A novel word-agnostic gesture-typing continuous authentication scheme for mobile devices

Max Smith-Creasey, Muttukrishnan Rajarajan

School of Mathematics, Computer Science and Engineering, City, University of London, London, United Kingdom

Abstract

In this study we produce a new continuous authentication scheme for gesture-typing on mobile devices. Our scheme is the first scheme that authenticates gesture-typing interactions in a word-agnostic format. The scheme relies on groupings of features extracted from the word gesture after it has been reduced to parts common to all gestures. We show that movement sensors are also important in differentiating between users. We describe the feature extraction processes and analyse our proposed feature set. The unique process of our authentication scheme is presented and described. We collect our own gesture typing dataset including data collected during sitting, standing and walking activities for realism. We test our features against state-of-the-art touch-screen interaction features and compare feature extraction times on real mobile devices. Our scheme authenticates users with an equal error rate of 3.58% for a single word-gesture. The equal error rate is reduced to 0.81% when 3 word-gestures are used to authenticate.

Keywords: continuous authentication, gesture-typing, biometrics, mobile security

1. Introduction

Mobile devices are one of the most popular technologies in the world. Until recently, authentication techniques used on mobile devices were based on a token of what the user knows, such as a PIN, password or gesture pattern. However, attacks to bypass the tokens (e.g.: via smudge attacks [3]) have been developed and weakened such approaches.

Most devices now implement a physiological biometric such as fingerprint scanning to authenticate the user. Nevertheless, such biometrics can be captured and replayed to gain device access [6]. Furthermore, a problem with all of these techniques is that they provide one-time authentication. Once a user has authenticated themselves the device makes no more authentication attempts and leaves the device open to impostor use.

To combat the weaknesses with the discussed authentication methods, researchers have developed continuous authentication (or active authentication) techniques. Such techniques continuously collect biometric data to build a profile of the user. Future samples of data from the device can be compared to this profile to produce a confidence score such that a decision may be made to permit access to the device.

Many schemes have been proposed to continuously authenticate users based on biometrics such as touch gestures [11], facial features [9] and combinations of different biometrics [26]. However, one of the most popular methods is to continuously authenticate is to use keyboard interactions [12]. Such studies use keystrokes because they occur frequently and stem from an existing body of similar PC-based schemes [4]. Whist there has been considerable research into keystroke analysis, there has been no extensive investigation into authenticating gesture typing interactions.

Gesture typing provides users an alternative way to input text via an on-screen keyboard by swiping their finger across letters to form a word. We find that, despite gesture typing existing on most mobile devices, there is no state-of-the-art continuous authentication scheme to ensure the user is genuine. The research in this area is currently minimal and lacks real-world practicality, e.g.: by requiring the scheme to be trained on a word before it can be authenticated [5]. In [23] the authors investigate gesture-typing but focus on word identification rather than providing an authentication scheme.

The main focus of this paper is to address the need we have identified for a scheme that robustly continuously authenticates gesture typing behaviour. We hypothesize that a novel set of features customised for gesture-typing can be extracted from word-gestures to identify users in a word-agnostic manner. We posit that this can yield better results than current state-of-the-art features. In addition to touch-features we also investigate features extracted from the movement sensor readings of the mobile device during gesture typing. Our investigation also considers different user activities during which gesture typing may be performed. Therefore, our contributions are threefold:

Email addresses: max.smith-creasey@city.ac.uk (Max Smith-Creasey), r.muttukrishnan@city.ac.uk (Muttukrishnan Rajarajan)

- A novel continuous authentication scheme that authenticates gesture-typing interactions. Unlike other schemes, we show that our scheme is word-agnostic and authenticates based on features only.

- We present a new feature set for gesture-typing interactions that incorporates unique aspects of gesture-typing such as redirections and pauses. We show our feature set better authenticates users compared to state-of-the-art features. We also show computational time requirements on real mobile devices.

- We test the effect of using gesture-typing data collected during three activities (sitting, standing and walking). We show and discuss the impact of activity on the different feature groups.

The remainder of this paper is organized as follows. Section 2 explores the work related to our scheme and describes the limitations of the research thus far. Section 3 presents the general concept and unique architecture for our scheme and describes the new dataset we have produced. Section 4 describes the novel feature groupings we create and use within our scheme. Section 5 discusses the experiments and results from our scheme to show its effectiveness. Section 6 concludes the research and discusses the potential future work that may be derived from our scheme.

2. Related Work

One of the most popular ways to continuously authenticate users on mobile devices is via keyboard interactions. Early research into the feasibility of using keyboard interactions was realised in [7]. The authors harnessed keystroke latency and hold-time characteristics to differentiate users. Their approach achieved equal error rates (EER) of 12.8%.

In [12] the authors present a sensor enhanced scheme using accelerometer and gyroscope sensor collected during keystrokes. The use of movement sensor data improved authentication accuracy. However, the scheme is limited in that it only authenticates known words (e.g.: passwords) and is therefore not continuous. In [24] the authors present a scheme for keystroke and tap based continuous authentication. Their scheme identified a unique set of features also based on accelerometer and gyroscope data. Their scheme produces an EER of 7.16%. The researchers in [8] collect touch-gesture features and accelerometer data during different activities (e.g.: sitting). The authors found accuracy increased when the training data and test data were from the same activity scenario. Accelerometers and gyroscope readings are used in [12] to aid fixed-text password authentication. Whilst based on keystroke dynamics, none of these schemes address gesture typing.

Researchers in [11] provide one of the first sets of generalised touch-gesture features for continuous authentication. They achieve EERs of 0-4% depending on the scenario but are tailored for swipe gestures, not the nuances of gesture typing. Similarly, in [20] a touch-gesture continuous authentication scheme is produced but limited to swipe-based gestures. In [2] and [1], touch-gestures are used to authenticate but gestures are limited to swipes in a non-continuous architecture. Authors of [10] benchmark their touch-gesture authentication scheme against the state-of-the-art by using multiple touch-gesture datasets. The focus of the study is on swipe gestures and uses features derived from earlier studies that are not tailored to gesture typing. Research into touch-gestures has focused largely on taps, keystrokes and swipes [27], indicating that gesture-typing has not be as comprehensively considered.

The first study to analyse gesture typing interactions as a biometric for continuous authentication was [5]. The study collected gesture typing data from 16 volunteers. Features derived from the well-known touch features proposed in [11] were extracted from sub-gestures within the word gestures. The study achieves good results when considering multiple words but suffers from instability for single words. The system trains Support Vector Machine (SVM) classifiers for each word. The real-world practicality of this is limited as it requires the system to know how the user performs each word-gesture prior to authentication.

Researchers in [23] evaluate the use of screen interrupts (context-switches) that occur within the Android OS during gesture typing. They find that pauses in typing gestures can be identified via the interrupt frequency. This information is shown to enable words and sentences to be recognised. The study also showed that authors of sentences could be identified given the interrupt signal. The work, however, was not intended to address continuous authentication and therefore is limited in this context.

We analyse the characteristics of the related work in Table 1. Primarily, the works are analysed in terms of whether they continuously authenticate, are applicable to gesture typing and whether they analyse the impact of user activity on the scheme. Furthermore, the areas of interest that features are derived and extracted from are also noted. From the analysed related work we find few feasible studies authenticating gesture-typing interactions as part of a robust continuous authentication scheme.

3. Proposed Approach

In this section we describe the practical reasoning and theoretical justification for our architecture, data collection process, feature extraction and verification processes.

3.1. General Idea

Our scheme builds on the observation that different words will be associated with different *word-gestures*. An example of this can be seen in Figure 1. The diverse differences in word-gestures for different words make them difficult to compare. Because we endeavour for our scheme to

Table 1: This table shows the characteristics and considered feature groupings used in other studies employing touch-interaction dynamics. Our work considers a wider variety of features applicable to gesture typing.

		Gesture Typing	Cont. Auth.	Activity Analysis	Feature Grouping					
					Whole-gesture	Sub-gesture	Redirect	Pause	Accel.	Gyro.
Study	Frank, et al. [11]		✓		✓					
	Burgbacher and Hinrichs [5]	✓	✓			✓				
	Simon, et al. [23]	✓						✓		
	Antal and Szabo [2]				✓					
	Alpar and Krejcar [1]				✓					
	Mondal and Bours [20]		✓		✓					
	Fierrez, et al. [10]		✓		✓					
	Kumar, et al. [19]		✓		✓				✓	
	Jain and Kanhangad [15]		✓		✓				✓	✓
	Sitova, et al. [24]		✓	✓	✓				✓	✓
	Giuffrida, et al. [12]				✓				✓	✓
	Crawford and Ahmadzadeh [8]		✓	✓	✓					✓
	This work	✓	✓	✓	✓	✓	✓	✓	✓	✓

be word-agnostic (such that we do not require a classifier for each word) we work on reducing each word-gesture to generalised features that can be compared across words.

In our scheme, when touch data is obtained from a word-gesture the gesture is processed to provide regions of interest for feature extraction. Points of redirection or pause are first identified in the gesture trajectory. The whole-gesture is then split the word into sub-gestures (similar to [5]) based on these points of redirection and pause positions. The resulting sub-gestures are typically smooth trajectories between letters.

The scheme also collects data from the accelerometer and gyroscope sensors. Data is recorded from each of the sensors during the gesture-typing interaction. We also record a set period before and after the interaction such that resistance and stability features identified in [24] may also be employed. We note that for battery preservation the movement sensors are only recorded when the keyboard is active.

Our scheme extracts features from the different identified areas of interest as *feature groupings*. This results in six feature groupings, namely (1) whole-gesture (2) sub-gesture (3) redirection (4) pause (5) accelerometer (6) gyroscope. These feature groupings and the extraction processes are discussed in more detail in Section 4.

A classifier is trained for each feature grouping based on data collected from the users. When a new word-gesture is entered into the device the features extracted for each feature group are classified by the relevant classifier. Each classifier outputs a score for the confidence it has that the user is genuine. Score fusion techniques are applied to the scores to combine the scores into a single score that can then be compared to a threshold.

If a feature grouping produces no score (e.g.: if a word-gesture contains no redirection) we use a missing data imputation technique in which we average the neighbouring scores to produce a simulated score. In cases where a feature grouping produces multiple scores (e.g.: multiple redirections) we average the scores.

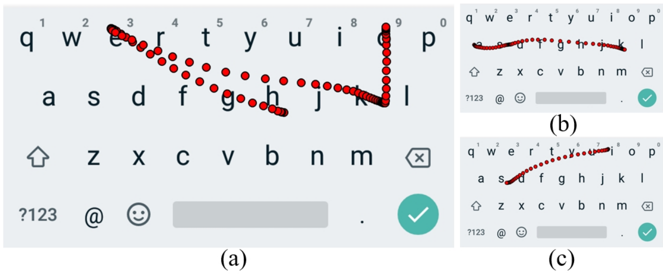

Figure 1: The coordinates captured as a user uses gesture typing to enter the words (a) "hello", (b) "ask" and (c) "is" onto the device. We note the diverse nature of gestures for different words.

3.2. Dataset

In this section we describe the custom gesture typing dataset we use in this study. We explain the data collection process and the type of data that is collected.

3.2.1. Data Capture

Because there is no publicly available gesture-typing dataset, we collect our own dataset. We design a custom data collection application for Android mobile devices. We build a mechanism that collects touch data as the user performs gesture-typing on the Android keyboard. Our application also records sensor readings from the accelerometer and gyroscope sensors every 20ms. We use a collection period of 20ms as it was shown in [24] that this sampling rate yields accuracy and efficiency. Data collected is written to local device storage.

We collected data from 20 volunteers. All volunteers were familiar with the concept of gesture typing and no constraints (other than to perform an activity) were placed on volunteers. Volunteers were provided a Nexus 4 device with our application on. Each volunteer participates in three activities: sitting, standing and walking. During each activity we ask the user to type words of their choice until they reach 100 words (our application informs them of this). To account for potential bias of data collected from one session we require each volunteer perform three sessions for each activity, resulting in 3 sessions for each of the 3 usage activities. In total each user typed approximately 900 words. Our dataset contains approximately 18,000 words.

3.2.2. Data Description

Our application stores the data collected in a raw form before pre-processing, feature extraction and scaling. Here, we briefly describe the data from each modality collected.

The data collected for a gesture-typing pattern is a collection of touch points containing the x and y coordinates of a point in the gesture pattern, the pressure at that point, the area the finger covers and the timestamp. Each time the finger moves on the screen a new touch point is recorded with this information.

Data samples from the accelerometer and gyroscope comprise of a tri-axis vector $\{x, y, z\}$ coupled with a timestamp. The accelerometer measures acceleration of the device in the x (lateral), y (longitudinal), and z (vertical) axes. The gyroscope measures orientation as x (pitch), y (roll) and (azimuth) z.

3.3. Classification

Our experiments test the proposed framework with three different types of classifiers. We use Logistic Regression (LR), Naive Bayes (NB) and Random Forest (RF) classifiers in our study. We use these classifiers because they have been successfully tested on touch gesture features before in [22] and [25]. Furthermore, they are easily available in the Weka machine learning library [13].

3.4. Normalisation

Often in systems with multiple classifiers for different types of biometrics or features, the scores produced are not homogeneous. They therefore do not necessarily map to the same score domain. We note that although our classifiers provide outputs in the range [0, 1] it is possible that scores occupy different ranges within this range. We therefore evaluate the following normalisation techniques:

- **tanh-estimator**: Given by the following where μ_G and σ_G are the mean and standard deviations of genuine scores.

$$s_i^{'} = \frac{1}{2}\left\{tanh(0.01 \times (\frac{scr_i - \mu_G}{\sigma_G})) + 1\right\} \quad (1)$$

- **min-max**: Converting a score s_i to a normalised score $s_i^{'}$ using the maximum and minimum scores is given by.

$$s_i^{'} = \frac{s_i - min}{max - min} \quad (2)$$

- **z-score**: The z-score normalisation technique is equated by the following equation.

$$s_i^{'} = \frac{s_i - \mu}{\sigma} \quad (3)$$

3.5. Biometric Fusion

Using data from each feature grouping to make a decision requires fusion such that a score representative of all feature groupings can be obtained. There are multiple techniques for fusing biometric data [21]. The techniques are at the (a) *feature level* at which features are fused (e.g.: into a single vector) (b) *score level* at which scores of multiple classifiers are combined (c) *decision level* at which decisions from each classifier is used to form a decision (e.g.: majority voting).

This study experiments with score-level information fusion. Score level fusion is the most common form of biometric fusion [14]. We choose this technique because of its proven successful application in previous biometric studies [16]. There are several techniques for combining classifier scores [18], in this study we explore the following (where s_i is a score from a classifier i):

- **Sum Rule**: The sum rule is total of all scores from all classifiers.

$$score = \sum_{i=1}^{n} s_i \quad (4)$$

- **Product Rule**: In the product rule all scores are multiplied together.

$$score = \prod_{i=1}^{n} s_i \quad (5)$$

- **Minimum Rule**: The minimum rule is the minimum score of all of the scores.

$$score = min(s_1, s_2, s_3, ..., s_n) \quad (6)$$

- **Maximum Rule**: The maximum rule is the maximum score of all of the scores.

$$score = max(s_1, s_2, s_3, ..., s_n) \quad (7)$$

3.6. Feature Evaluation Methods

We find the correlation between the features via Pearson's correlation coefficient. This coefficient, r, is given for features x and y when we have n samples of each by the Equation in 8. We note $\bar{x}$ and σ_x are the mean and standard deviation of x, respectively.

$$r = r_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right) \quad (8)$$

The closer the score of r to -1 or 1, the higher the correlation between features x and y.

3.7. Accuracy Metrics

We assess our system by using the following metrics:

- **False Acceptance Rate (FAR)**: This is the rate that an impostor is wrongly classified as the genuine user.
- **False Rejection Rate (FRR)**: This is the rate that the genuine user is wrongly classified as an impostor.
- **Equal Error Rate (EER)**: This is the rate at which FAR and FRR are equal to each other. FAR and FRR sets are usually obtained as an acceptance threshold is adjusted. FAR and FRR pairs are correlated such that if one increases the other decreases. In our experiments EER=(FAR+FRR)/2 for the FAR and FRR with the smallest difference.
- **Receiver Operating Characteristic (ROC)**: This plots a curve that is used to assess the performance of a binary classifier system. ROC uses the axis of true positive rate and false positive rate for different acceptance thresholds.

4. Feature Description

In this section we describe the process and reasoning we undertake to produce novel and distinguishing features from raw data collected from mobile devices.

We consider our feature set based on their compatibility with the requirements outlined for effective biometric features in [17]. Most importantly, our features are designed to be (a) *universal*: must obtainable from all individuals (b) *distinctive*: must be able to separate individuals based on the feature (c) *permanence*: the feature must be sufficiently invariant (d) *collective*: must be able to measure the feature in a quantitative manner. Our features are also designed for the practicality requirements so therefore our features must also have the properties of (e) *performance*: must be able to be collected efficiently whilst maintaining accuracy (f) *acceptability*: the volunteers are comfortable with the data being collected (g) *circumvention*: must be difficult to forge.

4.1. Touch-based Features

In our dataset we observe that the word-gesture patterns performed have little similarity between them (e.g.: the words "hello", "is" and "ask" as can be seen as gestures Figure 1). Furthermore, the tolerance of gesture typing keyboards is strong enough such that a word can be accurately predicted despite many different ways in which the pattern forming the word is performed. Extracting features from gestures representing full words can therefore yield very few features due to the lack of consistency between them. As discussed in Section 3.1, we propose different feature groupings extracted after pre-processing. We note that gestures of length $\leq$8 are frequently found to be non-word interactions and are therefore omitted.

Table 2: Features extracted when a pause is detected in a gesture.

ID	Description
1	Time Paused
2	Pressure At Pause
3	Area At Pause
4	Relative Paused Time
5	Time Difference 3 Points Before Pause
6	Time Difference 3 Points After Pause
7	Relative Pressure During Pause
8	Relative Area During Pause
9	Pressure 3 Points Before Pause
10	Pressure 3 Points After Pause
11	Area 3 Points Before Pause
12	Area 3 Points After Pause

4.1.1. Pause Features

Most words input via gesture-typing, especially longer words, have pauses in. We attribute this to two reasons (1) the user may pause when they have reached the letter they desire (where there is not always also a redirection, e.g.: at "s" in "ask" in Figure 1) and (2) the user may pause at points where they are considering or identifying the location of the next letter in the word.

For these features we iterate over touch-points in the word gesture to compute the time difference between each touch-point. From point p_1 to p_n in a gesture made up of points $\{p_0, p_1, p_2, ..., p_n\}$ we compute the time difference between p_m and p_{m-1} as $p_{m-1_{time}} - p_{m_{time}}$. Time differences between the points are stored in a vector that will have a length of $n-1$.

We iterate over the time differences and identify pauses as values greater than a heuristically defined limit. This heuristic computation is based on whether the pause $p \geq (\omega \times \sigma) + \mu$, where σ is the standard deviation of the pause times, μ is the mean of the pause times and ω is a multiplier of 0.5. The value of 0.5 for ω is chosen because it results in pause features that yield the best accuracy when sensitivity parameters of 0.5, 1.0, 1.5 and 2.0 are tested.

Points of pause in the whole gesture are used to divide it into sub-gestures for feature extraction. Here, however, we describe the pause-based features that we extract from the points of pause occurrence. We propose 12 pause features in Table 2. We note a relative feature is the feature at the point of interest divided by the average of all other features of that type. Such novel features capture nuances of gesture typing.

4.1.2. Redirection Features

Most word-gestures involve a change in direction at the point where a user reaches a letter and moves their finger toward a new letter. Figure 1 shows this for the word "hello"; as the finger moves between some letters the direction of the line changes. Occasionally, we find that a redirect occurs within a seemingly straight line due to finger instability.

We compute points of redirection by iterating over the x and y points of the gesture. For a gesture of l touch points we consider vectors of three sequential points given

Table 3: The features extracted from an identified redirection point.

ID	Description
1	Area At Point of Redirect
2	Pressure At Point of Redirect
3	Sharpness Of Redirect
4	Pressure 3 Points Before Redirect
5	Pressure 3 Points After Redirect
6	Area 3 Points Before Redirect
7	Area 3 Points After Redirect
8	Relative Pressure at Redirect
9	Relative Area at Redirect
10	Time Finger At Redirect
11	Relative Time Finger At Redirect

by (x_{n+0}, y_{n+0}), (x_{n+1}, y_{n+1}) and (x_{n+2}, y_{n+2}) where $n \geq 0$ and $n \leq (l-2)$. This vector represents two lines within the full gesture. We compute the arctangent angle for each line. We compute the absolute difference between the two angles to provide a value between 0 and 2π.

The computed angles form a vector of length $l-2$ that can be used to heuristically detect significant direction deviation. A redirect is present when angle $\theta \geq (\omega \times \sigma) + \mu$, where σ is the standard deviation of the pause times, μ is the mean of the pause times and ω is a multiplier of 0.5. As with pauses features, the value of 0.5 for ω is chosen because it results in redirection features that yield the best accuracy when sensitivity parameters of 0.5, 1.0, 1.5 and 2.0 are tested.

The points of redirection are one of the metrics used to produce sub-gestures. We also extract features specifically from the redirection point because we hypothesise that redirection behaviour is unique to individuals. Redirection features are shown in Table 3.

4.1.3. Whole Word-gesture Features

These features are generic and extracted from the whole word-gesture. The features require no pre-processing and are not split into sub-gestures. Such features are particularly useful in classifying shorter words with no redirection or pause, e.g.: the word "is" in Figure 1 is a simple straight line. We display the gesture features we propose in Table 4.

The features selected are generic enough to be agnostic to the number of redirections and pauses. This allows us to use features identified in some of the previous literature that were constructed for simpler scrolling touch gestures.

4.1.4. Sub-gesture Features

Sub-gestures are computed by first dividing the whole word gesture based on points of redirection. These sub-gestures are then each further divided based on points of pause. This provides us with all sub-gestures such that we can then extract sub-gesture features. The features we extract from each sub-gesture are shown in Table 5.

4.2. Sensor Features

In addition to our features obtained from the touch-screen of a mobile device, we also collect features from

Table 4: Features extracted from whole-word gestures.

ID	Description
1	Start Pressure
2	End Pressure
3	Max. Pressure First 3 Points
4	Min. Pressure First 3 Points
5	Mean. Pressure First 3 Points
6	Start Area
7	End Area
8	Max. Area First 3 Points
9	Min. Area First 3 Points
10	Mean Area First 3 Points
11	Mean Area
12	Mean Pressure
13	Mean Velocity
14	Standard Deviation Area
15	Standard Deviation Pressure
16	Standard Deviation Velocity
17	Time Between This and Prev. Word-gesture
18	Fluidity (Frequency of Pauses)
19	Acceleration At 5th From First Point
20	Acceleration At 5th From Last Point

Table 5: Features extracted from sub-gestures.

ID	Description
1	Mean Velocity During
2	Min. Velocity During
3	Max. Velocity During
4	Standard Deviation of Velocity
5	Velocity at 50% of Gesture
6	Velocity at 25% of Gesture
7	Velocity at 75% of Gesture
8	Mean Pressure
9	Min. Pressure
10	Max. Pressure
11	Standard Deviation of Pressure
12	Pressure at 50% of Gesture
13	Pressure at 25% of Gesture
14	Pressure at 75% of Gesture
15	Mean Area
16	Min. Area
17	Max. Area
18	Standard Deviation of Area
19	Area at 50% of Gesture
20	Area at 25% of Gesture
21	Area at 75% of Gesture
22	Min. Acceleration
23	Max. Acceleration
24	Acceleration at 50% of Gesture
25	Acceleration at 25% of Gesture
26	Acceleration at 75% of Gesture
27	Max. Deviation from Start to End Line
28	Max. Angle in Gesture
29	Min. Angle in Gesture
30	Mean Angle in Gesture
31	Standard Deviation Angle in Gesture

Table 6: Features obtained from each dimension of a sensor (shown for X).

ID	Description
1	Mean X During
2	Standard Deviation X During
3	Standard Deviation X 100ms Before
4	Standard Deviation X 100ms After
5	Difference Mean X Before & Mean X After
6	Max. X During
7	Min. X During
8	Max. X During Relative To Mean X Before
9	Min. X During Relative To Mean X Before
10	Max. X During Relative To Mean X After
11	Min. X During Relative To Mean X After
12	Mean X 100ms Before
13	Mean X 100ms After

sensor data. Sensor features have been shown to enhance touch gesture features [15]. The authors in [24] explored the use of sensor data for keystroke and tap authentication. We apply their proven hypothesis to form our movement sensor feature set for gesture typing.

For each sensor we produce a new feature grouping. Each feature grouping here, however, contains the same features. Furthermore each feature is contained within a feature grouping 4 times representing the x, y, z and m (computed magnitude, given by $\sqrt{x^2 + y^2 + z^2}$) dimensions. We extract features from sensor readings before, during and after the gesture typing interaction. We use 100ms of readings before and after the gesture. We show the sensor features in Table 6.

4.3. Feature Analysis

We find that each feature holds the distinctive property when alone used in a Random Forest classifier. When trained on one feature alone for each user, each feature yields an EER $< 50\%$ indicating it yields information about the user.

The results of Pearson's correlation coefficient convey little correlation between features. We find 79.46% of correlation results are between -0.5 and 0.5 and 94.89% of correlation results lie between -0.75 and 0.75. These results indicate that most features are not strongly correlated.

5. Experimental Results and Discussion

In this section we perform experiments on our continuous gesture-typing authentication scheme under different conditions. Features that form our samples are extracted following the techniques in Section 4.

5.1. General Accuracy

We first compute the EERs of each feature grouping exclusively. We perform this experiment with each of the three classifiers discussed in Section 3.3 such that we may identify the most appropriate classifier. For each user, the training set consists of 2 sessions from each activity and the remaining 1 session from each activity forms the test set. This gives 600 training samples and 300 test samples per user. For each user, classifiers are trained positively on the genuine training set and negateively on the impostor training sets. We use 3-fold cross-validation to verify our results. We record all scores to compute the FAR and FRR at any threshold. This is used to compute EERs for each feature grouping. We show these EERs from different classifiers in Table 7. We find the Random Forest classifier performs best in all cases.

Table 7: EERs for different features groupings and classifiers. LR, NB and RF are Logistic Regression, Naive Bayes and Random Forest, respectively.

	Classifier		
Feature Grouping	LR (EER%)	NB (EER%)	RF (ERR%)
Redirect	33.93	32.75	22.56
Pause	31.96	32.36	19.03
Sub-gesture	29.56	37.08	20.04
Whole-gesture	23.06	31.14	15.08
Accelerometer	19.03	42.92	12.46
Gyroscope	21.13	43.55	16.09

Next, we perform an experiment to identify the best combination for score normalisation and fusion for gesture typing features. We repeat the previous experiment but add the additional step of normalising and then fusing the scores produced from classifiers. We note that this experiment uses the Random Forest classifier due to its superior performance in the previous experiment. We use the normalisation techniques discussed in Section 3.4 and the fusion techniques discussed in Section 3.5. We show the EERs for this experiment in Table 8. Sum score yields the lowest EERs of all fusion methods. The lowest EERs for normalisation are obtained via the min-max method, though is comparable to no when normalisation is used. This appears to be due to the scores occupying almost the full range of [0-1], resulting in normalisation having little effect. We continue to use min-max normalisation in case a range does differ.

Table 8: The EERs when normalisation and fusion techniques are tested.

	Fusion Technique			
Normalisation	Sum (EER%)	Product (EER%)	Max (EER%)	Min (EER%)
min-max	3.58	21.42	8.58	8.03
z-score	3.60	40.97	9.37	18.42
tanh-estimator	5.43	5.45	10.74	11.15
None	3.58	21.42	8.58	8.03

5.2. Activity Comparison

In this experiment we test our scheme when evaluated on data from different activities. We recall the activities in our dataset are sitting, standing and walking. We train our scheme as we did in the previous experiment. This time, however, for each user, the scheme uses only 2 sessions from a specific single activity for training and 1 session

Table 9: The EERs are show for tests with different feature groupings whilst the activity the system is trained and tested on is varied.

	Activity			
Feature Grouping	Sitting (EER%)	Standing (EER%)	Walking (EER%)	All (EER%)
Redirect	23.76	22.33	22.13	24.56
Pause	22.06	20.61	19.96	22.16
Sub-gesture	21.27	20.37	18.36	21.71
Whole-gesture	15.88	15.24	14.58	16.85
Accelerometer	12.67	8.47	11.86	14.77
Gyroscope	13.58	13.44	17.67	18.18

Table 10: Cross comparison results when samples from different activities are used for training and testing. We note that $a \to b$ implies that the scheme was trained on session a and tested on session b.

Activity Session	**EER (%)**
All → All	4.76
Sitting → Sitting	3.32
Standing → Standing	3.91
Walking → Walking	4.08
Sitting → Standing	12.15
Sitting → Walking	17.24
Standing → Sitting	12.18
Standing → Walking	33.15
Walking → Sitting	18.29
Walking → Standing	12.65

from a specific activity for testing. For each experiment, this gives 200 training samples and 100 testing samples per user. Such that we can compare activity specific results with mixed activity results, we create mixed activity training and testing sets for users with an equal amount of samples evenly selected across all activities. In these experiments we use 3-fold cross-validation to verify our results. We use Random Forest classifiers and min-max normalisation with sum score fusion for combining scores.

We show the effects of features from different activities on different feature groupings in Table 9. We see that lower EERs are obtained when the training and testing sets are of the same activity rather than when trained on the mixed activity set. In Table 10 we show cross-comparison results of different activities tested on all feature groupings. Again, we see the best EERs occur when the training and testing sets are of the same activity. The worst results occur when training and testing sets are taken from different activities. The EERs are especially high for comparisons involving the walking activity, indicating gesture-typing behaviours during walking are significantly different to sitting and standing.

5.3. Comparison with State-of-the-Art Features

In this experiment we compare our proposed features with features from other state-of-the-art studies. We first compare our features with [11], in which the authors propose 30 features for touch-screen gestures. The features have since been used in many other touch-gesture authentication studies. We also compare our features to those used in [5], where a total of 10 features were derived from [11]

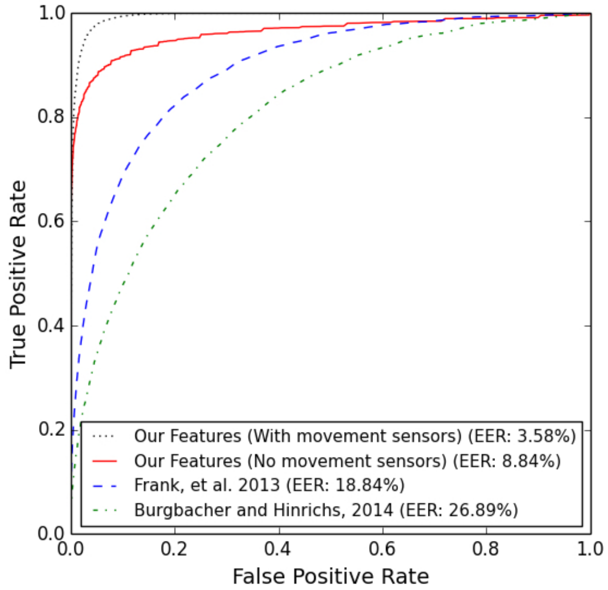

Figure 2: The ROC curve and resulting EERs for our proposed features compared to studies that produce their own touch-gesture feature sets.

for sub-gesture releavent features. The proposed feature descriptions can be found in their respective studies.

In this experiment we use the features presented in both studies and compare them to our proposed features. We use our scheme in Section 5.1 (such that all activities are used in training and testing) with Random Forest classifiers and min-max normalisation with sum-score fusion for combining scores. We extract the two sets of comparison touch-gesture features from identified word subgestures (as is done in [5]) such that the features can operate on straight sections of the gesture (as the features were designed for). We train and test the system on these features. Subgesture scores from each word are averaged to produce a final score. We then extract our own feature groupings and perform the same experiment. On our features, we perform tests with and without movement sensor features because the comparison studies did not include such features.

We show the results of this experiment as an ROC curve in Figure 2. We find our proposed features yield better results (both with and without movement sensor inclusion) when compared to the comparison features. We conclude that this is due to our features being custom designed for the nuances of gesture-typing. Furthermore, whilst we use more features than the comparison studies, our feature extraction process is not time intensive. We install the comparison experiment on a Nexus 4 device. We show average times taken to extract feature sets in Table 11. These results indicate that the extraction of our features is practical for continuous and efficient authentication.

Table 11: Average times (in milliseconds) to extract features from our scheme (with and without sensor data) and comparison schemes.

Our Feats. (with sensors) Time (ms)	Our Feats. (no sensors) Time (ms)	Burgbacher & Hinrichs, 2014 Time (ms)	Frank, et al. 2013 Time (ms)
3.13	1.90	1.40	1.41

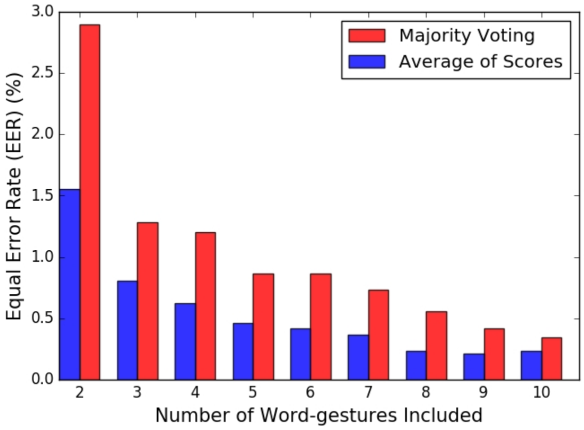

Figure 3: The graph represents the EERs when multiple word-gestures are combined with averaging and majority voting approaches.

5.4. Using Multiple Word-Gestures

In many continuous authentication systems the results of multiple authentication results are used to form a final decision because it is less volatile than a single result. We experiment with authenticating multiple word gestures because it is likely a user would input multiple words at a time (e.g.: when writing messages or emails).

We use the scheme in Section 5.1 with Random Forest classifiers and min-max normalisation with sum-score fusion for combining scores. We test multiple techniques for authenticating multiple word gestures. We first experiments with a simple average over multiple word-gesture scores which is compared to the threshold. Secondly, we experiment with using majority voting where the user is accepted if the majority of the scores surpass the threshold.

Figure 3 shows the EERs for both techniques using multiple words. We find that averaging the scores performs better than majority voting. Both techniques generally decrease the EER as the number of words increase. The lowest EERs from averaging and majority voting were 0.21% and 0.35%, respectively. We conclude that increasing the number of words in the authentication decision increases robustness at little additional cost; attackers would only be able to input several words before detected.

6. Conclusion

We presented a word-agnostic gesture-typing feature set and continuous authentication scheme. We describe our dataset and the creation process for our features and scheme. We showed our features yield results better than state-of-the-art features and the effect of training the scheme on data we collected during three different activities. We describe and discuss the superiority of our scheme over state-of-the-art approaches to touch-gesture and keyboard-based continuous authentication schemes.

In future work will improve our fusion techniques by weighting the contributions of different feature groupings. Secondly, we will build an activity recognition system such that gesture-typing samples can be classified by a classifier trained on the same activity, since we found this to be the most effective approach. Lastly, we will implement an anomaly detection scheme such that we may use fewer data samples in training by using only genuine user data.

Acknowledgment

This research work is carried out as part of a research studentship funded by British Telecommunications, UK.

References

[1] Alpar, O., Krejcar, O., 2015. Biometric swiping on touchscreens. In: Saeed, K., Homenda, W. (Eds.), Computer Information Systems and Industrial Management. Springer International Publishing, Cham, pp. 193–203.

[2] Antal, M., Szab, L. Z., 2016. Biometric authentication based on touchscreen swipe patterns. Procedia Technology 22, 862 – 869, 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania.

[3] Aviv, A. J., Gibson, K., Mossop, E., Blaze, M., Smith, J. M., 2010. Smudge attacks on smartphone touch screens. In: Proceedings of the 4th USENIX Conference on Offensive Technologies. WOOT'10. USENIX Association, Berkeley, CA, USA, pp. 1–7.

[4] Bours, P., 2012. Continuous keystroke dynamics: A different perspective towards biometric evaluation. Information Security Technical Report 17 (1), 36 – 43, human Factors and Biometrics.

[5] Burgbacher, U., Hinrichs, K., 2014. An implicit author verification system for text messages based on gesture typing biometrics. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '14. ACM, New York, NY, USA, pp. 2951–2954.

[6] Cao, K., Jain, A. K., 2016. Hacking mobile phones using 2d printed fingerprints. In: MSU Technical report.

[7] Clarke, N. L., Furnell, S. M., Jan 2007. Authenticating mobile phone users using keystroke analysis. International Journal of Information Security 6 (1), 1–14.

[8] Crawford, H., Ahmadzadeh, E., 2017. Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics. In: Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017). USENIX Association, Santa Clara, CA, pp. 163–173.

[9] Fathy, M. E., Patel, V. M., Chellappa, R., April 2015. Face-based active authentication on mobile devices. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1687–1691.

[10] Fierrez, J., Pozo, A., Martinez-Diaz, M., Galbally, J., Morales, A., Nov 2018. Benchmarking touchscreen biometrics for mobile authentication. IEEE Transactions on Information Forensics and Security 13 (11), 2720–2733.

[11] Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D., 1 2013. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. Information Forensics and Security, IEEE Transactions on 8 (1), 136 –148.

[12] Giuffrida, C., Majdanik, K., Conti, M., Bos, H., 2014. I Sensed It Was You: Authenticating Mobile Users with Sensor-Enhanced Keystroke Dynamics. Springer International Publishing, Cham, pp. 92–111.

[13] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., Nov. 2009. The weka data mining software: An update. SIGKDD Explor. Newsl. 11 (1), 10–18.

[14] He, M., Horng, S.-J., Fan, P., Run, R.-S., Chen, R.-J., Lai, J.-L., Khan, M. K., Sentosa, K. O., 2010. Performance evaluation of score level fusion in multimodal biometric systems. Pattern Recognition 43 (5), 1789 – 1800.

[15] Jain, A., Kanhangad, V., Dec. 2015. Exploring orientation and accelerometer sensor data for personal authentication in smartphones using touchscreen gestures. Pattern Recogn. Lett. 68 (P2), 351–360.

[16] Jain, A., Nandakumar, K., Ross, A., Dec. 2005. Score normalization in multimodal biometric systems. Pattern Recogn. 38 (12), 2270–2285.

[17] Jain, A. K., Ross, A., Prabhakar, S., Jan. 2004. An introduction to biometric recognition. IEEE Trans. Cir. and Sys. for Video Technol. 14 (1), 4–20.

[18] Kittler, J., Hatef, M., Duin, R. P. W., Matas, J., Mar. 1998. On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20 (3), 226–239.

[19] Kumar, R., Phoha, V. V., Serwadda, A., Sept 2016. Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns. In: 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS). pp. 1–8.

[20] Mondal, S., Bours, P., May 2015. Swipe gesture based continuous authentication for mobile devices. In: 2015 International Conference on Biometrics (ICB). pp. 458–465.

[21] Ross, A., Jain, A., Sep. 2003. Information fusion in biometrics. Pattern Recogn. Lett. 24 (13), 2115–2125.

[22] Serwadda, A., Phoha, V. V., Wang, Z., Sept 2013. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS). pp. 1–8.

[23] Simon, L., Xu, W., Anderson, R., 2016. Dont Interrupt Me While I Type: Inferring Text Entered Through Gesture Typing on Android Keyboards. In: 16th Privacy Enhancing Technologies Symposium (PETS).

[24] Sitová, Z., Šeděnka, J., Yang, Q., Peng, G., Zhou, G., Gasti, P., Balagani, K. S., 2016. Hmog: New behavioral biometric features for continuous authentication of smartphone users. IEEE Transactions on Information Forensics and Security 11 (5), 877–892.

[25] Smith-Creasey, M., Rajarajan, M., Dec 2016. A continuous user authentication scheme for mobile devices. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST). pp. 104–113.

[26] Smith-Creasey, M., Rajarajan, M., Aug 2017. Adaptive threshold scheme for touchscreen gesture continuous authentication using sensor trust. In: 2017 IEEE Trustcom/BigDataSE/ICESS. pp. 554–561.

[27] Teh, P. S., Zhang, N., Teoh, A. B. J., Chen, K., 2016. A survey on touch dynamics authentication in mobile devices. Computers & Security 59, 210 – 235.