVA)

In the case of OVM authentication, an unknown user is given a binary classification as either the one known owner of a device or an attacker. Because previous input gestures are only available from the device owner, threshold-based classification can be utilized to determine the originator of any inputs. The accuracy of generative models is evaluated when applied to the OVA problem.

4.2.1 Gaussian Generative Models

The techniques utilized in Section 4.1.2 can be easily modified to work in a one versus all case. However, instead of having a separate model for each user in a group of users, only the model for the actual owner of the device is maintained. When a new input gesture is recorded, the probability of it being generated by the user is calculated and compared to a preselected threshold value. If the probability is greater than the threshold value, the originator of the gesture is assumed to be the user and they are allowed continued access to the device. If the probability is below the threshold, the user is assumed to be an attacker and they are denied access to the device.

Varying the threshold has a direct effect on classification accuracy for the algorithm. Decreasing the threshold reduces the number of attackers who are incorrectly given access to the device, which is termed the false positive rate (FPR). By forcing all input gestures to be closer to the user ideal before granting access to the device, FPR is reduced. However, gestures produced by the actual owner of the device are not perfectly ideal. This means that in some cases, the user may be incorrectly denied access to their own device – the proportion of the time this occurs is

referred to as the false negative rate (FNR). FNR can be reduced by increasing the classification threshold, but this in turn increases FPR. In order to simplify accuracy analysis, we consider equal error rate (EER) as the measure of classification accuracy. EER is the value of the FPR and FNR when the threshold has been tuned such that they are equal.

5.0 RESULTS

In this section, the methods of estimating classification accuracy of *Invisible Shield* are described in detail. Also examined are the experimental results obtained using the classification algorithms detailed in Section 4.

5.1 OVM ANALYSIS

To demonstrate classification accuracy in the OVM scenario, the final 70 gestures recorded from each of the 20 subjects (1400 gestures in total) are classified. By using the final gestures recorded by each user, large variances from users altering their interactive preferences, i.e. orientation or method of gripping the device, were eliminated. By the time users recorded their last gestures, they were likely comfortable using the recording device and were no longer radically altering the way they interfaced with it.

We consider training sets of $G \in \{2, \dots, 100\}$ gestures, requiring a total of the last 170 unique gestures from each individual. This is determined by the classification of the 70th-to-last gesture recorded from a subject, which requires at most the 100 gestures occurring before it as training data.

Although many subjects recorded significantly more than 170 gestures, three users recorded less than 200. In the context of this work, it was deemed more valuable to have sample sets from a larger group of user than it was to have larger sample sets from a smaller group of users. As such, the limit of 170 gestures per user was determined in order to be able to equally represent each user in the classification stage, as well as allow at least a 30 gesture buffer during which all users were becoming comfortable with the device. Extra sample data from other subjects is discarded as the additional gestures would result in uneven representation in testing data, influencing classification accuracy.

When attempting to classify a gesture φ_w, training data is considered the final G recorded gestures for users $n \neq w$. However, for user $n = w$, training data is instead the G gestures that occur immediately before the current gesture. This approximates our defined use case where the gesture profiles for all other users of a device are already known and static, while the current user's gesture profile is continuing to evolve as they interact with the phone or tablet.

The accuracy results from the *k*NN classification scheme can be seen in Figure 5. The 3-dimensional surface maps illustrate the classification accuracy for each combination of $G \in \{2, \ldots, 100\}$ and $q \in \{2, \ldots, 50\}$. The top row of results is generated using spatially standardized data, and the bottom row from temporally standardized data. Each column in the figure represents a different method of data normalization, with the first column containing results from non-normalized data, the second from naïve normalized data, and the third from z-score normalized data.

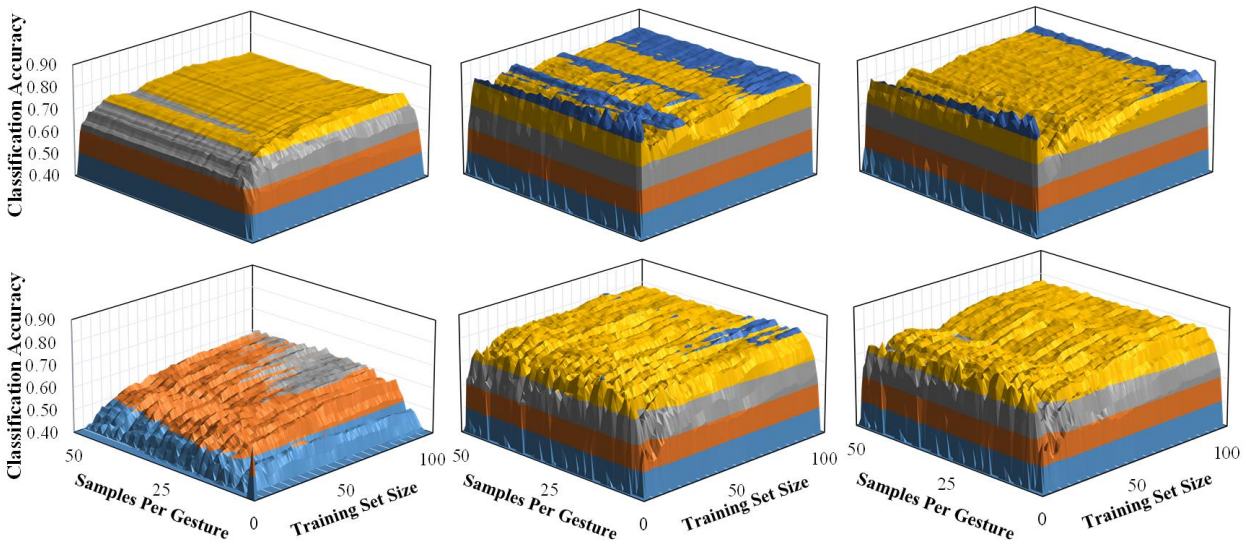


Figure 5. Accuracy results using *k*NN classifier. From left to right: no normalization, naïve normalization, and z-score normalization. Top row is spatially standardized, bottom row is temporally standardized.

The accuracy results indicate that spatially standardized user gestures allow for *k*NN classification algorithms to achieve higher classification accuracy than with temporally standardized data when utilizing the same number of the sample points. Further, naïvely normalizing feature data before attempting to classify gestures yields higher identification accuracy than that achieved using z-score normalized data sets.

Figure 6 illustrates the influence of sample rate over the average classification accuracy for the *k*NN algorithm. It can be seen from the figure that increasing sample rate beyond 10Hz does not increase classification accuracy. In the normalized cases, average classification accuracy ceases to increase after reaching a sample rate of just 4Hz. This would indicate that the characteristic features embedded in user gestures likely occur in the 2Hz range, and is consistent with the findings in [22]. This is significant as capturing this data is possible even at extremely

low sample rates, meaning *Invisible Shield* can function optimally with a *k*NN classification algorithm while imposing very little load on mobile device resources.

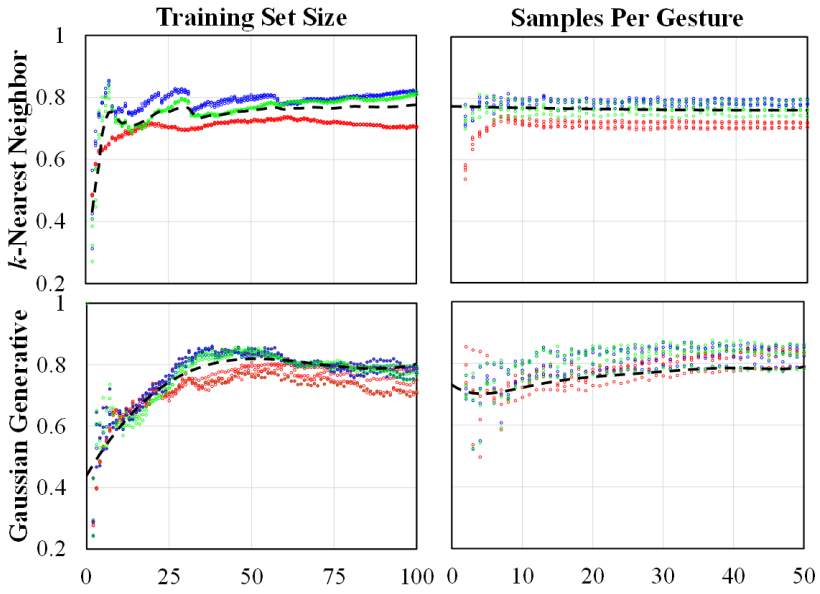


Figure 6. Impact of training set size and sample rate on classification accuracy.

The *k*NN algorithm achieves a peak classification accuracy of 90.7% with spatially standardized data, naïve normalization, $G = 7$, and $q = 6$.

For Gaussian generative models, the process for selecting training and testing data is identical to that of the *k*NN algorithm. The classification accuracy of the multivariate Gaussian generative model is included in Figure 7. The surface maps in the figure are positioned identically to that of the *k*NN results in Figure 5, with each combination of standardization, normalization, training set size, and sampling rate being represented.

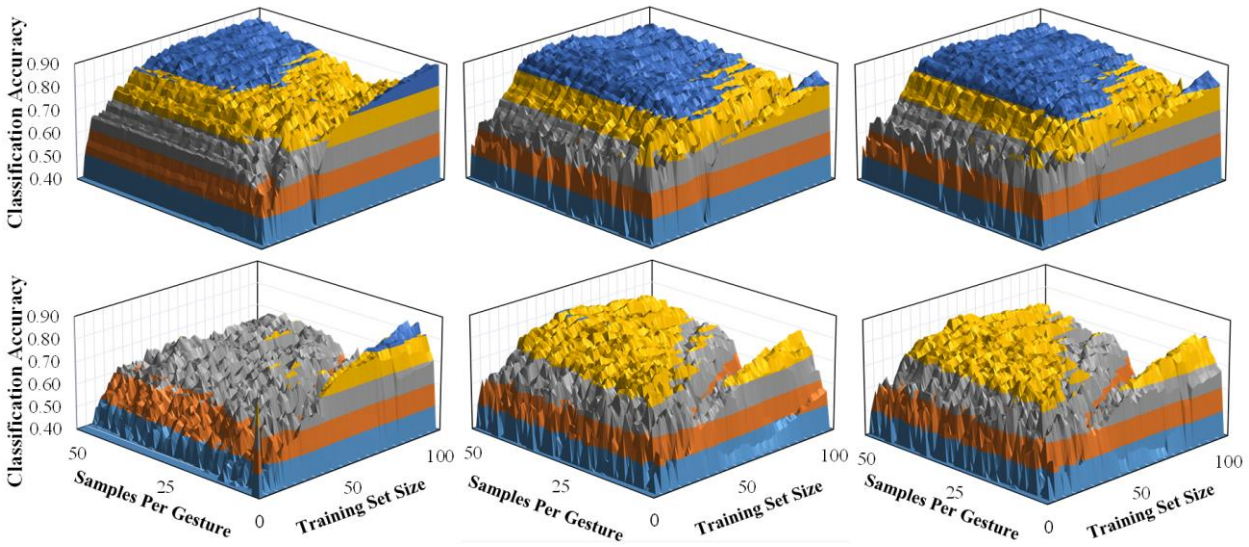


Figure 7. Accuracy results using a multivariate Gaussian classifier. The top row is spatially standardized, the bottom row is temporally standardized. From left to right: no normalization, naïve normalization, and z-score normalization.

The results are to be expected after reviewing the classification accuracy of the *k*NN algorithm. Spatially standardized data is again favored, and naïve normalized data gives the highest accuracy results on average. Figure 6 illustrates the impact of training set size and sample rate on average classification accuracy. The data indicates that a training set of $G \cong 40$ gestures is the best on average. Unlike *k*NN, generative model accuracy continues to rise as sample frequency increases.

Of note, however, is that for every combination of standardization and normalization strategies, there exists an extreme peak in classification accuracy which occurs at a sample rate of approximately 4Hz. This is reflected in the results, with the generative model's maximum classification accuracy of 89% occurring with $G = 99$ and $q = 2$ when applying spatial standardization and naïve normalization. The peak is followed by a rapid decline in classification

accuracy as the sample rate is increased beyond 4Hz, only returning to a comparable level after the sample rate has increased much further.

While this "early peak" phenomenon is present in when utilizing a *k*NN algorithm, it is much more obvious here. This is likely brought about through the use of PCA, which indicates that there exists a large portion of the internal variance for a user which occurs at a frequency just above 2Hz. Unlike the unique, characteristic data that was previously identified as occurring near 2Hz, these higher frequency variations are almost identical for every user, causing a significant drop in classification accuracy once sample rate in increased to the point where they are detected and utilized. The results indicate that this identical information only occurs within a small frequency band, as when higher frequency variances are captured with faster sampling rates the classification accuracy slowly returns to its previous level.

5.2 OVA ANALYSIS

The accuracy of our OVA classification algorithm is evaluated by again performing classification for each of the N subjects utilizing the final 70 gestures from each to provide φ_w. We refer to φ_w as the target gesture.

A multivariate Gaussian generative model is created for the user based on the G gestures that were recorded immediately prior to the target gesture. From this model, the logarithm of the probability of the user generating φ_w is calculated. Also calculated is the probability of generating the 10 first input gestures from users $n \neq w$, which we refer to as imposter gestures.

The first 10 gestures from users $n \neq w$ are utilized as imposter gestures as when they were created, subjects were likely not yet familiar with using the gesture recording device. Similarly, an attacker who has just taken a user's phone or tablet would not be immediately comfortable using it, causing initial imposter gesture patterns to be highly variant in nature. By using the first gestures created by each user, we are able to closely approximate the real-world scenario of an attacker who has just taken the device.

Figure 8 displays the EER results when using the multivariate Gaussian generative model in OVA authentication. Surprisingly, in this case, EER is lowest when no data normalization is performed, while our spatial standardization technique continues to yield the best results.

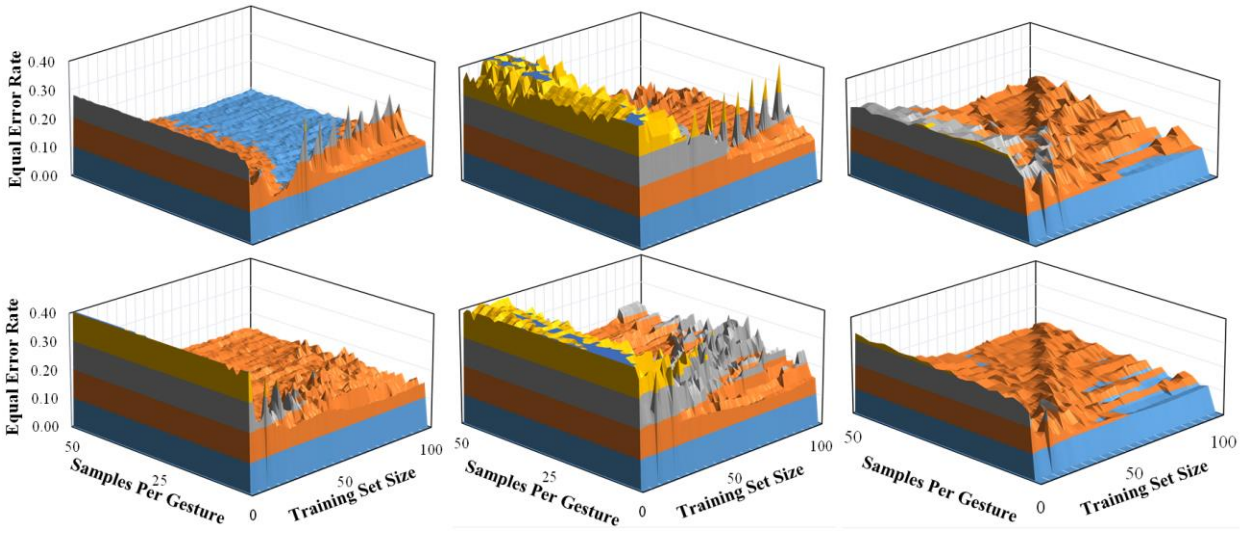


Figure 8. Equal error rates utilizing a multivariate Gaussian classifier. The top row is spatially standardized, the bottom row is temporally standardized. From left to right: no normalization, naïve normalization, and z-score normalization.

5.3 RESOURCE REQUIREMENTS

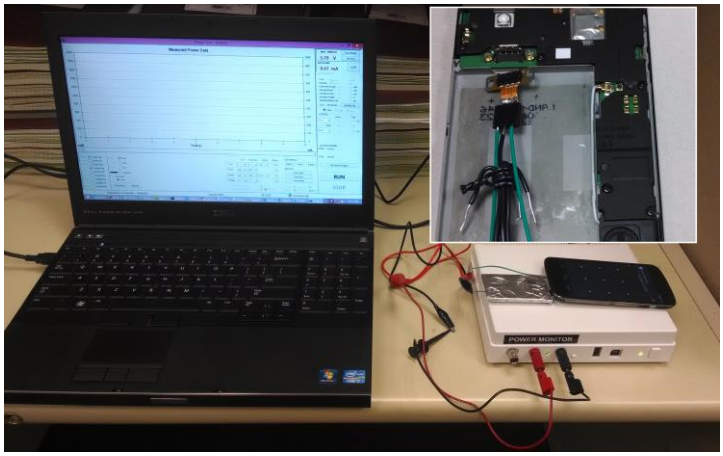


Figure 9. Power measurement setup.

Runtime and power benchmarking is performed on the LG Nexus 4 devices while running the algorithms utilized in *Invisible Shield* in order to determine real world resource utilization. Figure 9 shows our benchmarking setup. In order to be certain that all power utilized by the device is being recorded, we remove the back cover and battery from the phone and modify the internal connector to draw power directly from a Monsoon Power Monitor. Figure 10 contains sample points collected using this test bench which show the power consumption of the device while performing the different calculations required by *Invisible Shield*. In the figure, the blue sample points are taken from the device while idling with the screen turned on displaying our gesture recording app. These measurements serve as a baseline by which to determine the additional power that is required by *Invisible Shield*. The green sample points are recorded while the additional sensors required by *Invisible Shield* (the accelerometer and gyroscope) are enabled and reporting at the highest sample rate, similar to the case when a user is interacting with the

device. Finally, the red sample points are recorded while the additional sensors are enabled and the device is performing calculations as part of our machine-learning algorithms, e.g. matrix multiplication, PCA, etc. The trend lines in the figure detail the average power consumption in each case, and highlight the extra power overhead required by *Invisible Shield* in each of its functional stages.

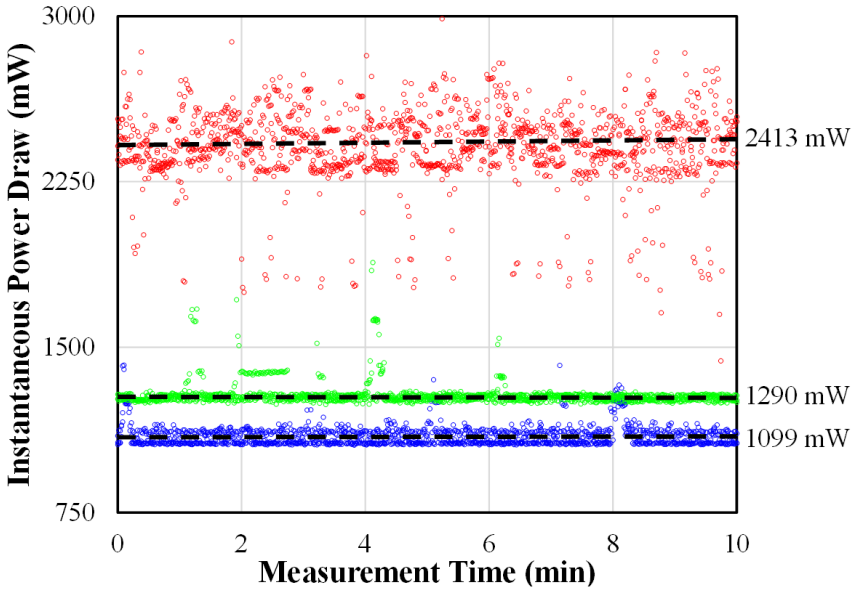


Figure 10. Recorded power levels for LG Nexus 4 device.

Additionally, runtime benchmarking is performed using additional LG Nexus 4 phones. Android OS system timestamps and a benchmarking application are utilized to determine total processing time required to perform user authentication utilizing the two main machine-learning algorithms considered as part of *Invisible Shield*. Runtime benchmarking is performed for each combination of G and q, and is combined with the results from power benchmarking to determine impact on device battery life and usability.

Utilizing the parameters which achieve the highest classification accuracy with the *k*NN algorithm, a runtime of 46ms is required to determine which user a newly-recorded input gesture originates from. Device usability is not heavily impacted by this required processing time, even if performed after every input gesture. Further, modern mobile devices have multiple processor cores which would allow the task to be offloaded onto another core in the background while to user continues to utilize the device. Because of this capability, the main concern in modern systems instead becomes that of power consumption. In the case of the Nexus 4, assuming a user performs 1000 gestures on their phone everyday (100 unlocks followed by nine further gestures each time), *Invisible Shield* requires only 0.4% of the total battery power per day when utilizing a *k*NN classification algorithm.

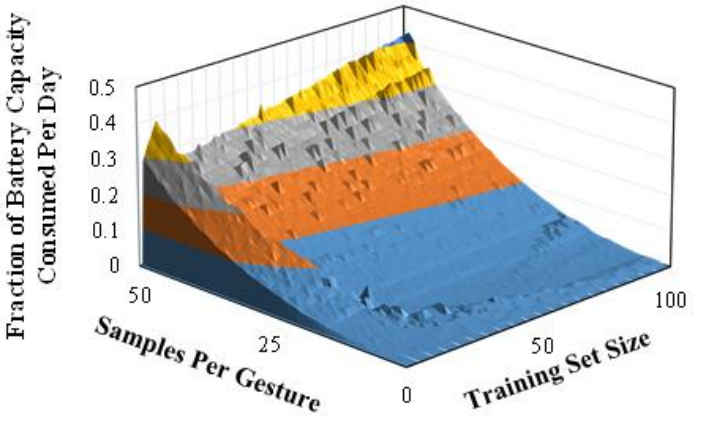


Figure 11. Power consumption of *Invisible Shield* over a day of use.

With Gaussian generative models, the processing time is significantly longer at 181ms to achieve best case classification accuracy, which would likely be noticed by the user and would utilize 1.6% of the device's battery life per day. However, this figure assumes that a new user

model is generated after every new gesture, requiring PCA to be performed each time. When the classification and model generation stages are decomposed, it is found that just performing PCA requires 128ms of computation, while classification only requires 53ms, which itself is on the same order as the *k*NN classification algorithm. The results in Figure 11 show that the system power requirements for this scheme become unmanageable as sampling rate increases beyond 25 times per gesture. We note that this result assumes that a new user model is generated after every new gesture, requiring PCA to be performed each time. When the classification and model generation stages are decomposed, we found that just performing PCA requires 128ms of computation, while classification only requires 53ms. Hence, by reducing the frequency of model generation, power requirements could be reduced even further.

6.0 DISCUSSION

Contrary to our initial hypothesis, our results showed that naïve normalization yields the most accurate results. This indicates that although outliers do play a role in gesture classification, it is a positive one. Happily, naïve normalization has the added benefit of requiring fewer calculations than z-score normalization, although the difference in real-world scenarios is likely negligible.

In all cases, our spatial standardization scheme yielded better results than traditional temporal standardization, given than both contained the same number of sample points. However, note that spatially standardized sample points each contain an additional feature than those in temporally standardized data. This is because unlike in temporally standardized data, each sample point must store timing information as sample rate is not constant between gestures. This causes each temporally standardized $y_{n_m}^{(i)}$ to be of dimensionality $F-1$, while each spatially standardized $y_{n_m}^{(i)}$ is of dimensionality F. The result is that spatially standardized gestures require 10% more memory to store than temporally standardized gestures, and 21% more calculations during matrix operations. However, accuracy results can be realized which are similar to those achieved using temporally standardized data while using approximately half of the sample points, greatly reducing the required computations overall.

When using both *k*NN and generative models for OVA classification, it is noted that a peak in classification accuracy occurs when a sampling rate of 4Hz is utilized, itself allowing for the monitoring of 2Hz signals. The peak quickly drops off as the sampling rate increases to include sources of variance at higher frequencies. This finding is significant as it would suggest that classification accuracy can be greatly improved if gesture data were to be subjected to a low-pass or band-stop filter before being analyzed with *Invisible Shield*. This would allow for the removal of higher frequency sources of variance that all users share.

While *k*NN classification performs extremely well for such a lightweight algorithm, classification accuracy was likely limited due to the inability to weight feature values. By adopting a distance metric which is able to weight feature values, classification accuracy would likely be greatly improved. We pursued this possibility through the use of a large margin nearest neighbor (LMNN) algorithm, and were able to achieve classification accuracy as high as a 98.5% when utilizing the same dataset examined in this work. However, the computational complexity of generating parameters used by LMNN was such that it quickly became obvious that it was infeasible to implement on mobile devices. Hence, it is not formally examined here.

Finally, in the case of both main algorithms it was shown that the required computation time and draw on system resources is negligible when classifying new inputs, meaning *Invisible Shield* can feasibly run in the background on a modern device. However, for multivariate Gaussian generative models, it was discovered that it was not realistic to regenerate a user model after every new input due to the computational complexity of performing PCA. This can be negated through the use of a more computationally efficient dimensionality reduction algorithm. However, the problem opens for consideration the question of how often a user model should be

recalculated in order to maintain classification accuracy. In this work, we assume after every gesture. However, as user gestures change slowly over time, this may not be the case.

7.0 RELATED WORK

This section summarizes the previous works of others in this area or the topics related to it, and how our work can be differentiated from them.

7.1 DEVICE INTERACTION

Previous research has shown a strong link between hand biometrics and mobile device usage, with thumb link length and an individual's degree of joint motion being the two largest contributors to input variation [8]. Similar studies have found that the size and shape of a mobile device also exerts a large influence over the way a user interacts with it, indicating that the same user would create different input patterns between two different devices [5]. These observations are leveraged in [13] where Cai et al. are able to identify the characters being selected on a mobile device's on screen keyboard by utilizing only the information recorded from the device's camera and accelerometer. In their research, it was found that some features, e.g., the striking and supporting force placed on the touchscreen, were highly variant between users, and as such needed to be filtered and ignored in their analysis. While not beneficial to Cai et al., features such as these that allow gesture to function as a form of identification.

7.2 GESTURE AUTHENTICATION

In a recent study on gesture-based mobile authentication, Feng et al. utilized data recorded from a smartphone's embedded sensors and those of a custom-made sensor glove in order to uniquely identify a user [14]. A false positive rate (FPR) of 2.15% and a false negative rate (FNR) of 1.63% were achieved through the use of decision tree, random forest, and Bayes Net classifiers. Although the results are encouraging, requiring the use of such a specialized piece of equipment when interacting with a smartphone is not entirely feasible outside of a laboratory environment, largely due to the added cost and inconvenience placed on an end user by such a device.

7.3 ALTERNATE DEFINITIONS

Conventionally, "gesture" has been used to describe a user making a unique, dedicated motion using their entire arm, wrist, and hand while holding a device [7][9][11][15]. In the significant body of research that identifies gestures as such, a low FPR and FNR were achieved when using them as a form of authentication by utilizing a very efficient dynamic time warping (DTW) algorithm to assist with classification [7][9][11][15]. Although fundamentally different in nature, the success of the DTW algorithm when applied to these conventional gestures likely translates well into our modern definition, as both utilize similar information collected from device sensors. However, up to this point, our work has focused on the use of the previously described algorithms over these.

8.0 CONCLUSION

In this paper, we proposed a gesture-based mobile authentication scheme referred to as *Invisible Shield*. We defined two distinct authentication scenarios, "one versus many" and "one versus all," which allowed for the definition of classification algorithms specifically suited to each.

In the case of OVM, two classification algorithms were investigated, i.e., a *k-nearest neighbor* and a generative model system based on Gaussian distributions. High classification accuracy was achieved in each system, 90.7% and 89%, respectively. Further, the results for each case revealed that gesture data formatted using a spatial standardization method and naively normalized yielded the highest classification accuracy. In addition to this, it was shown that high classification accuracy can be achieved when using a sample rate as low as 4Hz, as the results indicate there is a significant amount of user unique gesture information that occurs in the region of approximately 2Hz.

As part of OVA analysis, a generative model system based on Gaussian distributions is examined and applied, yielding an EER as low as 7.7%. Results here mirror those found in OVM, with spatially standardized data yielding more accurate results.

Finally, system resource utilization by the above algorithms is examined. It is discovered that a *k*NN algorithm only requires 0.4% of a user's battery life per day, making it an ideal choice for continuous, background authentication.

BIBLIOGRAPHY

[1] P. I. LLC, "Smartphone Security: Survey of U.S. consumers," AVG Technologies, 2011.

[2] M. Federico *et al.*, "Fast, Automatic iPhone Shoulder Surfing," in Proc. of *ACM Conf. on Computer and Communications Security* (*CCS*), 2011.

[3] Aviv, *et al.*, "Smudge Attacks on Smartphone Touch Screens," in Proc. of *USENIX Conf. on Offensive Technologies*, 2010.

[4] Motorola Mobility LLC, Hello Skip Goodbye PIN Introducing Motorola Skip for Moto X, http://motorola-blog.blogspot.com/2013/08/hello-skip-goodbye-pin-introducing.html

[5] M. B. Trudeau *et al.*, "Thumb Motor Performance Varies by Movement Orientation, Direction, and Device Size During Single-Handed Mobile Phone Use," Journal of *Human Factors and Ergonomics Society,* 2012.

[6] M. Trudeau *et al.*, "Thumb Motor Performance is Greater for Two-Handed Grip Compared to Single-Handed Grip on a Mobile Phone," in Proc. of *Human Factors and Ergonomics Society*, 2012.

[7] A. Shirazi *et al.*, "Assessing the Vulnerability of Magnetic Gestural Authentication to Video-based Shoulder surfing Attacks," in Proc. of *Human Factors in Comp. System*, 2012.

[8] D. Odell *et al.*, "Enabling Comfortable Thumb Interaction in Tablet Computers: a Windows 8 Case Study," in Proc. of *Human Factors and Ergonomics Society,* 2012.

[9] Y. Niu *et al.*, "Gesture Authentication with Touch Input for Mobile Devices," in Proc. of *Security and Privacy in Mobile Information and Communication System*, 2012.

[10] Nickel *et al.*, "Benchmarking the Performance of SVMs and HMMs for Accelerometer-based Biometric Gait Recognition," in Proc. of *Signal Proc. and Info. Technology,* 2011.

[11] J. Liu *et al.*, "User Evaluation of Lightweight User Authentication with a Single Tri-axis Accelerometer," in Proc. of *Human-Computer Interaction with Mobile Devices and Services,* 2009.

[12] N. Kirschnick, *et al.*, "User Preferences for Biometric Authentication Methods and Graded Security on Mobile Phones," in Proc. of *Usable Privacy and Security*, 2010.

[13] L. Cai *et al.*, "TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion," in Proc. of *USENIX Workshop on Hot Topics in Security*, 2011.

[14] T. Feng *et al.*, "Continuous Mobile Authentication using Touchscreen Gestures," in Proc. of *Int'l Conf. on Technologies for Homeland Security*, 2012.

[15] J. Casanova *et al.*, "Authentication in Mobile Devices through Hand Gesture Recognition," Journal on *Info. Security*, 2012.

[16] Jolliffe, "Principal Component Analysis," Springer Statistics, 2002.

[17] K. Wienberger *et al.*, "Convex Optimizations for Distance Metric Learning and Pattern Classification," IEEE Magazine on *Signal Proc.*, 2010.

[18] OVUM Analysis, "Ovum Expects Smartphone Shipments to Reach 1.7 Billion in 2017 and Android to Dominate as OS", http://ovum.com/section/home/

[19] E. Marieb, Human Anatomy & Physiology, 6th Ed. Pearson.

[20] M. Shahzad *et al.*, "Secure Unlocking of Mobile Touch Screen Devices by Simple Gestures: You can see it but you cannot do it." in Proc. of *Mobile Comp. & Networking*. 2013.

[21] E. Kreyszig, Applied Mathematics, 4th Ed. Wiley Press.

[22] M. F. Lupu *et al.*, "Bandwidth Limitations in Human Control Tasks," in Proc. of the *Int'l. Conf. on Robotics*, 2011