Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9963129, 15 pages
https://doi.org/10.1155/2021/9963129

Research Article

Analyzing the Effectiveness of Touch Keystroke Dynamic Authentication for the Arabic Language

Suliman A. Alsuhibany and Afnan S. Almuqbil

Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia

Correspondence should be addressed to Suliman A. Alsuhibany; salsuhibany@gmail.com

Received 31 March 2021; Revised 14 July 2021; Accepted 25 August 2021; Published 11 September 2021

Academic Editor: Ding Wang

The keystroke dynamic authentication (KDA) technique was proposed in the literature to develop a more effective authentication technique than traditional methods. KDA analyzes the rhythmic typing of the owner on a keypad or keyboard as a source of verification. In this study, we extend the findings of the system by analyzing the existing literature and validating its effectiveness in Arabic. In particular, we examined the effectiveness of the KDA system in Arabic for touchscreen-based digital devices using two KDA classes: fixed and free text. To this end, a KDA system was developed and applied to a selected device operating on the Android platform, and various classification methods were used to assess the similarity between log-in and enrolment sessions. The developed system was experimentally evaluated. The results showed that using Arabic KDA on touchscreen devices is possible and can enhance security. It attains a higher accuracy with average equal error rates of 0.0% and 0.08% by using the free text and fixed text classes, respectively, implying that free text is more secure than fixed text.

1. Introduction

The rapid progress engendered by mobile devices has exponentially accelerated the use of smartphones and other digital devices [1, 2]—owing to the power of networking, mobility sensing, and mobile device computing. According to a cybersecurity statistics report [3], malware variants in mobile crypto-jacking increased from eight in 2017 to a staggering 27 in the first five months of 2018. In March 2020, the new Android malware samples per month were 482,579 [4]. Among them, Trojans are the most popular form of malware affecting Android devices, as reported by the AV-Test [5]. This increment in malware variants demands robust security mechanisms for mitigating the enhanced risk.

Security measures suffer from severe security and usability limitations. Many authentication methods have been used for mobile security, such as personal identification numbers, face recognition, and fingerprint scanning [6]. Although these mechanisms ensure security, they can be easily compromised [7]. For example, passwords can be shoulder-surfed, leaked, or guessed, or other password-defying channels can be used to break-in. Also, passwords can be shared with friends and written to remember [8, 9]. Similarly, fingerprint authentication is equally susceptible to being spoofed by imitating the fingertip structure, often generated using a concealed fingerprint [10]. The mobile system sometimes fails to recognize the fingertip, requiring multiple attempts. Similar to passwords and fingerprint recognition, facial recognition can be spoofed using a video, photo, or 3D mask to forge the faces of the mobile users [11]. Besides being vulnerable to illegitimate use, these authentication systems require additional hardware to support their services, ultimately adding to the cost of the device.

To investigate security in a mobile environment, this work considers the technology of keystroke dynamics authentication (KDA) in terms of its efficiency for ensuring the required security. In particular, biometric security authentication is divided into two characteristics: physiological and behavioral. Although the fingerprint, face, veins, and iris are all physiological characteristics unique to each user, behavioral characteristics include studying certain behavior-based patterns.

KDA relies on behavior-based authentication characteristics, specifically, the manner and rhythm of users when

typing characters. Every user has a unique behavioral pattern based on typing strength, the interval between characters, finger position, and angle of usage. The classification of keystroke dynamics is accomplished based on the target input of keystrokes in the form of either fixed or free text—the fixed text class refers to predefined text that must be entered every time the user wishes to sign in to the device/system. Alternatively, the free text class does not involve the predefined text requirement and bypasses the memorization requirement for users. Thus, this study considers both free text and fixed text classes of KDA to become the first study to adopt this approach.

Considering the challenges faced by traditional authentication methods, the dynamic behavioral techniques of biometrics have been calibrated and found harder to forge. That is, keystroke dynamics, when applied on keyboards, reveal only timing information or the elapsed time between releasing and pressing a key and the duration for which the key was held down. Based on this, behavioral authentication systems offer several advantages in countering the deficiencies of traditional authentication. First, generating the same pattern of movement is more challenging to imitate. Even when the movement pattern is imitated; differences in body structure, such as finger shape, height, and orientation on the touchscreen, can differ, leading to changes in the movement patterns. Furthermore, the built-in physical sensors in digital devices can easily detect these minor differences and consequently block access. Further, every user has a unique way of inputting data into a device; any unauthorized user can copy a password; however, it is not easy to imitate the touch style, type, and pattern of the authentic user.

Keystroke dynamics has produced substantial research, with the focus being increasingly placed on the keypad area of smartphones. Recently conducted research has reported an error equal rate (EER) of 0% [12].

These promising results are attributed to tools already embedded in keypads, such as the accelerometer, gyroscope, and other sensors, which facilitate accurate pattern information compared to a fixed keyboard. In general, there are differences between the data collected by physical keyboards and touchscreen keyboards [13]. In addition to classical timing features, keystroke dynamics on touchscreen keypads enable additional features for authentication, such as pressure on the screen during typing and the area of keys covered by the fingers [14, 15].

Although Arabic has been analyzed using physical keyboards in [16–18], revealing an efficient performance, it has not been analyzed in touchscreen keyboards. Therefore, this study analyzes the effectiveness of touch KDA for Arabic.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the methodology, including data collection, feature extraction, and classification methods for user verification. Section 4 presents the experimental results, which are further outlined and discussed in Section 5. Finally, Section 6 presents the conclusion with suggestions for future research.

2. Related Works

This section is divided into three parts. First, the literature on the use of KDA systems as a verification tool in touchscreen-based digital devices is reviewed. Second, the application of Arabic in KDA was explored. Finally, studies that have applied the KDA system to other languages are discussed.

2.1. Use of KDA in Touchscreen-Based Digital Devices. This section discusses KDA studies in terms of classes, types of features, and classification methods.

2.1.1. KDA Studies Based on the Classes. To reiterate, keystroke dynamics can be categorized into two classes: free text and fixed text. The free text has the potential to verify the authenticity of users in the log-in process alongside the capacity to continuously monitor users after log-in based on their typing pattern [19]. In contrast, the fixed text is primarily employed to protect valid users from various threats in the log-in process. Based on these characteristics, hard-keyboard-based free KDA has been actively studied [20–23]. Conversely, the utilization of free KDA on smartphone devices has received relatively little research attention.

As shown in Tables 1 and 2, the performance of the free KDA is slightly lower than that of the fixed KDA. This low performance has several limitations. First, its keystroke feature is limited, and the number of keystrokes might not be sufficient, as shown in [24]. Draffin et al. [25] failed to provide adequate information because they only used the time feature. In Gascon et al. [26], the worst performance of non-identifiable users, which can hardly be distinguished from others, managed to have a 58% true positive rate (TPR) and 35% false positive rate (FPR). This is unreasonable for practical authentication systems. The study by Kim and Kang [12] was the only one to use free text, and it managed to obtain good results, scoring 0.07% EER and 0% EER in English and Korean, respectively. Conversely, the fixed text has become a saturated research area, yielding interesting results, such as 0.01% EER in Buriro et al. [27].

2.1.2. KDA Studies Based on the Feature Type. In 2009, the earliest research was conducted to analyze KDA as a verification tool in touchscreen devices. This research was performed by Saevanee and Bhattarakosol [28] and focused on finger pressure on the touchscreen as a biometric source for keystroke dynamics analysis, using laptops as a base. With the release of Android 1.6, many new features were added to digital devices, such as fingertip size, device orientation, and device angle, which broadened the application of the technique. In 2010, a further advancement occurred after Android 2.3, which included a rotation vector, a gyroscope, a linear accelerometer, and gravity, giving new life to studies being conducted in this area. Notably, a study by Zheng et al. [29] researched using KDA on mobile touchscreens specifically. In addition to examining common touchscreen-based features, they studied the efficiency of KDA using accelerometer sensors. Kambourakis et al. [30] used Android devices and proposed two new features to evaluate keystroke dynamics: speed and distance features.

Table 1: Recent studies of the free KDA for touchscreen devices.

Study	Year	Number of subjects	Methodology	Features	Classifier	FAR, FRR, or EER
[25]	2013	13	15 keys	Time, pressure, gyroscope, coordination, and size	MLP	14% FAR, 2.2% FRR
[26]	2014	315	Predefined free text (150 keystrokes)	Time, accelerometer, gyroscope, and oriented sensor	SVM	92% FAR, 1% FRR
[24]	2015	35	Predefined free text (3,000 keystrokes)	Time	Statistical method, KNN, Gaussian estimation, Parzen window kernel estimation, and support vector data description (SVDD)	8.99% EER
[12]	2020	50	10 samples of 200 keystrokes each	Time, acceleration, and coordination	TT, R, TTPR, TTMR, *Kolmogorov-Smirnov statistic*, and *Cramér-von Mises criterion* (*CM*)	0% EER
Our study	2021	45	200 keystrokes	Time, acceleration, gyroscope, pressure and coordination	ANN, KNN, SVM, Euclidean distance, Manhattan distance, and random forest	0% EER

Roh et al. [31] also used accelerometer sensors but added four additional features: keystamps, gyroscope sensor, touch size, and touch coordinates. Research on several other features, such as motion data and time interval, was conducted by Lee et al. [32], who evaluated keystroke dynamics features using both motion and motionless data to determine which feature yielded more accuracy—motion data yielded more precise results.

2.1.3. KDA Studies Based on the Classification Methods. Several techniques and methods have been proposed to categorize the typing behavior of users, such as machine learning techniques, distance-based matrices, and statistical mechanisms, which have been rarely implemented.

Many studies have employed distance matrices, such as the Manhattan, Euclidean, and Bhattacharya distances. For example, Lee et al. [32] and Coakley et al. [33] implemented Euclidean and Manhattan distance matrices, which resulted in the Manhattan distance producing better accuracy in both studies.

Some researchers have implemented data preprocessing through scaling [7] and standardization before applying distance matrices. Such preprocessing techniques might be crucial for calculating the similarities between features. Although these techniques have generated satisfactory results in some studies, other studies have deemed the results unacceptable.

Likewise, Lee et al. [32] applied the Manhattan and Euclidean distances with two different scaling techniques: standard and MinMax scaling. The best results were obtained by applying the Manhattan distance with standard scaling. Similarly, Roh et al. [31] used the Manhattan and Euclidean distances with mean, absolute deviation, and standard deviation. Their results suggested that preprocessing might not always be useful since they demonstrated that although the outcome generated by preprocessing was good for the average EER, it was worse for the best EER.

Machine learning techniques have been proposed in various studies, as shown in Table 1. Random forest, k-nearest neighbor (KNN), and multilayer perceptron (MLP) classifiers are the most frequently used machine learning techniques. In Sen and Muralidharan [34], MLP obtained a better accuracy rate than decision trees, naïve Bayes, and KNN. Random forest was considered better than MLP in Salem et al. [35], in which the EER score was 0.45%. In Kambourakis et al. [30], there are three renowned classifiers: MLP, KNN, and random forest. Eventually, MLP was rejected because it was incapable of running with memory restrictions when the classifiers were provided with an upper bound of 512 MB of memory. Moreover, KNN was used in Ehatisham-Ul-Haq et al. [36] in addition to three classifiers: support vector machine (SVM), decision trees, and Bayes net—the Bayes net and SVM classifiers delivered better results.

de Mendizabal-Vazquez et al. [37] used Euclidean distance alongside the MLP classifier, with MLP delivering better performance when the sample was increased and when the correct identification rate was 90%. However, the Euclidean distance performed well despite operating with a smaller sample, reaching an EER of 20%.

Based on the preprocessing using machine learning techniques, three preprocessed sample groups were created by De et al. [37]: (1) a linear discriminant analysis (LDA) group, (2) a principal component analysis (PCA) group, and (3) an original data group. The best results were obtained for the PCA group. It was also argued that a considerable reduction in data size due to PCA eased the implementation of these methods on mobile devices, which tend to have strong limitations because of their processing capacity and battery life.

2.2. KDA Using Other Languages. All previous keystroke dynamics studies conducted on touchscreen-based devices included only English as the input language, excluding one

Table 2: Recent studies of the fixed KDA for touchscreen devices.

Study	Year	Number of subjects	Methodology	Features	Classifier	FAR, FRR, or EER
[14]	2013	152	17-digit passphrase 10 times	Time, pressure, size, and coordination	K-means	4.59% FRR 4.19% FAR
[32]	2018	22	6-digit PIN	Accelerometer, gravity, rotation, pressure, time, size, and coordination	Euclidean and Manhattan distances	7.89% EER
[29]	2014	80	4-digit PIN/8-digit PIN	Time, acceleration, pressure, and size	Nearest neighbor distance	3.65% EER
[31]	2016	15	4-digit PIN	Flight time, acceleration, pressure, and size	SVR, scaled Euclidean, scaled Manhattan, KNN, and random forest	8.71% EER
[30]	2014	20	10 alphanumeric characters and 47 characters including spaces	Hold time, intertime, distance, and speed	Random forest, KNN, and MLP	12.5% EER
[27]	2015	12	4-digit PIN	Time, accelerometer, gravity, magnetometer, gyroscope, and orientation	Binary classifiers, Bayes net, and random forest	0.01% FAR 0.01% FRR
[7]	2019	104	4-digit PIN	Flight time, acceleration, pressure, and size	SVR, scaled Euclidean, scaled Manhattan, KNN, and random forest	8.71% EER
[6]	2016	150	4-digit PIN 16-digit PIN	Time and size	Gaussian estimation, z-score, and standard deviation drift	6.26% EER
[35]	2019	7	Static, 8 characters (complex passwords)	Time, pressure, size, and coordination	MLP, decision trees, and random forest	0.45% EER
[36]	2017	10	10 different password templates	Time, accelerometer, gyroscope, and magnetometer	Decision tree, KNN, SVM, and Bayesian network/Bayes net classifier	99.18% accuracy
[33]	2016	52	10-digit PIN	Time, pressure, screen location, accelerometer, and gyroscope	Euclidean distance and Manhattan distance	4.3% EER
[34]	2014	10	4-digit PIN	Time and pressure	Decision tree, naïve Bayes, KNN, and MLP	14.1% FAR 14% FRR
[37]	2014	80	4-digit PIN	Time, accelerometer, gyroscope, pressure, and finger size	Euclidean distance and MLP	20% EER

study that included both Korean and English [12]. This demonstrates a lack of language variation in such systems. However, some experiments have considered other languages in fixed keyboard environments. The first study conducted using another input language was in Gunetti et al. [38], which used Italian. This study demonstrated that KDA works in languages other than English and produces accurate results. Two features were examined in this study: the digraph latency and keystroke duration. The researchers compared the samples typed using English and Italian, providing evidence that keystroke dynamics are useful even when the typing samples are written in different languages. Japanese was also checked for accuracy by Samura and Nishimura [39], who employed the keystroke timing for every single letter and combinations of two letters composed of consonant and vowel pairs in the text. This experiment was performed on 112 participants divided into three groups, depending on their typing skills. The findings included a recognition accuracy of nearly 100% in the group that could write more than 900 letters in five min.

For generalising the KDA scheme to other language, a study in [12] conducted an experiment using two languages: Korean language, which is the native language of the participants, and the English language. Their results showed that the accuracy was higher when the native language was used. Likewise, studies in [2–4] compared between two languages (Arabic and English) with the Arabic native speakers using fixed keyboard, and the accuracy was higher using Arabic language. Thus, we aim in this paper to analyze the effectiveness of touch KDA for Arabic language with the Arabic native speakers.

2.3. KDA Using Arabic. The first study to consider Arabic in the analysis of the accuracy of keystroke dynamics was performed by Alsultan et al. [16], who used the key pairing approach via an Arabic alphabet keyboard. This study classified every character pair based on its relationship and keyboard location. Five keystroke features were extracted from each key pair. Their findings were extended by Alsuhibany et al. [17], who combined three features: keystroke duration, keystroke latency, and di-graph duration through Euclidean distance classification. This study yielded accurate results retched to 0.1 EER. Moreover, Alsuhibany et al. [18] further broadened this research by applying Bhattacharya and Euclidean distance measures, and the results showed that the Bhattacharyya distance was more accurate for both Arabic and English inputs.

Tables 1 and 2 provide a comparison of the most recent studies on keystroke dynamics for a touchscreen environment. As shown in Table 1, our study is compared with other studies that applied the free text technique [12, 24–26]. The comparison is based on many factors, such as EER, the number of keystrokes used, features, and classifiers. In particular, a study in [12], that achieved the best results compared with other studies, is comparable with our study. Although the best accuracy rate was the same (0.0 EER) reached by both studies, the number of keystrokes in our study was noticeably less, which increased the usability of our system.

3. Methodology

This section explains the typical touch dynamics biometric authentication system and its components. Figure 1 indicates that the system operation largely consists of several functional blocks (architectural components), each performing a well-defined function. These components and their respective operations are described as follows.

3.1. Data Collection. For the data collection, a touch-stroke authentication system was implemented using an Android application, which records raw data when a user touches a key. Moreover, when a user writes text and touches the submission button, a user profile is generated and stored in the local database of the device. This profile comprises five features: time stamps, acceleration, gyroscope, pressure, and coordination. Using an Android device (i.e., Huawei Nova 3i), data from 45 participants were collected. Most participants were in the same age bracket (19-25) and owned Android touch-technology smartphones.

It is important to note that all collected data was used for the pompous of this experiment and will be kept stored on the principal investigator drive with no names and an indemnifier of the participants.

3.2. Feature Extraction. This section describes the features used in the study. Specifically, touchscreen devices are designed to capture more features than traditional keyboards. Therefore, the features used in our study were selected because of their efficient performance in the state-of-the-art for activity recognition. To rephrase, the existing research [27, 31, 36] has well-established the excellent performance of these features for behavioral authentication.

3.2.1. Timing Features. The timing features of keystroke dynamics were attained from two keyboard actions: depression and release. Depression is the timestamp recorded when a key is held down (D), whereas Release is the timestamp recorded when the key is released (U). Timing features were obtained by capturing the time stamps for every event, as shown in Figure 2. Furthermore, very basic and consecutive events can exist in the following combinations:

(i) *Keystroke Duration or Hold Time (Down-Up)*. The key is pressed until it is released. Figure 3 shows the hold times for four randomly selected participants. The difference between users' behaviors when pressing the buttons can be seen, and the average hold time is, in most cases, more constant between users

(ii) *Keystroke Latencies/Flight Time*. This is also identified as Down-Down (DD) or Press-Press (PP)—the time between two consecutive key presses

(iii) *Di-Graph Duration*. This is the elapsed time between the release of the first key and the depression of the second key. It is known as the Up-Down (UD). Figure 4 shows the digraph duration for four users. Although there exists little difference between users, the rhythm of the digraph is less constant between users, as well as between the actions of individual users

3.2.2. Nontiming Features. Four nontiming features were used in our experiment: coordination, pressure, accelerometer, and gyroscope sensors.

The coordinate values are extracted for the horizontal and vertical axes at the time of the key press on the touchscreen device. These coordinate values are 2-raw data, one for the x-axis and another for the y-axis for each action. Figure 5 shows a scatter plot of five users who pressed one key when each user had different Cx and Cy coordinates.

The pressure force is returned when the user presses a key on the touchscreen. The returned pressure measurements are an abstract unit, ranging from 0 (no pressure) to 1 (normal pressure). However, higher than one values can also occur depending on the calibration of the input device. In essence, the pressure values are 1-raw data, which is the pressure force for each action. The accelerometer calculates the accelerometer (m/s^2) of the three axes, lateral x-, longitudinal y-, and vertical z-axis, as shown in Figure 6(a), by considering gravity values. Figure 7 shows the visualization of the values of the three axes of acceleration for two randomly chosen participants. The gyroscope measures the rate of rotation (rad/s) of a device using three axes: x – axis (pitch), y-axis (roll), and z-axis (yaw), as shown in Figure 6(b). The accelerometer and gyroscope numbers are uneven by samples and must be reshaped as regular forms. Lee et al. [2] used five formulae for the grouped data: average value (mean), root mean square, the sum of positive values, the

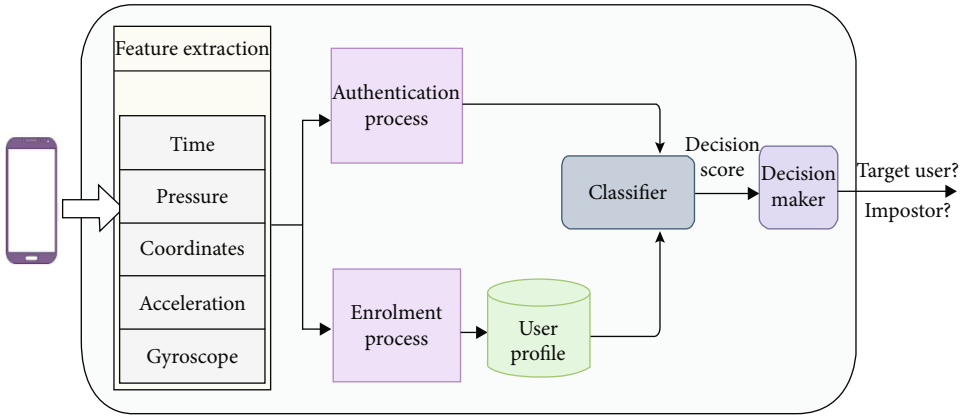

Figure 1: Typical touch dynamic biometric authentication system.

Description	Press "a"	Release "a"	Press "b"	Release "b"
Timestamp	000 (ms)	300 (ms)	400 (ms)	650 (ms)

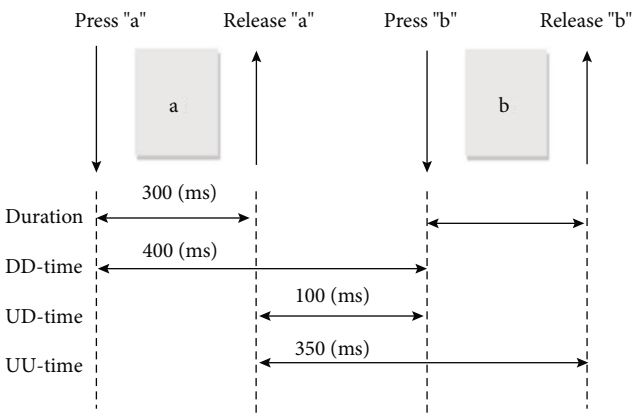

Figure 2: Timing features of KDA.

sum of negative values, and standard deviation. Since the error rate was improved when the "mean" formula of motion data was added, we employ the mean formula in our study.

3.3. Preprocessing. Each user has a distinct pattern when typing on a keyboard. However, a user may unintentionally deviate from his or her specific range of data by mistakenly performing an action that does not match the usual pattern. Therefore, outlier data for each participant were detected using an interquartile range. Then, these data were removed before feeding the classifiers to improve classification performance. For features that include more than one dimension, such as acceleration, gyroscope, and coordination, we remove the common value of outliers between these dimensions. For example, the record will be deleted if the three dimensions of acceleration have common outlier values.

Figure 8 graphically illustrates the boxplot for each feature and the removed outlier values.

3.4. Classification. After extracting the users' typing features and creating their profile templates, a classification process was undertaken to determine the similarities and differences between the users' templates. In particular, the standardized classifiers were used, including artificial neural network (ANN), KNN, SVM, Euclidean distance, Manhattan distance, and random forest, which were written using Python. Each of these classifiers is explained as follows.

An ANN is a series of algorithms that determine relationships within a dataset through a process that operates similarly to a human brain. Although there are several ANNs, our study utilized MLP owing to its high performance confirmed by recent studies [34].

Moreover, KNN estimates how likely a data point is to be a member of one group based on data points of which groups are nearest to it. SVM is a supervised machine learning algorithm used for both classification and regression. It aims to find a hyperplane in an N-dimensional space, where N is the number of features that distinctly classify the data points. Random forest unsystematically creates and merges multiple decision trees into one "forest." The goal is not to rely on a single learning model but instead on a collection of decision models for improving accuracy. The Euclidean distance involves calculating the distance between two n-dimensional vectors $p(p_1, p_2, \cdots, p_n)$ and $q(q_1, q_2, \cdots, q_n)$ in a straight line. Its formula is given by Equation (1):

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \quad (1)$$

The Manhattan distance calculates the distance between two n-dimensional vectors, $p(p_1, p_2, \cdots, p_n)$ and $q(q_1, q_2, \cdots, q_n)$, by subtracting the values and then summing their absolute values, as shown in Equation (2).

$$d(p, q) = \sum_{i=1}^{n} |q_i - p_i|. \quad (2)$$

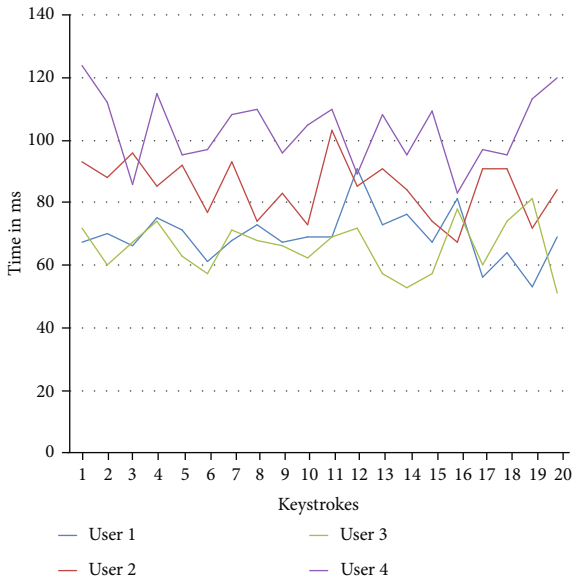

Figure 3: Hold time for four users.

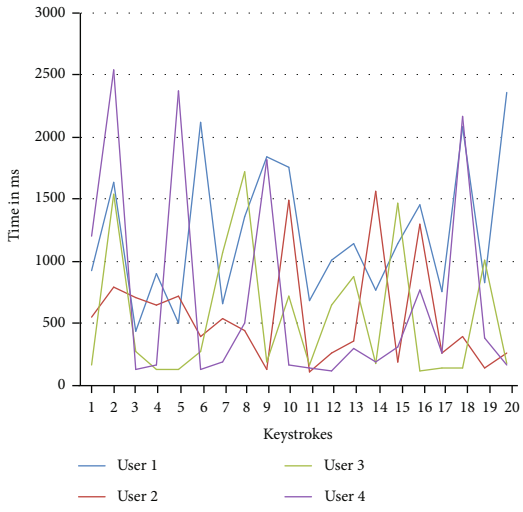

Figure 4: Di-graph duration for four users.

4. Evaluation

A controlled laboratory experiment was conducted in which the participants were asked to use the developed application. The following sections present the setup and procedure of the experiment.

4.1. Experimental Setup. This experiment involved various subjects as normal users. In the following section, the design of the experiment is provided, along with a description of the participants, the materials involved, and the systems in the experiment.

4.1.1. Experimental Design. The experiment was conducted in a controlled laboratory so that distributions made no interference, and the desired data could be collected without any biases. The experiment was divided into two sessions. In each session, the participants were required to enter one of the two keystroke authentication techniques, namely, fixed text or free text. Each session lasted 15 minutes, and there was a 10-minute break between sessions.

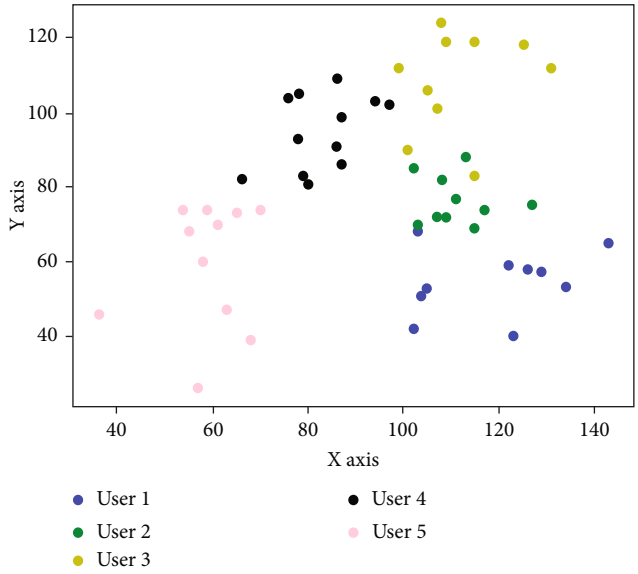

Figure 5: *Cx* and *Cy* of five users who press one key.

4.1.2. Participants. We recruited 45 participants, and the experiment was conducted for three weeks. The participants had varying typing skills and were between 19 and 25 years of age. All the participants were undergraduate students from different disciplines, and they were all native speakers. Most participants had a technical background. As demonstrated by Lee et al. [32], FAR is reduced in the case of the opposite gender for legitimate users. Therefore, to obtain adequate results, all participants in our experiment were women.

4.1.3. Materials. The stimulus material provided to the participants comprised two texts: a sign-up text and a log-in text. When the fixed text class was used, each participant entered the required phrase at least ten times for the sign-up phase and once for the log-in phase. One sample comprised 20 characters, as suggested by Lee et al. [32]. This study demonstrated that accuracy was the same in the first 20 actions. Subsequently, when the number of user actions was increased from 20 to 40, the user's input waiting time doubled with minor improvements in accuracy. In contrast, the sign-up text in the free text condition was 200 characters, whereas it was 198 characters in the log-in session. Although many studies have preferred to use a short free text [23, 40], it might not be enough to use only short texts to analyze keystrokes as they may not provide sufficient information for discriminating among users. Other studies [41, 42] have preferred a long free text. Huang et al. [41] argued that the reference profiles required 10,000 keystrokes, whereas a testing sample requires 1,000 keystrokes to produce satisfactory authentication performance. However, users might not find it convenient to enter longer texts. Therefore, an

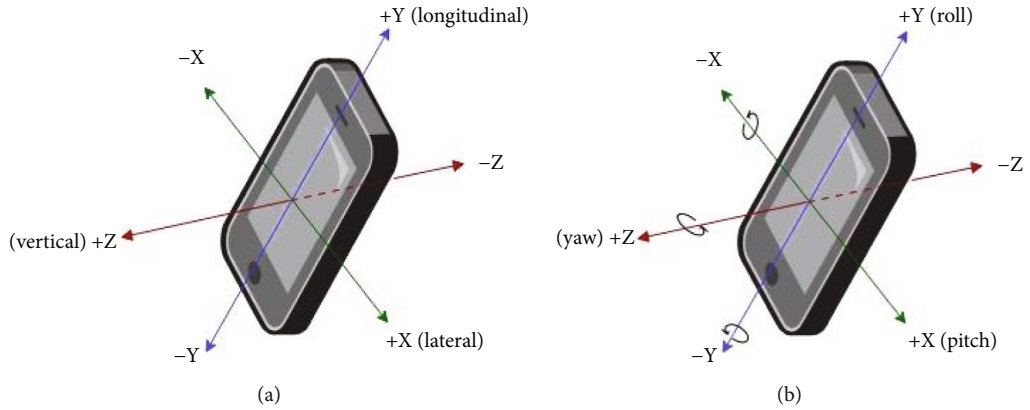

Figure 6: Axes of the motion sensors: (a) acceleration; (b) gyroscope.

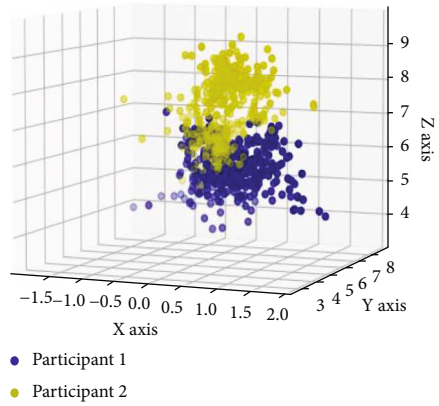

Figure 7: Scatter plot of three axes of acceleration data for two participants.

intermediate value was selected to determine the lengths of the characters. Table 3 shows the keystroke data collected during each session.

4.1.4. System. The proposed system was implemented on a Java Android application installed on an Android device. It was designed for users with varying technical experience. The application comprises the main page on which users can select one of the two keystroke biometric classes: free and fixed texts. Both classes are similar in terms of circumstance, which is as follows: after the user has selected the required technique, the page for entering the user's email address will be displayed. This page includes a text box in which the email address can be typed along with a button to check the validity of the email address. When the correct email address is entered, the sign-up page is shown, which has the required text to enter and the textbox to enter this text, as shown in Figure 9. When the user presses the submit button, the written text is matched with the required text; data features and the email address will be saved locally in the database. In contrast, when the text fails to match, an error page will be displayed, after which the user can make another attempt. After the sign-up phase, the user is directed to the log-in page.

Android does not have any mechanism to monitor the keyboard for security purposes, such as implementing a keylogger [30]. Thus, it was essential to design and implement a custom keyboard, which could be easily installed on each device used to authenticate users. This keyboard was developed for all touchscreen Android mobile devices.

4.1.5. Evaluation Metrics. Three metrics were used to evaluate the accuracy of the biometric authentication system: false rejection rate (FRR), false acceptance rate (FAR), and EER. FAR is the percentage ratio of the number of acceptances of an imposter user as a legitimate user. A low FAR indicates that fewer illegitimate users were falsely accepted, thereby indicating increased security.

FRR is the percentage ratio of rejecting a legitimate user by considering him/her to be an imposter. A low FRR indicates that fewer legitimate users were falsely rejected, thereby indicating the increased usability of the method.

EER is a single-number performance metric used to measure and compare the accuracy of various biometric systems. This metric is obtained by placing a graph, one for FAR and one for FRR, against a matching threshold, and then taking the interception point of the two graphs. The EER formula is given by Equation (3).

$$\text{EER} = \frac{\text{FAR} + \text{FRR}}{2}. \tag{3}$$

Usually, a low FRR and a low FAR result in a lower EER. A lower EER indicates a good performance using a biometric authentication method. However, because FAR and FRR are negatively correlated, it is impractical to lower both metrics.

4.2. Experimental Procedure. This section explains how the experiment was conducted—instructions for the participants, the experimental process, and the data collection procedure.

4.2.1. Instructions to Participants. The participants were initially instructed to type the provided text as normal. All participants were required to switch off their phones (or set them to silent) and avoid chatting with friends. This was done to prevent any interruption during typing. All participants were asked to sit while holding the smartphone in their hands, as this position yielded more accuracy, as stated

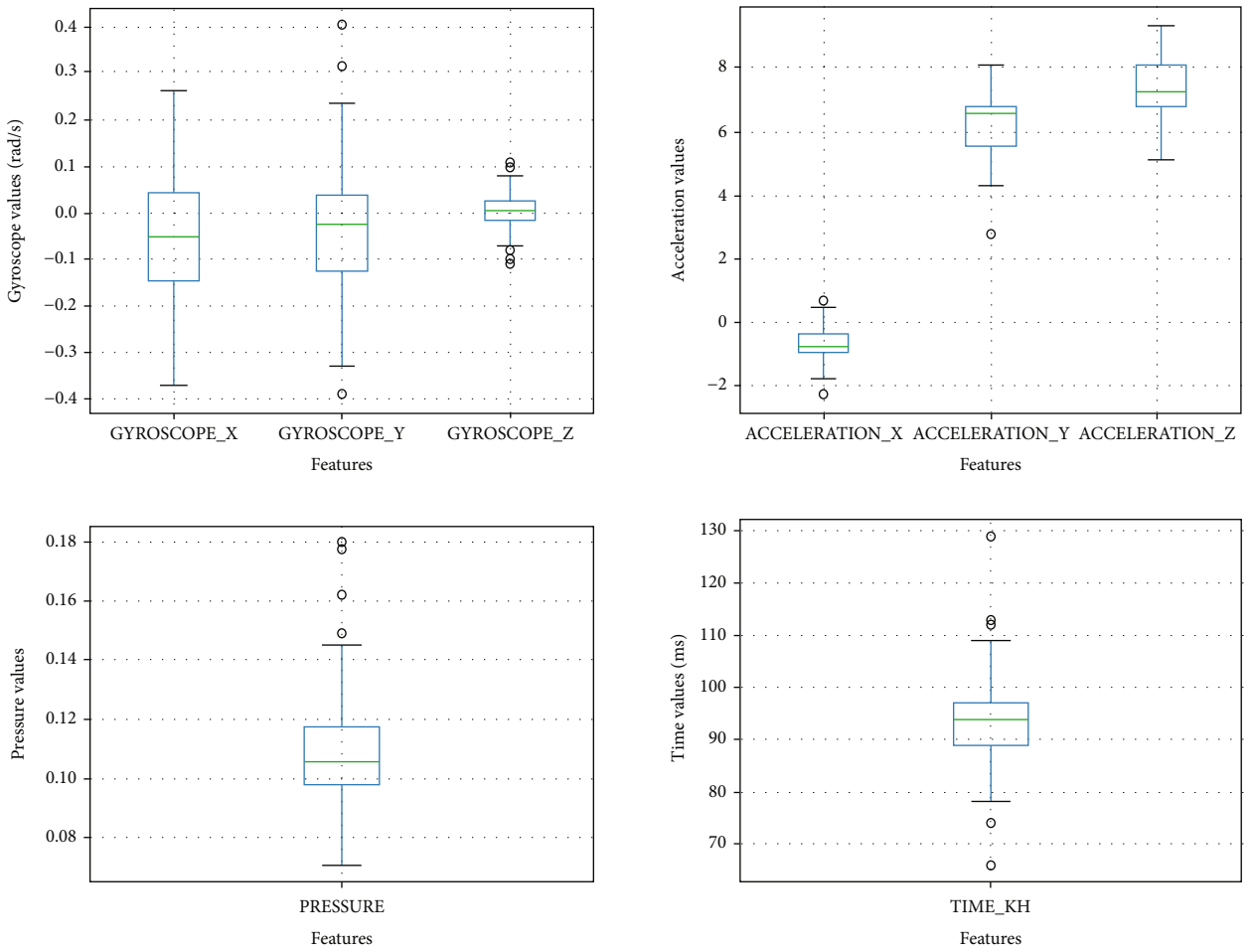

Figure 8: Boxplots of the feature data for one user.

Table 3: Experimental settings of collected keystroke data.

Session#	Technique type	Number of training samples	Number of testing samples	Number of actions in sign-up phase	Number of actions in log-in phase
First session	Free text	1	1	200	198
Second session	Fixed text	10	1	200	20

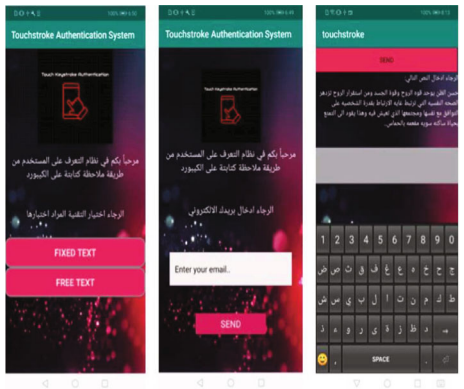

Figure 9: Interfaces of an Android-based keystroke collector.

in [31]. Before beginning the experiment, the participants were given some time to test the system by discarding the first trial. During typing, the participants were told that they could use the backspace or spacebar keys when needed. Lastly, a confirmation message was shown, indicating that the experiment has ended.

4.2.2. Procedure. The procedure was implemented in two distinct phases to authenticate users using KDA on mobile phones. The first phase of enrolment is also known as profile building. In this phase, the typing rhythm was collected in different trials to select the most similar profiles for the typing behavior of the user. For the second phase, the user was required to enter the log-in text, which was matched and compared with the stored text. Each time, the authenticated

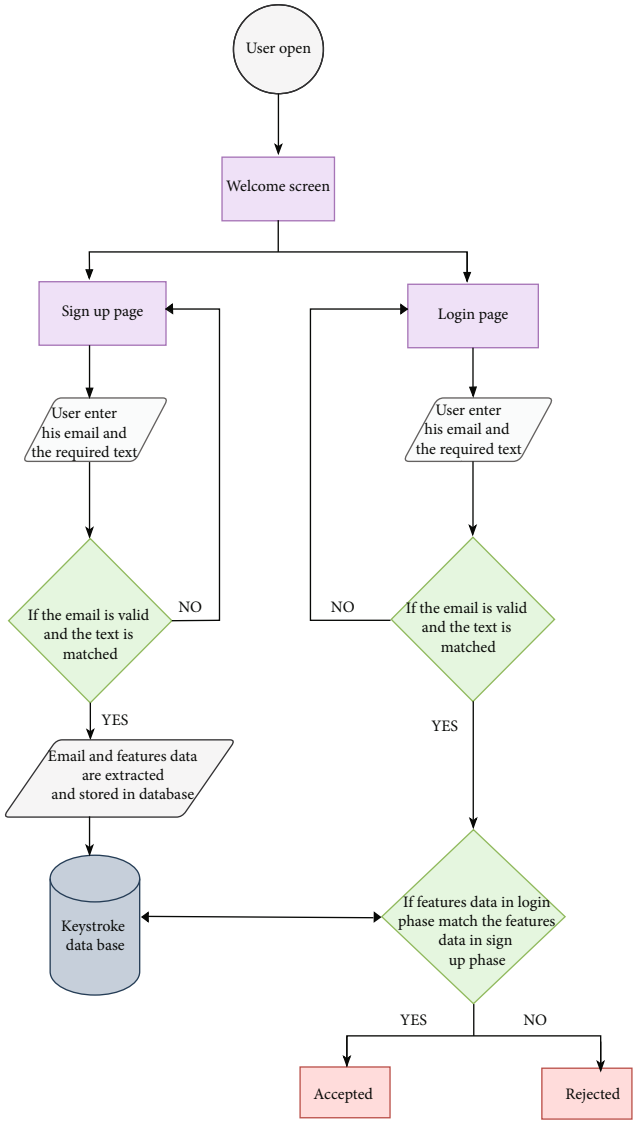

Figure 10: Flowchart of the proposed system.

user was required to enroll with the system for comparison with the profile stored in the database. A flowchart is shown in Figure 10.

4.2.3. Collected Data. For every successful attempt, a user profile was created, which included data for time keystrokes, accelerometers, gyroscopes, pressures, and coordinates. Table 4 shows all 13 features for each class; however, the number of dimensions is different. In particular, in a fixed text class, every sample comprises 20 sets of 13 features. Hence, by subtracting three values from DD, PP, and UD features, a single sample comprises 257 fields. Conversely, in the free text class, whenever a participant touched a particular key on the keyboard, it was presented as 13 features, where each feature appears in a single dimension in DB. Subsequently, the mean, maximum, and minimum for each feature are stored in another table.

Table 4: Features and dimensions for each KDA class.

Feature set	Description	Number of dimensions 1 sample (20 characters) (fixed text)	(Free text)
Time	Hold time	20	1
	Flight time (DD)	19	1
	Flight time (PP)	19	1
	Di-graph duration	19	1
Coordination	At TouchDown	40	2
Pressure	At TouchDown	20	1
Acceleration	At TouchDown	60	3
Gyroscope	At TouchDown	60	3
# of features		13	13
# of dimensions		257	13

5. Results

All participants successfully completed their tasks in three weeks. The level of accuracy obtained in the research indicates that this approach can improve the performance of touch-based systems when typing in Arabic. Specifically, the combination of the five features, namely, time, acceleration, coordination, pressure, and gyroscope, by applying a random forest classifier yields 0.0% EER using a free text database and 0.086% EER using a fixed text database, as shown in Figures 11 and 12. Although we have compared between our results and prior studies' results in Table 1, Table 5 compares between our approach and the result of [12]. This shows clearly that our approach is more usable due to the less number of keystrokes, though the accuracy rate of both studies was the same (0.0 EER).

This section presents the results, first with machine learning methods and then with distance-based metrics.

5.1. Machine Learning Methods. It is evident from Figures 11 and 12 that the free text class has a low FAR in comparison to the fixed text class. For example, the KNN classifier scored the FAR of 0% with the free class using a combination of features. However, in the fixed text class, the FAR was 0.34%. Notably, applying the coordination feature in the fixed text database yielded a better FRR than that in the free text database. In the fixed text classifier, the best result obtained was 0.18% FRR, whereas in the free text classifier, the FRR reached 0.6% using SVM.

5.2. Distance-Based Metrics. This experiment utilized two popular distance matrices: Manhattan and Euclidean distances. As suggested by Alsuhibany et al. [18], the standard deviation of the user's profile was utilized to set a threshold for the Euclidean distance. Therefore, we used this threshold for both Euclidean

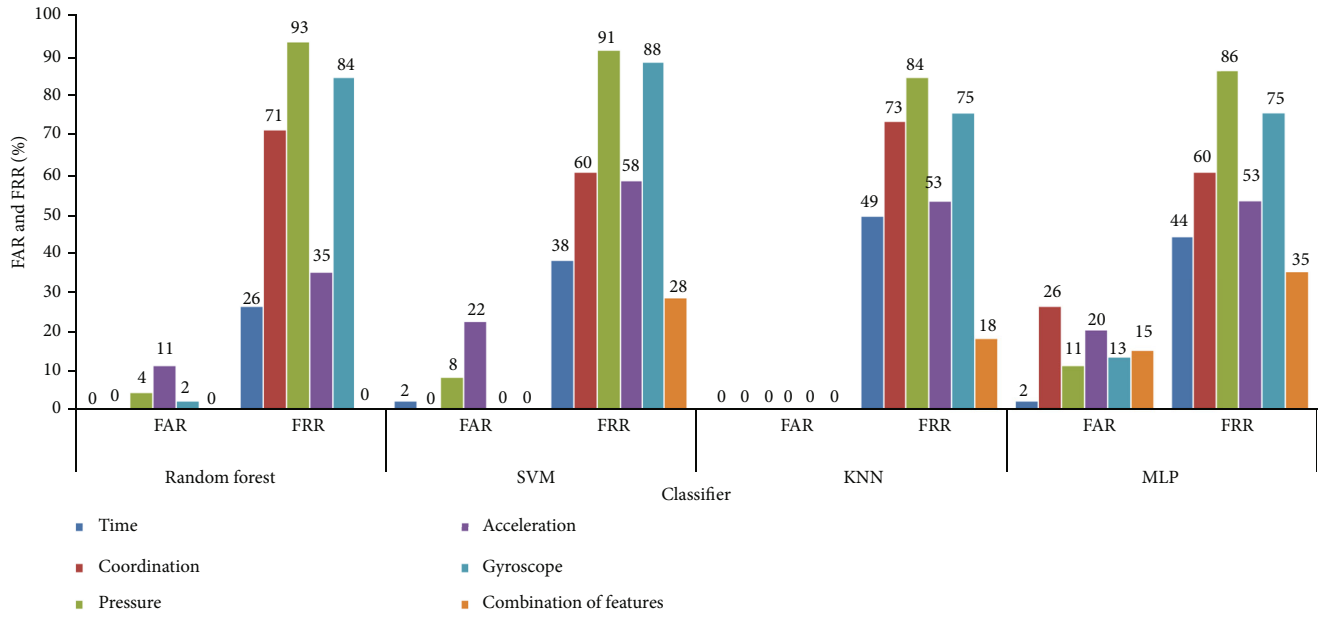

Figure 11: FAR and FRR for each supervised classifier using the free text class.

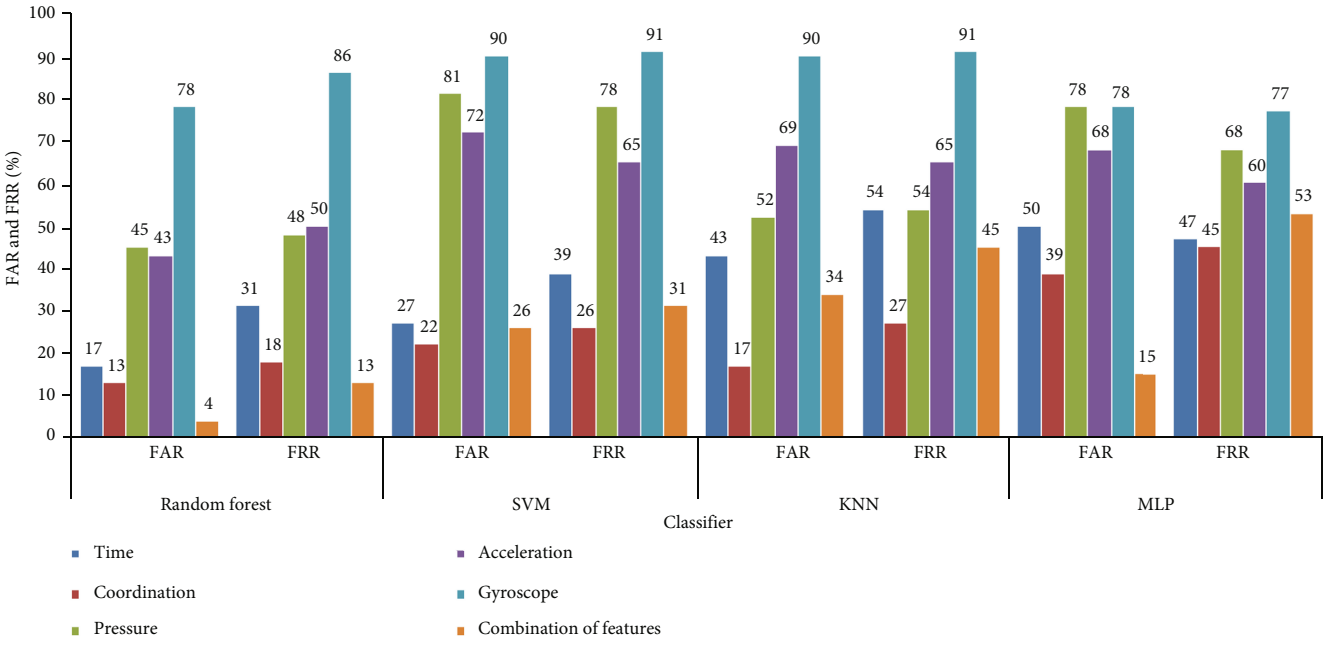

Figure 12: FAR and FRR for each supervised classifier using the fixed text class.

Table 5: A comparison of our system result and the result of [12].

Study	Methodology	Features	Classifier	EER
[12]	10 samples of 200 keystrokes each	Time, acceleration, and coordination	TT, R, TTPR, TTMR, Kolmogorov-Smirnov statistic, and Cramér-von Mises criterion (CM)	0.0 EER
Our study	200 keystrokes	Time, acceleration, gyroscope, pressure, and coordination	ANN, KNN, SVM, Euclidean distance, Manhattan distance, and random forest	0.0 EER

and Manhattan distances. To enhance the performance of the Manhattan distance, the threshold was changed to the summation of the standard deviations of the two users' profiles, which tends to engage in the process of authentication. This result is shown in Table 6, where it is evident that the performance of the Manhattan distance using the free text class has been

Table 6: Difference between two thresholds using the Manhattan distance in the free text class.

	Time		Coordination		Pressure		Acceleration		Gyroscope		Combination of features	
	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR
First threshold	10	15	80	5	30	0	10	40	0	90	30	35
Second threshold	0	15	80	0	10	0	25	0	15	10	10	15

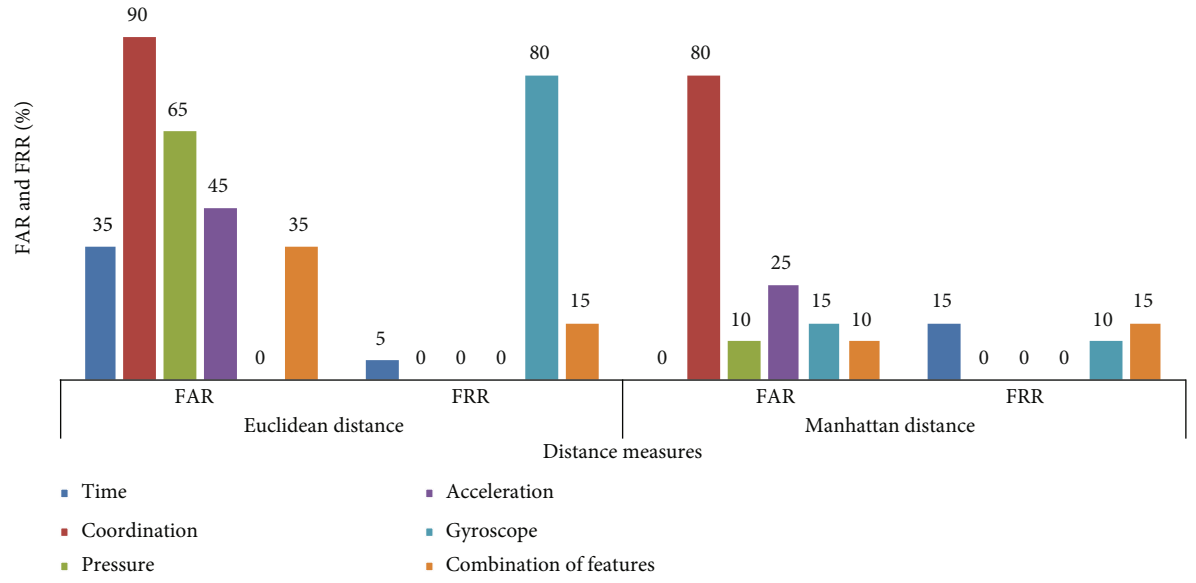

Figure 13: FAR and FRR for each distance-based metric using the free text class.

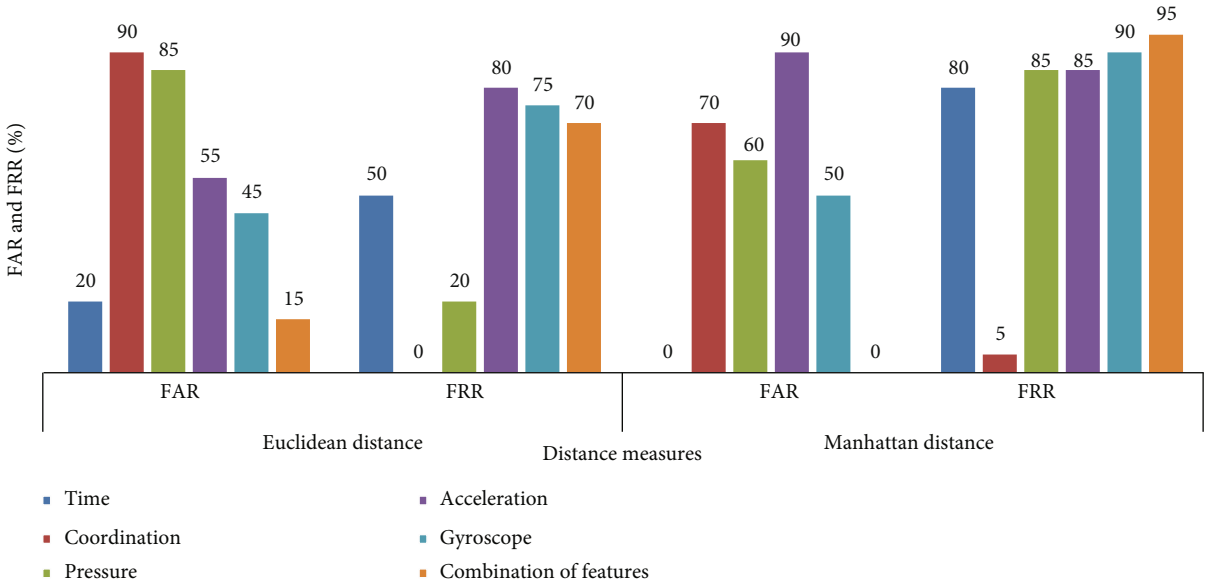

Figure 14: FAR and FRR for each distance-based metric using the fixed text class.

significantly increased using the second threshold. In fact, the results of the fixed text class remain unchanged even after setting the new threshold. The results of the two text classes were acceptable, as shown in Figures 13 and 14.

6. Discussion

This section interprets the results obtained using machine learning methods and distance-based metrics.

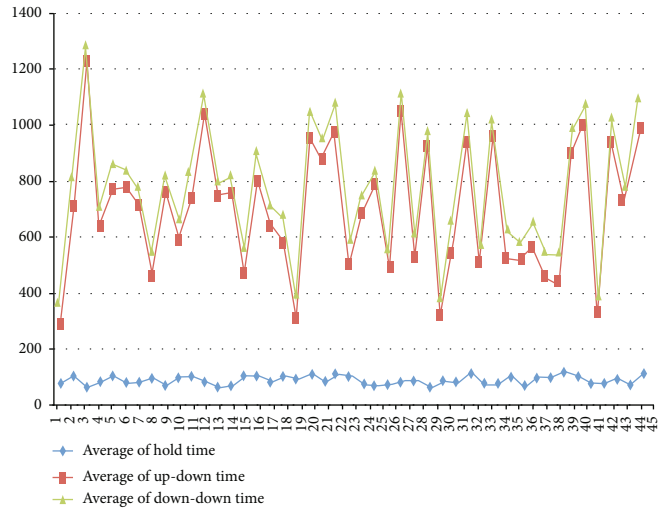

Figure 15: Average of the hold time, Up-Down time, and Down-Down time for each user.

6.1. Machine Learning Methods. This section compares the results based on three distinct points: the keystroke dynamic class, method of classification, and features. That is, a low FAR indicates a higher level of security. In this case, since the free text has a low FAR, it indicates that the free text is more secure than the fixed text. This may be attributed to the size of the text, which was provided in the log-in phase of the free text technique because it was longer than that in the fixed text technique. Consequently, it will be challenging for an intruder to imitate the typing pattern of the user for a longer text.

The obtained results are consistent with Alsultan and Warwick [43]—the free text ensures more safety and security than the fixed text from many threats, such as spyware, shoulder surfing, and social engineering. Hence, it can be concluded that as more keystrokes are input in the log-in phase, a more robust authentication model will be achieved. However, from the perspective of system users, inputting considerable keystrokes may be inconvenient.

In contrast, a lower FRR indicates increased usability. Since coordination achieved better FRR through fixed text, it ensures enhanced usability for the fixed text category, implying that every user has a particular coordinate for each button, as shown in Figure 5. Thus, this feature is rendered more convenient in the fixed text class.

Therefore, to determine the most appropriate machine learning method for the classification engine of the proposed system, the study undertook several preliminary classification experiments, which included four popular classifiers: SVM, MLP, k-NN, and random forest. As suggested by Kambourakis et al. [30], k-NN and random forest classifiers were most crucial in performance on mobile devices. This refers to the fact that a user will select not only the most effective algorithm in classification but also the one that promptly executes commands over the handheld device or smartphone, despite limited memory and CPU. In addition to the increased performance of the random forest classifier, it yielded higher accuracy as it outperformed the other three classifiers with the most features. In contrast, in addition to the less effective result produced by MLP, in Kambourakis et al. [30], the MLP classifier was excluded because it could not run on limited memory capacity. Thus, it can be inferred that random forest is the best option for the KDA system because it can run on low-memory devices.

Furthermore, Figure 11 (free text) clearly indicates that the combination of all features delivers the best results using all four classifiers. When each feature is used independently, time is the best feature, followed by acceleration. Figure 15 depicts the average of the hold-time, UP time, and DD time using the free text class, implying that the time features are unique for each person, except for a few users who can be distinguished by their other features. Figure 12 (fixed text) implies that combining all the features provides better results in two classifiers: random forest and MLP. In contrast, the coordination feature was the highlight feature in the case of SVM and KNN classifiers.

In general, it was observed that the combination of different features delivered highly accurate results in both fixed and free texts. Additionally, time was a crucial feature in the free text technique, whereas coordination was more effective in the fixed text technique.

6.2. Distance-Based Metrics. Two parameters that influence the accuracy of distance-based metrics are the number of keystrokes used in the testing phase and the threshold. In the fixed text technique, the classification is intended to determine whether a user is legitimate by computing the distance between the mean point of reference samples and the one sample used in the log-in phase, comprising merely 20 letters. Conversely, in the free text technique, the distance computes the difference between the mean point of the sign-up data and the mean point of the log-in data, comprising 198 letters. This is an explanation for the higher results obtained using the free text class. Therefore, it is suggested to increase the number of samples entered by the user in the log-in phase in the fixed text class to three samples [31]. When the threshold was altered in the Manhattan distance, there was an increase in performance using the free

text. Hence, notably, the new threshold for the Manhattan distance, the Manhattan distance outperformed the Euclidean distance.

7. Conclusion and Future Works

This research extended the previous studies on KDA that used Arabic with conventional keyboards by investigating KDA in Arabic on soft keyboards. Currently, touchscreen devices are embedded with sensors that can improve the performance of the system. This study extracted five features to determine the keystroke patterns of the users: accelerometer, time, touch coordinates, touch pressure, and gyroscope sensor. The performance of the features was assessed using six methods of validation: SVM, KNN, Euclidean distance, Manhattan distance, random forest, and neural network. The system was analyzed through two keystroke dynamic classes: free text and fixed text, for determining the most effective approach. Subsequently, the results of both techniques were compared.

The results indicate that a verification system using Arabic is possible with touchscreen devices and can enhance security. It exhibits a higher rate of accuracy using the free text class, with an average EER of 0.0%, whereas an average EER of 0.08% can be obtained by using the fixed text class when combining the features and the random forest classifier. Among both KDA classes, the free text class had a lower FAR throughout the entire study, irrespective of the feature set used, thereby implying that the free text is more secure than the fixed text.

To improve our results, the experiments will be carried out on a tablet device to investigate whether the size of the screen has any impact on authentication accuracy. Moreover, additional classifiers, which were not a part of this study, can be included, and advanced scenarios, features, and methodologies can be considered in the future. Lastly, the number and diversity of the participants should be increased in the future experiments to better assess the associated outcomes with this particular behavioral trait.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

References

[1] "Number of smartphone users in the U.S. 2010-2023," March 2021, https://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/.

[2] "Smartphone users in South Korea 2015-2025," March 2021, https://www.statista.com/statistics/467171/forecast-of-smartphone-users-in-south-korea/.

[3] "Cybersecurity statistics report," March 2021, https://preyproject.com/blog/en/cybersecurity-statistics/.

[4] "Development of Android malware worldwide 2016-2020," March 2021, https://www.statista.com/statistics/680705/global-android-malware-volume/.

[5] "Distribution of Android malware 2019," March 2021, https://www.statista.com/statistics/681006/share-of-android-types-of-malware/.

[6] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "TDAS: a touch dynamics based multi-factor authentication solution for mobile devices," *International Journal of Pervasive Computing and Communications*, vol. 12, no. 1, pp. 127–153, 2016.

[7] Y. Wang, C. Wu, K. Zheng, and X. Wang, "Improving reliability: user authentication on smartphones using keystroke biometrics," *IEEE Access*, vol. 7, 2019.

[8] J. Han, S. M. Kywe, Q. Yan et al., "Launching generic attacks on iOS with approved third-party applications," in *International Conference on Applied Cryptography and Network Security*, M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds., vol. 7954 of Lecture Notes in Computer Science, , pp. 272–289, Springer, 2013.

[9] O. Berkman and O. M. Ostrovsky, "The Unbearable Lightness of PIN Cracking," in *Financial Cryptography and Data Security*, pp. 224–238, Springer, 2007.

[10] C. Sousedik and C. Busch, "Presentation attack detection methods for fingerprint recognition systems: a survey," *IET Biometrics*, vol. 3, no. 4, pp. 219–233, 2014.

[11] N. Erdogmus and S. Marcel, "Spoofing in 2D face recognition with 3D masks and anti-spoofing with kinect," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–8, Arlington, VA, USA, 2013.

[12] J. Kim and P. Kang, "Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features," *Pattern Recognition*, vol. 108, 2020.

[13] M. El-Abed, M. Dafer, and C. Rosenberger, "RHU keystroke touchscreen benchmark," in *2018 international conference on Cyberworlds (CW)*, pp. 363–368, Singapore, 2018.

[14] M. Trojahn, F. Arndt, and F. Ortmeier, "Authentication with keystroke dynamics on touchscreen keypads-effect of different n-graph combinations," in *In 3rd International Conference on Mobile Services, Resources, and Users (MOBILITY)*, pp. 114–119, Lisbon, Portugal, 2013.

[15] P. Gautam and P. R. Dawadi, "Keystroke biometric system for touch screen text input on android devices optimization of equal error rate based on medians vector proximity," in *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 1–7, Malabe, Sri Lanka, 2017.

[16] A. Alsultan, K. Warwick, and H. Wei, "Free-text keystroke dynamics authentication for Arabic language," *IET Biometrics*, vol. 5, no. 3, pp. 164–169, 2016.

[17] S. A. Alsuhibany, M. Almushyti, N. Alghasham, and F. Alkhudier, "Analysis of free-text keystroke dynamics for Arabic language using Euclidean distance," in *2016 12th International Conference on Innovations in Information Technology (IIT)*, pp. 1–6, Al Ain, United Arab Emirates, 2016.

[18] S. A. Alsuhibany, M. Almushyti, N. Alghasham, and F. Alkhudier, "Investigating the effectiveness of Arabic

language for free-text keystroke dynamics authentication," *International Journal of Computer Science and Software Engineering*, vol. 6, 2017.

[19] A. Alsultan, K. Warwick, and H. Wei, "Non-conventional keystroke dynamics for user authentication," *Pattern Recognition Letters*, vol. 89, pp. 53–59, 2017.

[20] X. Lu, S. Zhang, P. Hui, and P. Lio, "Continuous authentication by free-text keystroke based on CNN and RNN," *Computers & Security*, vol. 96, 2020.

[21] B. Ayotte, J. Huang, M. K. Banavar, D. Hou, and S. Schuckers, "Fast continuous user authentication using distance metric fusion of free-text keystroke data," in *2019 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW);*, pp. 2380–2388, Long Beach, CA, USA, 2019.

[22] C.-J. Tsai and K.-J. Shih, "Mining a new biometrics to improve the accuracy of keystroke dynamics-based authentication system on free-text," *Applied Soft Computing*, vol. 80, pp. 125–137, 2019.

[23] A. A. Ahmed and I. Traore, "Biometric recognition based on free-text keystroke dynamics," *IEEE Transactions on Cybernetics*, vol. 44, no. 4, pp. 458–472, 2014.

[24] P. Kang and S. Cho, "Keystroke dynamics-based user authentication using long and free text strings from various input devices," *Information Sciences*, vol. 308, pp. 72–93, 2015.

[25] B. Draffin, J. Zhu, and J. Zhang, "Keysens: Passive User Authentication through Micro Behavior Modeling of Soft Keyboard Interaction," in *International Conference on Mobile Computing, Applications, and Services*, pp. 184–201, Springer, 2013.

[26] H. Gascon, S. Uellenbeck, C. Wolf, and K. Rieck, "Continuous authentication on mobile devices by analysis of typing motion behavior," in *Schutz und Zuverlässigkeit*, pp. 1–12, Gesellschaft für Informatik e.V., 2014.

[27] A. Buriro, B. Crispo, F. Del Frari, and K. Wrona, "Touchstroke: smartphone user authentication based on touch-typing biometrics," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pp. 27–34, Spinger, 2015.

[28] H. Saevanee and P. Bhattarakosol, "Authenticating user using keystroke dynamics and finger pressure," in *2009 6th IEEE Consumer Communications and Networking Conference*, pp. 1-2, Las Vegas, NV, USA, 2009.

[29] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: user verification on smartphones via tapping behaviors," in *2014 IEEE 22nd International Conference on Network Protocols*, pp. 221–232, Raleigh, NC, USA, 2014.

[30] G. Kambourakis, D. Damopoulos, D. Papamartzivanos, and E. Pavlidakis, "Introducing touchstroke: keystroke- based authentication system for smartphones," *Security and Communication Networks*, vol. 9, no. 6, 554 pages, 2016.

[31] J.-H. Roh, S.-H. Lee, and S. Kim, "Keystroke dynamics for authentication in smartphone," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1155–1159, Jeju, Korea (South), 2016.

[32] H. Lee, J. Y. Hwang, D. I. Kim, S. Lee, S.-H. Lee, and J. S. 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.

[33] M. J. Coakley, J. V. Monaco, and C. C. Tappert, "Keystroke biometric studies with short numeric input on smartphones," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–6, Niagara Falls, NY, USA, 2016.

[34] S. Sen and K. Muralidharan, "Putting 'pressure' on mobile authentication," in *2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, pp. 56–61, Singapore, 2014.

[35] A. Salem, A. Sharieh, A. Sleit, and R. Jabri, "Enhanced authentication system performance based on keystroke dynamics using classification algorithms," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 8, p. 8, 2019.

[36] M. Ehatisham-Ul-Haq, M. A. Azam, J. Loo et al., "Authentication of smartphone users based on activity recognition and mobile sensing," *Sensors*, vol. 17, no. 9, p. 2043, 2017.

[37] I. de Mendizabal-Vazquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Sanchez-Avila, "Supervised classification methods applied to keystroke dynamics through mobile devices," in *2014 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–6, Rome, Italy, 2014.

[38] D. Gunetti, C. Picardi, and G. Ruffo, "Keystroke Analysis of Different Languages: A Case Study," in *International Symposium on Intelligent Data Analysis*, pp. 133–144, Springer, 2005.

[39] T. Samura and H. Nishimura, "Keystroke timing analysis for individual identification in Japanese free text typing," in *2009 ICCAS-SICE*, pp. 3166–3170, Fukuoka, Japan, 2009.

[40] F. Monrose and A. Rubin, "Authentication via keystroke dynamics," in *CCS '97: Proceedings of the 4th ACM conference on Computer and communications security*, pp. 48–56, New York, NY, USA, 1997.

[41] J. Huang, D. Hou, S. Schuckers, and Z. Hou, "Effect of data size on performance of free-text keystroke authentication," in *IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015)*, pp. 1–7, Hong Kong, China, 2015.

[42] J. Huang, D. Hou, and S. Schuckers, "A practical evaluation of free-text keystroke dynamics," in *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pp. 1–8, New Delhi, India, 2017.

[43] A. Alsultan and K. Warwick, "Keystroke dynamics authentication: a survey of free-text methods," *International Journal of Computer Science Issues*, vol. 10, 2013.