Hindawi
Mobile Information Systems
Volume 2020, Article ID 4165457, 12 pages
https://doi.org/10.1155/2020/4165457

Research Article

Enhancing Accuracy in a Touch Operation Biometric System: A Case on the Android Pattern Lock Scheme

Allan Ng'ang'a and Paula M. W. Musuva

School of Science and Technology, United States International University Africa, Nairobi 47381-00100, Kenya

Correspondence should be addressed to Allan Ng'ang'a; aknganga@yahoo.com

Received 13 February 2020; Revised 14 May 2020; Accepted 21 May 2020; Published 10 June 2020

Academic Editor: Nicola Bicocchi

The main objective of this research study is to enhance the functionality of an Android pattern lock application by determining whether the time elements of a touch operation, in particular time on dot (TOD) and time between dot (TBD), can be accurately used as a biometric identifier. The three hypotheses that were tested through this study were the following–H_1: there is a correlation between the number of touch stroke features used and the accuracy of the touch operation biometric system; H_2: there is a correlation between pattern complexity and accuracy of the touch operation biometric system; H_3: there is a correlation between user training and accuracy of the touch operation biometric system. Convenience sampling and a within-subjects design involving repeated measures were incorporated when testing an overall sample size of 12 subjects drawn from a university population who gave a total of 2,096 feature extracted data. Analysis was done using the Dynamic Time Warping (DTW) Algorithm. Through this study, it was shown that the extraction of one-touch stroke biometric feature coupled with user training was able to yield high average accuracy levels of up to 82%. This helps build a case for the introduction of biometrics into smart devices with average processing capabilities as they would be able to handle a biometric system without it compromising on the overall system performance. For future work, it is recommended that more work be done by applying other classification algorithms to the existing data set and comparing their results with those obtained with DTW.

1. Introduction

Mobile computing devices have been growing in popularity globally over the recent years. In Kenya, a report by Communications-Authority [1] highlighted that there were 38.3 million mobile subscriptions coupled with a mobile penetration of 89.2%. Additionally, Jumia [2] reported that, of the 3.1 million devices that they had sold in 2015, 1.8 million (58%) were smartphones. The ubiquity and convenience of mobile computing phones have resulted in them being a rich source of personal information. This is because their usage has been extended to high security activities such as mobile banking transactions which require utmost protection of the credentials in use. Most of the mobile devices currently in use maintain the privacy and security of the device by authenticating a user at each login. The most common method for doing this has been with text-based password schemes which Borkar et al. [3] showed to be a weak form of security. Research by Sae-Bae et al. [4] showed that the speed at which users type on flash glass, what is currently on smart mobile devices, was 31% slower that typing on a physical keyboard. The same research showed that users countered this problem by shortening their passwords which in turn led to a shorter login time and thereby resulted in an insecure mobile computing environment. From this research, it was inferred that login time was very critical on the part of the users when it came to authenticate them.

These challenges have led researchers to explore the use of graphical passwords as an alternative to the text-based passwords as they had been shown to have better cognitive features. Android pattern lock passwords, the subject focus of this research, have however been shown to have several inherent weaknesses. The first is that the password created

had a low entropy making it easy to break as was highlighted by Aviv et al. [5] who showed that the password space of an Android pattern was only able to generate 389,112 possible patterns a factor that made it prone to automated brute-force attacks. The second weakness is that they were vulnerable to smudge attacks which could have potentially led to the extraction of sensitive information as highlighted in [5] and [6]. The third weakness according to Sae-Bae et al. [4] was that the Android pattern lock was susceptible to shoulder-surfing attacks where an attacker could stand closely to the person unlocking the phone and see the password.

The possibility of integrating biometrics with the Android pattern lock to enhance security of the Android pattern lock emerged as the next frontier of research with Xu et al. [7] highlighting that some of the data that could be collected during pattern input included the curve, size, timing, and pressure of a touch operation. They further highlighted that the most frequently used touch operations were the keystroke, pinch, and slide. In line with this, this study sought to establish whether the time elements of a touch operation could be leveraged upon to as a biometric input to improve the accuracy of an Android pattern lock scheme. The choice of behavioural rather than physiological biometrics was intentional because it has been shown by Jesse [8] to be potentially cheaper in terms of implementation. This study therefore seeks to establish ways in which the authentication accuracy on Android smart phones can be enhanced using multiple touch biometric features, password complexity, and training. The scope of this study was limited to mobile devices that use the Android Operating System as the graphical pattern lock application that came with such devices by default was utilized.

2. Related Works

Keystroke Biometrics has previously been studied in an aim to establish whether it can be incorporated as a reliable added layer of security. Specifically, researchers have narrowed down their research based on the various features that can be extracted from modern mobile devices.

Closely related to our study, the keystroke features that Xu et al. [7] analysed were Max-Size of One Tap, Max-Pressure of One Tap, Dwell Time (TOD), and Flight Time (TBD) with data acquired from 32 participants and the Support Vector Machine (SVM) algorithm used for classification. For their accuracy results, keystroke was ranked second in terms of overall distinctiveness performance with handwriting ranked first. With regard to permanence, keystroke ranked third with slide and pinch recording higher accuracy scores after 21 days. When error rate was calculated with the incorporation of additional users to model the mock attacker, only keystroke showed a consistent improvement in accuracy up until the 30^{th} user. Handwriting and pinch all stagnated in performance after the 28^{th} user while slide had a decline in performance after the 28^{th} user. Their study required the users to get familiar with the acquisition tool before data collection began, and while this was helpful in acquiring consistent data, the downsides are that it does not simulate a real-world environment with diverse users some of who may not be very familiar with using touch screen smart phones.

Alghamdi and Elrefaei [9] extracted 31 features for their study, some of which included Down Time, Event Time, Hold Time, Up-Down Time, and Down-Down Time. They used the median vector proximity (MVP) classifier to assess the performance of the features acquired. They were able to obtain an equal error rate of 12.9% with the 31 features. Once they added an additional two features, namely, finger size and pressure, the equal error rate value dropped to 12.2%, thereby concluding that more features extracted resulted in more accuracy. The extraction of multiple features resulted in low EER figures; however, it raises the potential challenge of keeping device processing overheads to a bare minimum as this is the only way biometric technology can successfully permeate all levels of smartphones irrespective of their technological capabilities.

Sen and Muralidharan [10] captured a 4-digit passcode from 10 participants using an Android application where they extracted features including key hold time, key hold pressure minimum, key hold pressure maximum, and interkey time. They also simulated a same-user-and-attacker scenario. Among the 4 different classifiers which they tested their data against, multilayer perception (MLP) gave them the highest accuracy with an FRR of 14.06% and FAR of 14.1% which was better than their target system which had FRR of 21% and FAR of 19%. The challenge is that 4-digit passcodes, despite their wide usage, already suffer from low entropy. While the results here are promising, it leaves much more work to be done to ensure the added layer of security achieves maximum protection.

Lee et al. [11] developed an Android application to capture data from users who were to enter a 6-digit PIN, "766420." They aimed to extract two groups of biometric data, keystroke and motion (accelerometer). The keystroke data extracted was time (Down Time and Flight Time), size, and coordinates. Using their distance-based classifier, Euclidean and Manhattan, they were able to show that the accuracy of their model improved once motion data was added to keystroke data. In their case, EER dropped from 8.94% to 7.89%. While having the 6-digit PIN improves the password space when compared with a 4-digit PIN, the underlying issue remains how usable this would be for regular users who already struggle to come up with convincing 4-digit PINs that are not predictable. It is important for the system to maintain its usability while still being secure.

Angulo and Wästlund [12] collected data from 32 participants with a diverse mix of characteristics based on education, gender, and age. The features they extracted from their modified Android pattern lock device were the finger-in-dot time and the finger-in-between-dot, time all captured in milliseconds. The tests were performed with different Android devices such as Samsung Galaxy SII, Nexus S, HTC Legend, and HTC vision. They opted to test their EER results against various classifiers such as Euclidean, Manhattan, SVM, and Random Forest. They achieved the best result from the Random Forest classifier which had a mean EER of 10.39%, thereby confirming their research objectives. While

they managed to achieve their desired objectives, the methodology employed where data was gathered from different devices raises the problem of consistency in the data. It is therefore not known if they would have acquired the same results if the users had been requested to switch their input devices after a given number of trials. Their study also failed to mention whether they accounted for the varying degree of participant experiences in using touch screen smart phones, a factor which could have potentially contributed to varying results.

3. Materials and Methods

According to Zikmund [13], experimental designs involve four major design elements. These include manipulation of the independent variable(s), selection and measurement of the dependent variable(s), selection and assignment of experimental subjects, and control over extraneous variables. The specific type of experimental design that is implemented in this research study is a quasiexperimental design which involves a static group design where each of the subjects are identified as being a member of either the control or the experimental group. This research study also involves the implementation of a cross-sectional approach whereby comparisons are made at a single point in time as highlighted in [14].

This research study involves the use of two measurable touchstroke time variables, the finger-in-dot time (termed as time on dot) and the finger-in-between-dot time (called time between dot), because the time element of a touch operation has been previously shown by the researchers such as Dhage et al. [15] and Alghamdi and Elrefaei [9] to produce the most consistent and actionable information especially when calculating the false acceptance rates (FARs) and false rejection rates (FRRs). This was partially informed by the need to keep processing overheads to a bare minimum to make the system lucrative for any future integration with low-end smart devices. This approach differs from that of [9] who extracted a total of 31 features. While this helped improve their accuracy results, they also had to contend with the increased processing overheads that comes with the quantity of features. This research also involved gathering of input data from one device rather than multiple devices. This helps maintain data collection consistency. In contrast, similar research conducted by Angulo and Wästlund [12] involved data collection using multiple Android devices. The incorporation of pattern complexity was motivated by the research conducted by De Luca et al. [16] who while making use of these distinctions failed to capture the individual accuracy levels attributed to the varying lengths of patterns. This is a key gap that this research aims to cover. Similarly, the effect on user training on accuracy was motivated by studies performed by Zheng et al. [17]. However, this research considers the effects that "no training" has to accuracy and compares it to data obtained after training has been incorporated.

With all this factored in, the hypotheses formulated were as follows: H_1: there is a correlation between the number of touchstroke features used and the accuracy of the touch operation biometric system; H_{o1}: there is no correlation between the number of touchstroke features used and the accuracy of the touch operation biometric system; H_2: there is a correlation between pattern complexity and accuracy of the touch operation biometric system; H_{o2}: there is no correlation between pattern complexity and accuracy of the touch operation biometric system; H_3: there is a correlation between user training and accuracy of the touch operation biometric system; H_{o3}: there is no correlation between user training and accuracy of the touch operation biometric system. The accompanying conceptual framework based on the variables established and hypotheses formulated is shown in Figure 1.

The study population is primarily from all United States International University-Africa (USIU-A) students, members of faculty, and nonteaching staff from the Chandaria School of Business, School of Humanities and Social Sciences, and School of Pharmacy and Health Sciences. However, this study excludes the School of Science and Technology because they may have had a technical advantage over a common representative user in the population. Two main age demographic groups are targeted, namely, millennials and the old, both of whom need to be familiar with Android-powered smart devices. The reason the population is limited to Android-powered smart devices is because the study makes use of a modified Android pattern lock application that extracts the behavioural biometric measurements that we analysed in the study. Due to the limited time allocated for the study, nonprobability sampling as defined by Adam [18] was utilized. However, it is acknowledged that nonprobability sampling may have the tendency of resulting in bias during data collection and as such it would not be safe to assume that the sample group gave a correct representation of the target group.

A within-subjects design involving repeated measures is implemented when testing H_1, H_{o1}, H_2, and H_{o2} hypotheses, thereby resulting in the same 8 subjects used for all measures across both the control and experimental groups as highlighted in Tables 1 and 2.

Testing of H_3 and H_{o3} hypotheses involves the selection of a new set of 4 subjects for the experimental group to eliminate the possibility of training through exposure to previous procedures. This is highlighted in Table 3. The control group here was the set of first four subjects used when testing H_2 and H_{o2} hypotheses.

Objectives 1 and 2 involved the collection of user data without incorporating the aspect of user training. This involved the collecting of 3 input cycles for a total of 6 inputs. Research by Heimark [19] helped in arriving at this figure as they established that 5 inputs were the maximum and most ideal for testing without introducing the training factor that comes with repeated use of a system. Because our system implementation only accepted input cycles in pairs, the minimum number of pairs that we could use without incorporating the aspect of training and while still meeting the minimum threshold of 5 captures as defined by [19] was to use 3 input cycles which translated to 6 captures in total. This is highlighted in Table 4.

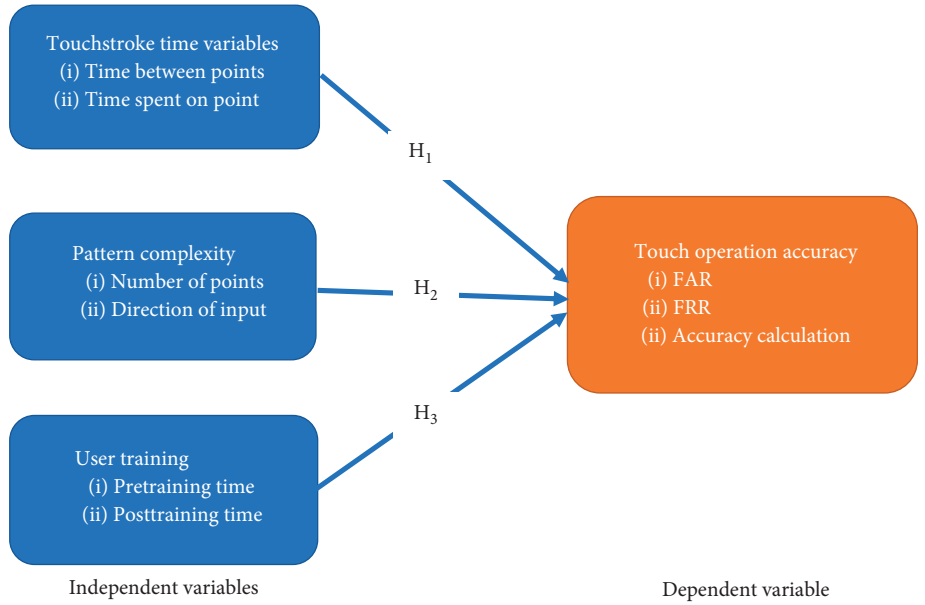

Figure 1: Conceptual framework showing the independent and dependent variables and the relationships between the 3 hypotheses.

Table 1: Sample size for H_1 and H_{o1} testing.

Blocking variable (gender)	Blocking variable (age)	Control group Simple password (TBD)	Experimental group Simple password (TBD + TOD)
Male	Millennials (<35)	2	2
	Old (>35)	2	2
Female	Millennials (<35)	2	2
	Old (>35)	2	2
Total		**8**	**8**

Table 2: Sample size for H_2 and H_{o2} testing.

Blocking variable (gender)	Blocking variable (age)	Experimental group Complex password (TBD)	Complex password (TBD + TOD)
Male	Millennials (<35)	2	2
	Old (>35)	2	2
Female	Millennials (<35)	2	2
	Old (>35)	2	2
Total		**8**	**8**

Table 3: Sample size for H_3 and H_{o3} testing.

Blocking variable (gender)	Blocking variable (age)	Experimental group Complex password with training (TBD)	Complex password with training (TBD + TOD)
Gender	Male	1	1
	Female	1	1
Age	Millennials (<35)	1	1
	Old (>35)	1	1
Total		**4**	**4**

Objective 3 involved the collection of user data while incorporating the aspect of user training. This involved the collecting of 10 input cycles for a total of 20 inputs. Research by Zheng et al. [17] was able to observe that by increasing the number of actions in user training, the user behavioral patterns became more precise. However, the accuracy remained at the same level after incorporation of 20 user actions. This is highlighted in Table 5 below:

This translates to an overall sample size of 12 subjects who give a total of 2,096 feature extracted data which includes time between dot (TBD) and time on dot (TOD). Selection of 12 subjects is backed by a research conducted by Miluzzo et al. [20] who conducted their research using 10 subjects and were able to obtain a total of 40,000 tap samples for their analysis having extracted both accelerometer and gyroscope data. Other researchers who also extracted a lot of features from few individuals were Beton et al. [21] who were able to obtain 120 intraclass authentication attempts and 1470 interclass attempts (simulating attacks) from 15 subjects. Because of the depth of this research, it was concluded that the 12 subjects selected were going to provide enough data.

The analysis method implemented is the Dynamic Time Warping (DTW) algorithm which was obtained from Wang [22]. According to De Luca et al. [16], the algorithm works by finding an optimal path between two vectors and is extensively used within fields of biometrics such as speech recognition, gait recognition, and fingerprint verification. The resulting output from the comparison, known as a warp distance, determines how similar a subsequent input is to the reference set. Consequently, a warp distance of zero means that the subsequent input and reference set are identical. Closer to our study, DTW [22] was used by [16, 21] to analyse their features. The DTW [22] values are pivotal especially when it comes to calculating the false acceptance rates (FARs), false rejection rates (FRRs), and accuracy as described by [16]. This algorithm was run in MATLAB R2016b downloaded from MathWorks [23].

An Android pattern lock application that captures the identified touch operation biometric variables was designed using Android Studio downloaded from the developer [24] installed on a HP laptop with a core i5 2 GHz processor, 4 GB RAM, and 256 GB SSD. Android Studio [24] was used to modify an Android pattern lock application so that the time on dot and time between dot data could be extracted. These data are stored by the application in an SQLite database and can be exported to a text file or HTML format file. Initial testing and running of the application is done using the inbuilt Android Studio Emulator which requires that the processor can handle a virtualization environment. When the application is launched, the user is presented with an interface that requires them to calibrate the touch points numbered from 1 to 9 as shown in Figure 2.

Figure 3 highlights the interface that users interact with. It has the pattern-input recording capability embedded in it. Additionally, the application is modified to generate a set of results from the user input as illustrated in Figure 4. Some of the statistics that can be generated are path stats (time between dot), button stats (time on dot), and grouped stats (time on dot + time between dot).

Figures 5 and 6 show the html output files for the time spent on dot (TOD) and time between dot (TBD) for the two pattern inputs. This is a repetition of the 1-2-5-8-9 sequence which was the path taken for our demo pattern input. The captures are in milliseconds for greater accuracy.

Input of the simple password involved the user sequentially touching the pattern lock points in the order, 1-2-5-8-9, making a total of 5 points. Additionally, input of the complex password involved the user sequentially touching the pattern lock points in the order, 7-5-9-6-2-1-4. The research study by Beton et al. [21] helped guide us in the selection of these points. Figures 7 and 8 illustrate the simple and complex password as input by the user.

The system operation can be best summarised through the activity diagram in Figure 9.

To obtain the reference set, each of the captures for a user was compared with the other using DTW [22]. The outputs in the matrix, as shown in Table 6, were then used to calculate the mean and standard deviation. These two values were used to calculate the upper boundary which defined the limit to which unlocks were determined to be valid. This upper boundary was calculated by summing up the mean and standard deviation values. In our case, the upper boundary value for user 1 was 0.172587 as highlighted in Table 7.

All the captures that were below the upper boundary value were the true positive (TP) inputs for the user while those that were above the upper boundary value were the false negative (FN) inputs for the user. Figure 10 goes ahead to illustrate this. According to De Luca et al. [16], true positives (TPs) are the correctly accepted users and false negatives (FNs) are the wrongly rejected users. These two values were used to determine the false rejection rate (FRR) for the user. Additionally, the following equation was defined in [16] which is used to calculate the FRR value:

$$\text{false rejection rate (\%)} = 100 * \frac{(\text{sum of FN})}{(\text{sum of FN} + \text{sum of TP})}. \quad (1)$$

In order to simulate an "attack" scenario, user 1 inputs were matched against inputs from the other 7 subjects so as to establish the number of inputs that would fall below the upper boundary (falsely accepted) and those that would fall above the upper boundary (correctly rejected). With reference to Table 8, the values in the row header were those of user 1 while those in the column header were those of user 2, whose values we used to "attack" user 1. Each user 1 capture was compared to the corresponding user 2 captures using DTW [22] and the resultant output inserted in the corresponding cell of the matrix.

A scatter diagram chart, as shown in Figure 11, was then drawn using these values with the upper boundary established during the generation of the reference set also included. This helped us determine the number of user 2 inputs that were correctly rejected (true negatives) and those that were wrongly accepted (false positives). According to

Table 4: Feature extraction of objectives 1 and 2.

	Objective 1–simple TOD	Objective 1–simple TBD	Objective 2–complex TOD	Objective 2–complex TBD
Feature data per input cycle	10	8	14	12
No. of input cycles	3	3	3	3
Total data per user	30	24	42	36
No. of users	8	8	8	8
Total feature extracted data	240	192	336	288

Table 5: Feature extraction of objective 3.

	Objective 3–complex TOD	Objective 3–complex TBD
Feature data per input cycle	14	12
No. of input cycles	10	10
Total data per user	140	120
No. of users	4	4
Total feature extracted data	560	480

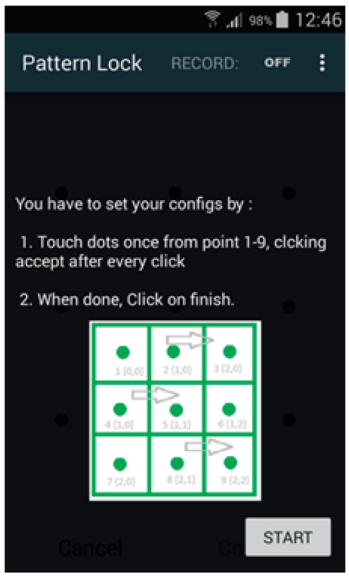

Figure 2: Touch point calibration.

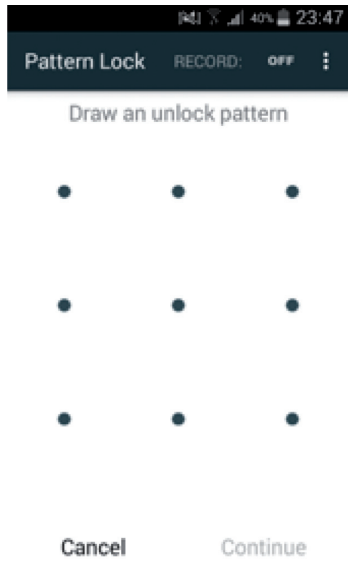

Figure 3: User interface.

De Luca et al. [16], true negatives (TNs) are defined as the correctly rejected attackers and false positives (FPs) as the wrongly accepted attackers. These two values were used to determine the false acceptance rate (FAR) for the user. Additionally, in [16], the following equation was defined which is used to calculate the FAR value:

$$\text{false acceptance rate } (\%) = 100 * \frac{(\text{sum of FP})}{(\text{sum of FP} + \text{sum of TN})}. \quad (2)$$

Table 9 was generated once user 1 had been "attacked" by all the other users. It helped highlight some of the key attributes under investigation such as the false acceptance rate (FAR) and the false rejection rate (FRR). Additionally, the accuracy as calculated through equation (3) is also presented in the table:

$$\text{accuracy } (\%) = 100 * \frac{(\text{sum of TN} + \text{sum of TP})}{(\text{sum of TN} + \text{sum of TP} + \text{sum of FN} + \text{sum of FP})}. \quad (3)$$

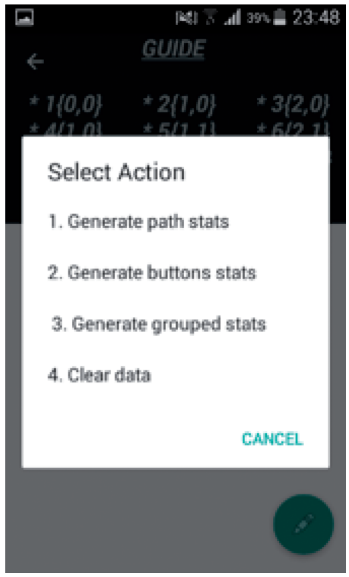

Figure 4: Generating statistics.

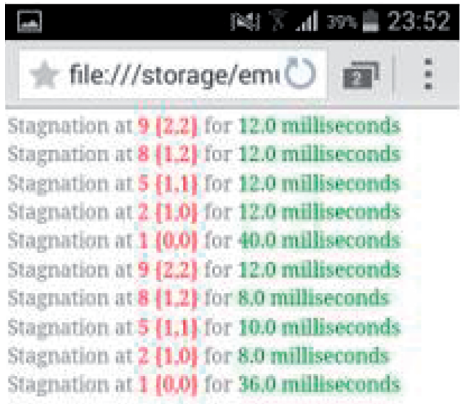

Figure 5: Time-on-dot HTML output.

Figure 6: Time-between-dot HTML output.

Figure 7: Simple password.

Figure 8: Complex password.

4. Results and Discussion

The procedures highlighted above for generating the user summary tables were replicated for all other users and results grouped according to the various hypotheses under testing. Presentation of results and subsequent discussion of the same will be according to these groupings.

4.1. Number of Touchstroke Features Used and Accuracy. The hypotheses being tested here were as follows: H_1: there is a positive correlation between the number of touchstroke features used and the accuracy of the touch operation biometric system; H_{o1}: there is no positive correlation between the number of touchstroke features used and the accuracy of the touch operation biometric system. A

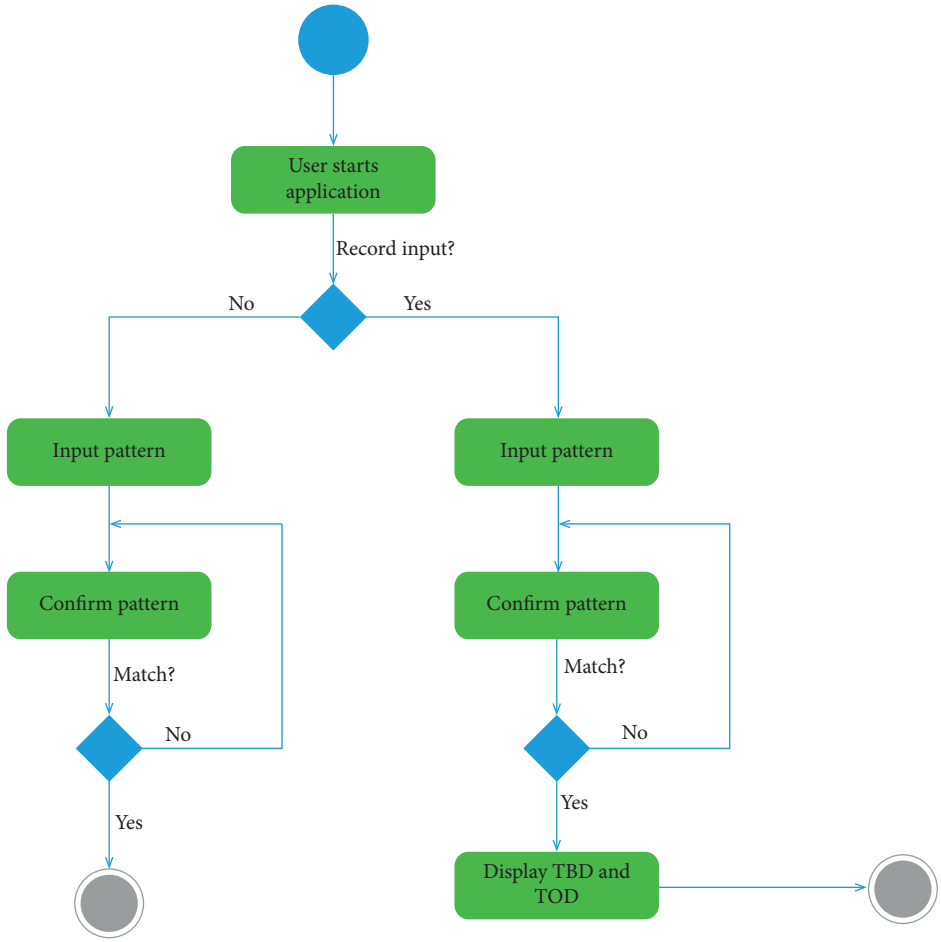

Figure 9: Activity diagram.

Table 6: Genuine user 1 reference set.

Genuine user 1		1	2	3	4	5	6
		1141	1143	923	876	849	1380
1	1141	0					
2	1143	0.000004	0				
3	923	0.0475	0.0484	0			
4	876	0.0702	0.0713	0.0022	0		
5	849	0.0853	0.0864	0.0055	0.000729	0	
6	1380	0.0571	0.0562	0.2088	0.254	0.282	0

Table 7: Genuine user 1 upper boundary.

Avg.	0.085042
Std. dev.	0.087545
Upper boundary	0.172587

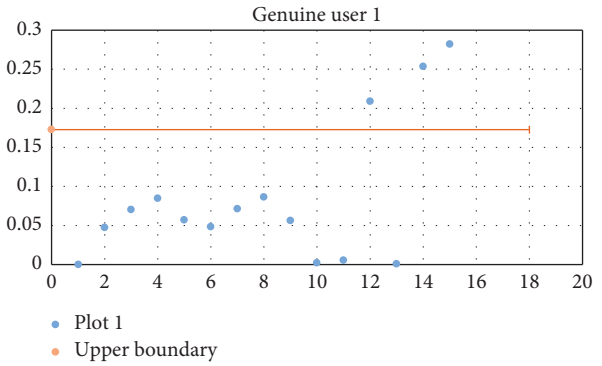

Figure 10: Genuine user 1 chart.

summary of results obtained for this hypothesis testing is highlighted in Table 10.

Results for H_1 revealed an increase in accuracy by lowering the false rejection rate (FRR) from 20% to 17% when an additional time feature was used. However, the false acceptance rates (FARs) increased from 34% to 39%, leading to an overall decline in accuracy from 68% to 62%. In comparison to a similar study, Alghamdi and Elrefeai [9] implemented the median vector proximity (MVP) classifier to assess the performance of the features acquired. They were able to obtain an equal error rate of 12.9% with 31 features. Once they added additional two features, namely, finger size and pressure, the equal error rate value dropped to 12.2%, thereby confirming that more features used resulted in more accurate systems. The research on the median vector proximity (MVP) classifier has been conducted by Al-jarrah [25] where he was able to establish that it was generally less

Table 8: Attacking user 1 with user 2.

Attack 1 with 2		1	2	3	4	5	6
		401	472	457	512	468	674
1	1141	0.5476	0.4476	0.4679	0.3956	0.4529	0.2181
2	1143	0.5506	0.4502	0.4706	0.3982	0.4556	0.22
3	923	0.2725	0.2034	0.2172	0.1689	0.207	0.062
4	876	0.2256	0.1632	0.1756	0.1325	0.1665	0.0408
5	849	0.2007	0.1421	0.1537	0.1136	0.1452	0.0306
6	1380	0.9584	0.8245	0.8519	0.7534	0.8317	0.4984

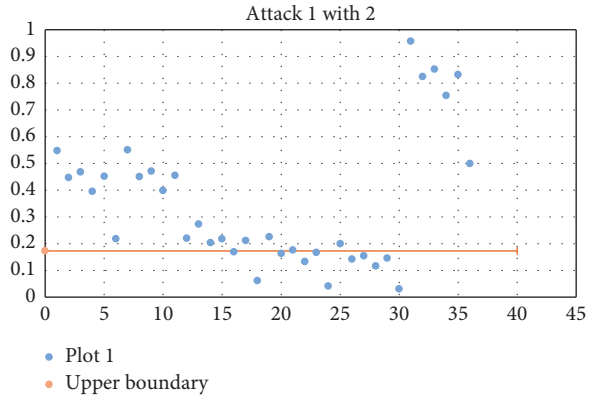

Figure 11: Attacking user 1 chart.

Table 9: User 1 accuracy summary table.

User 1						
True positives	False negatives	True negatives	False positives	Accuracy (%)	False rejection rate (FRR) (%)	False acceptance rate (FAR) (%)
12	3	144	108	58%	20	43

Table 10: H_1/Ho_1 summary table.

User	Control (TBD)			Experiment (TBD + TOD) accuracy		
	FAR (%)	FRR (%)	Accuracy (%)	FAR (%)	FRR (%)	Accuracy (%)
1	20	43	58	20	28	72
2	27	39	62	20	63	40
3	13	32	69	13	57	45
4	20	54	48	20	40	61
5	20	22	83	20	14	86
6	13	11	94	13	45	57
7	27	36	65	13	63	40
8	20	34	67	20	1	98
Avg.	**20**	**34**	**68**	**17**	**39**	**62**

affected by outliers (extreme values). Additionally, a comparison he did with 15 various classifier algorithms showed the MVP classifier to have the highest anomaly detection amongst all that were tested which included Manhattan, nearest neighbour, neural network, and fuzzy logic classifiers. From our results, we therefore concluded that the decrease in accuracy when an extra feature was extracted could be attributed to the DTW algorithm failing to manage the effect of outliers. The H_1 hypothesis was therefore rejected.

4.2. Pattern Complexity and Accuracy. The hypotheses being tested were as follows: H_2: there is a positive correlation between pattern complexity and accuracy of the touch operation biometric system; H_{o2}: there is no positive correlation between pattern complexity and accuracy of the touch biometric system. A summary of results obtained for this hypothesis testing is highlighted in Table 11 where one feature (time between dot) and a simple password were used.

When two features (time between dots and time on dot) were extracted, summary Table 12 was obtained.

Table 11: H_2/H_{o2} one-feature summary table showing improved accuracy.

User	Control (simple password)			Experiment (complex password)		
	FAR (%)	FRR (%)	Accuracy (%)	FAR (%)	FRR (%)	Accuracy (%)
1	20	28	72	33	61	40
2	20	63	40	27	9	90
3	13	57	45	13	27	74
4	20	40	61	13	8	92
5	20	14	86	27	57	45
6	13	45	57	27	23	77
7	13	63	40	27	29	71
8	20	1	98	20	41	60
Avg.	**17**	**39**	**62**	**23**	**32**	**69**

Table 12: H_2/H_{o2} two-feature summary table showing decreased accuracy.

User	Control (simple password)			Experiment (complex password)		
	FAR (%)	FRR (%)	Accuracy (%)	FAR (%)	FRR (%)	Accuracy (%)
1	20	43	58	33	85	18
2	27	39	62	20	71	31
3	13	32	69	13	16	84
4	20	54	48	20	9	91
5	20	22	83	20	55	47
6	13	11	94	20	27	73
7	27	36	65	20	32	69
8	20	34	67	20	60	43
Avg.	**20**	**34**	**68**	**21**	**44**	**57**

Table 13: H_3/H_{o3} one-feature summary table.

User	Control (no training)			Experiment (training)		
	FAR (%)	FRR (%)	Accuracy (%)	FAR (%)	FRR (%)	Accuracy (%)
1	33	85	18	15	3	95
2	20	71	31	12	25	77
3	13	16	84	15	26	76
4	20	9	91	14	23	78
Avg.	**22**	**45**	**56**	**14**	**19**	**82**

Results for H_2 when using one-touchstroke features (time between dot) showed a 7% increase in accuracy from 62% in the case of a simple password to 69% when a complex password was used. However, the degree of accuracy decreased by 11 percentage points when an extra feature was extracted. This is in correlation to what was established when testing H_1/H_{o1} where it was seen that the extraction of an extra feature resulted in decreased accuracy levels due to the DTW algorithm [22] failing to manage the effect of outliers. The H_2 hypothesis for the case of one feature extracted was therefore accepted. H_2 hypothesis for the case of two features extracted was rejected.

4.3. User Training and Accuracy. The hypotheses being tested here were as follows: H_3: there is a positive correlation between user training and accuracy of the touch operation; H_{o3}: there is no positive correlation between user training and accuracy of the touch operation. A summary of results obtained for this hypothesis testing is highlighted in Table 13, where one feature (time between dot) and a simple password were used.

When two features (time between dot and time on dot) were extracted, summary Table 14 was obtained.

Results for H_3 when using one-touchstroke features (time between dot) showed a 26% increase in accuracy from 56% in the case of no training to 82% when a training was incorporated. It is worth highlighting that the complex password was used throughout testing of H_3. However, the degree of accuracy decreased by 8 percentage points when an extra feature was extracted. These findings are consistent with what was established when testing both H_1/H_{o1} and H_2/H_{o2} where it was seen that the extraction of an extra feature resulted decreased the accuracy levels due to the DTW algorithm [22] failing to manage the effect of outliers. The H_3 hypothesis for the case of one feature extracted was therefore accepted. H_3 hypothesis for the case of two features extracted was rejected.

Table 14: H_3/H_{o3} two-feature summary table.

User	Control (no training)			Experiment (training)		
	FAR (%)	FRR (%)	Accuracy (%)	FAR (%)	FRR (%)	Accuracy (%)
1	33	61	40	9	71	38
2	27	9	90	15	46	58
3	13	27	74	16	0.7	98
4	13	8	92	12	55	51
Avg.	**23**	**32**	**69**	**13**	**43**	**61**

5. Conclusions

The main findings of this study show that the extraction of one feature coupled with a complex password and user training yielded increased accuracy levels throughout testing of all hypotheses. However, the extraction of an extra feature resulted in decreased accuracy levels, and this was attributed to the DTW algorithm [22] failing to properly manage the effect of outliers when extra features are introduced.

The contribution that was made through this research study was that it was shown that the extraction of one-touchstroke biometric feature coupled with user training was able to yield high average accuracy levels of up to 82%. This helps build a case for the introduction of biometrics into smart devices with average processing capabilities as they would be able to handle a biometric system without it compromising on the overall system performance.

For future work, it is recommended that more work be done by applying other classifiers such as the median vector proximity (MVP) classifier as used by [9] to the existing data set and comparing their results with those obtained with DTW [22]. The abovementioned classifiers were shown to be good at eliminating outliers and noise in the data set, thereby producing more accurate results. Additionally, further research can explore whether the use of other touchstroke biometric features can have a better impact on accuracy as opposed to those used in this study.

Data Availability

The datasets generated and analysed to support the findings of this study have been deposited in Mendeley Data (https://doi.org/10.17632/hsprcwcvhy.1).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Communications-Authority, "Quarterly sector statistics report," https://ca.go.ke/wp-content/uploads/2018/02/Sector-Statistics-Report-Q3-2015-16.pdf, 2016.

[2] Jumia, "The growth of the smartphone market in Kenya," 2015, https://www.jumia.co.ke/blog/whitepaper-the-growth-of-the-smartphone-market-in-kenya.

[3] B. Borkar, S. Sheikh, and P. P. D. Kaware, "4D password mechanism," 2016, http://www.onlinejournal.in/IJIRV2I5/044.pdf.

[4] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems—CHI'12*, May 2012.

[5] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," 2010, https://www.usenix.org/legacy/events/woot10/tech/full_papers/Aviv.pdf.

[6] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures—you can see it but you can not do it," in *Proceedings of the 19th annual international conference on Mobile computing & networking—MobiCom'13*, Miami, FL, USA, Octobar 2013.

[7] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: an experimental study on smartphones," 2014, https://www.usenix.org/conference/soups2014/proceedings/presentation/xu.

[8] Jesse, "Behavioral biometrics market value will more than double by 2020," 2015, http://www.technavio.com/blog/behavioral-biometrics-market-value-will-more-double-2020.

[9] S. J. Alghamdi and L. A. Elrefaei, "Dynamic user verification using touch keystroke based on medians vector proximity," in *Proceedings of the 7th International Conference on Computational Intelligence, Communication Systems and Networks*, June 2015.

[10] S. Sen and K. Muralidharan, "Putting "pressure" on mobile authentication," in *Proceedings of the Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, Janaury 2014.

[11] H. Lee, J. Hwang, D. Kim, S. Lee, S. Lee, and J. Shin, "Understanding keystroke dynamics for smartphone users authentication and keystroke dynamics on smartphones built-in motion sensors," *Security and Communication Networks*, vol. 2018, Article ID 2567463, 10 pages, 2018.

[12] J. Angulo and E. Wästlund, "Exploring touch screen biometrics for user identification on smart phones," in *IFIP Advances in Information and Communication Technology*, vol. 375Berlin, Germany, Springer. http://dl.ifip.org/db/conf/primelife/primelife2011/AnguloW11.pdf, 2012.

[13] W. Zikmund, "Exploring marketing research," 2010, http://www.cengage.com/maintenance/index.html.

[14] Research-Excellence, "What researchers mean by "cross-sectional vs. longitudinal studies"" Institute for Work and Health, Toronto, Canada, 2015, http://www.iwh.on.ca/wrmb/cross-sectional-vs-longitudinal-studies.

[15] S. Dhage, P. Kundra, A. Kanchan, and P. Kap, "Mobile authentication using keystroke dynamics," in *Proceedings of the International Conference on Communication, Information & Computing Technology (ICCICT)*, Janaury 2015.

[16] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you! implicit authentication based on touch screen patterns," in *Proceedings of the 2012*

ACM annual conference on Human Factors in Computing Systems—CHI'12, May 2012.

[17] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: user verification on smartphones via tapping behaviors," in *Proceedings of the 22nd International Conference on Network Protocols*, October 2014.

[18] H. M. Adam, "Research population," 2016, http://www.academia.edu/5563491/Research_Population.

[19] E. Heimark, "Authentication : from passwords to biometrics erlend Heimark," 2012, https://brage.bibsys.no/xmlui/handle/11250/262687.

[20] E. Miluzzo, A. Varshavsky, and S. Balakrishnan, "TapPrints : your finger taps have fingerprints categories and subject descriptors," 2012, http://miluzzo.info/pubs/sys015fp-miluzzo.pdf.

[21] M. Beton, V. Marie, C. Rosenberger, M. Beton, and C. Rosenberger, "Biometric secret path for mobile user authentication : a preliminary study," in *Proceedings of the World Congress on Computer and Information Technology (WCCIT)*, June 2013.

[22] Q. Wang, "Dynamic time warping (computer software)," 2016, https://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping-dtw.

[23] MathWorks, "R2016b (computer software)," MathWorks, Natick, MA, USA, 2016, https://au.mathworks.com/products/new_products/release2016b.html.

[24] Developers, "Android studio (computer software)," 2016, https://developer.android.com/studio.

[25] M. M. Al-jarrah, "An anomaly detector for keystroke dynamics based on medians vector proximity," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 6, pp. 988–993, 2012, https://pdfs.semanticscholar.org/c28d/d958da2ecb49663a5badaf6b866f87281543.pdf.