Mobile Based Continuous Authentication  
Using Deep Features

Mario Parreño Centeno  
Newcastle University  
Newcastle upon Tyne, UK  
m.parreno-centeno1@ncl.ac.uk

Yu Guan  
Newcastle University  
Newcastle upon Tyne, UK  
yu.guan@ncl.ac.uk

Aad van Moorsel  
Newcastle University  
Newcastle upon Tyne, UK  
aad.vanmoorsel@ncl.ac.uk

ABSTRACT
Continuous authentication is a promising approach to validate the user’s identity during a work session, e.g., for mobile banking applications. Recently, it has been demonstrated that changes in the motion patterns of the user may help to note the unauthorised use of mobile devices. Several approaches have been proposed in this area but with relatively weak performance results. In this work, we propose an approach which uses a Siamese convolutional neural network to learn the signatures of the motion patterns from users and achieve a competitive verification accuracy up to 97.8%. We also find our algorithm is not very sensitive to sampling frequency and the length of the sequence.

KEYWORDS
Continuous authentication, Motion authentication, Biometrics, Learning latent representations, Siamese CNN

1 INTRODUCTION
Smartphones are used in an almost endless list of activities ranging from accessing social networks to performing banking transactions. In such tasks, sensitive and confidential information is processed by the device, and the same characteristics which prompted an increase in their popularity, i.e., portability and ease of use, are also causes of security concerns.

Common authentication mechanisms are passwords/PINs and biometrics-based methods i.e. fingerprint and face recognition. These approaches are one-time authentication methods, with the user identity validated only at the beginning of the session. Continuous authentication provides access control during the entire work session, constantly confirming the individual’s identity. This may be useful to manage the access right during the work session, for instance to avoid someone takes over your identity when you walk away from your device, to control the access to a specific content when sharing it with a friend or to skip further identification steps making services more user-friendly.

With the introduction of sensors in mobile devices, it has become plausible to use motion behaviour analytics as an authentication approach and its practicability has been studied. Because of the limited accuracy of the proposed approaches [8, 9], some studies suggest multi-modal solutions including motion analytics [10], at the expense that processing more sources of data leads to increase the computation burden.

On the other hand, advanced machine learning algorithms are widely applied in the field of biometric authentication. For example, Siamese neural networks have been successfully used in the context of face verification to learn a distance metric of the data space which can be used to compare previously-unseen classes i.e. images from people not seen during training [2].

Motivated by this, we propose a continuous authentication system, which detects unauthorised use on the smartphone, using the characteristic motion patterns of each individual interacting with the device. We use a Siamese convolutional neural network (CNN) which learns a distance metric from a large dataset, based on which we can extract deep features for new users authentication.

The authentication process is performed continuously during the whole session, without requiring any explicit action from the user. The proposed approach meets the intrinsic design requirements of the platform, and achieve an accuracy rate near to 98% using a simple classifier (one-class SVM), increasing the effectiveness of similar authentication approaches based on motion behavioural analytics.

To evaluate our approach we use a publicly available dataset of behavioural data on smartphones [12]. With observations from a group of users included in the set, we train a feature extraction model to learn a meaningful representation of the data. We verify the identity of new users using the deep features and a one-class support vector machine model. As previous studies in the area [8–10], we evaluate the accuracy of the approach utilizing testing datasets including samples from two people i.e. the legitimate and fraudulent users.

This paper is organised as follows. In Section 2, we introduce the biometric authentication systems and survey the traditional authentication methods. Section 3 reviews the continuous approaches for smartphones proposed in the literature. In Section 4, we explain the methodology of the proposed scheme. Section 5 introduces the
dataset used to evaluate our system and presents the effectiveness results. Section 6 concludes with a discussion.

2 AUTHENTICATION MECHANISM

An authentication mechanism is a common measure to enhance the security in a mobile device. In an authentication process, the identity of the user is verified according to different sources of information provided, directly or indirectly, by the user. The smartphone authentication processes are traditionally divided into two categories: knowledge and biometric-based methods. Knowledge-based authentication methods are based on information the user possesses, while biometric-based methods on information which describes physical or behavioural characteristics of the user.

Knowledge-based methods such as PINs and passwords have been used for mobile phone authentication since the inception of this technology in the market more than twenty years ago and they are still widely used, despite their intrinsic weaknesses have been largely demonstrated. Biometric-based methods are considered more reliable and secure, and the recent embedding of many new sensors in mobile devices permit additional opportunities to develop new systems. Most of the biometric approaches developed so far such as face, iris, periocular, fingerprint and voice recognition operate as one-time authentication methods, where the user is validated when requesting access to the platform or service, leaving the device vulnerable to unauthorized access during the rest of the user-session.

Continuous authentication methods have been proposed to solve that problem and the development of appropriate methodologies in this context is a prominent area of research. The few continuous biometric authentication approaches proposed so far, are based in the context of Machine Learning techniques, such as k-nearest neighbour, support vector machines and neural networks [8–10]. These methods allow for an automatic recognition of a potential fraudulent users but do not always achieve sufficiently effectiveness results.

2.1 Continuous biometric authentication system

Biometric data describe physical and/or behavioural characteristics of individuals. A Biometric Authentication System (BAS) captures and processes biometric data for the purpose of user verification.

Nowadays, many different sensors that are able to capture biometric data such as environmental, locational and user-specific motion information have been incorporated in the smartphone and the capture process is easy and straightforward. BAS are believed to be trustworthy, and their applicability in smartphone platforms is receiving increased consideration over recent years.

The practical implementation of BASs is hampered by engineering challenges in the data acquisition process, as well as methodological challenges in the development of efficient machine learning algorithm that could achieve a satisfactory effectiveness rate.

Similarly to knowledge-based methods, BASs such as face and voice recognition are one-time authentication systems: the user is validated only at the beginning of the session. BASs based on information such as motion or location can be however designed as a continuous authentication system, constantly verifying the user-identity throughout the session. The re-authentication is performed with a given frequency depending on the system capability. Continuous BASs should not be intrusive and transparent and do not require any attention or action from users.

Implicit authentication [4], can be used either as a primary authentication method or an auxiliary fraud indicator for higher assurance [7]. In typical use cases, continuous authentication adds extra reliability to the system and improves usability. For example, when the authentication system expresses continued confidence in the identity of the user, a service provider may decide to skip further security queries, i.e. not requiring extra info to complete a new transaction.

3 CONTINUOUS AUTHENTICATION APPROACHES FOR SMARTPHONES

Pioneer authentication methods for smartphones were based on touchscreen analytics [5]. In [8], it was proposed an approach based on the weighted k-nearest neighbour algorithm. Experiments in a controlled environment showed an EER of 3.5% when the re-authentication time was set up to ten minutes. Decreasing the re-authentication time gives considerably worse results, thus limiting the practicality of the approach on real scenarios being the average smartphone session duration 72 seconds, as a recent study suggests [1]. Additionally, an obvious limitation of touch recognition for continuous authentication is the requirement of continuous input from the user. The smartphone activity usage is very diverse, and some of the most popular activities involve few typing.

With the embedding of sensors such as an accelerometer, magnetometer and gyroscope, motion authentication for smartphones
became a subject of an increasing number of studies. Some of the recent literature is focused on identifying users based on their holding patterns [3]. Since motion data on smartphones can be continuously collected, a similar methodology can be used to implement continuous BASs, although the literature in this area is very sparse [8].

In [9], a continuous motion recognition system was proposed based on accelerometer and gyroscope data. The method is applied in two phases. In the first phase, they transform the observations in a new set of features and estimated their general distribution using a Gaussian mixture model. When a new user joins the scheme, they perform maximum a posteriori adaptation of the mean vectors of the general data model to build a client-specific model. Then, both models are used to produce a verification score for each new observation. The authors obtained an ERR of 18.2% in real-world scenarios.

In [10], the authors introduced a multi-modal approach including accelerometer, gyroscope and touch-screen observations. The authors use a one-class SVM model to classify the samples as either belonging to the owner or a guest/attacker. They test the approach with a dataset collected in a controlled environment where users were asked to type a text when sitting and when walking. They obtained an EER of 7.16% when the user was walking and 10.05% when the user was sitting. The authors deferred the evaluation of the approach on real-world scenarios to future studies.

4 PROPOSED APPROACH

Figure 1 shows the machine learning pipeline of our approach which is based on motion data from the accelerometer, gyroscope and magnetometer sensors. In an off-line phase, we use a Siamese neural network to learn a meaningful representation of motion patterns. Samples from users are transformed into the embedded space learned by the Siamese network. We use the deep features of each user i.e. the legitimate user, to train a one-class support vector machine model. New observations captured with the same device will be classified as belonging to the owner or to a different person i.e. the intruder.

4.1 Window sequencing

In our approach, when an individual joins the authentication scheme, accelerometer, gyroscope and magnetometer data is constantly captured by the smartphone when the user is utilizing the device.

For each channel, we use min-max normalization to scale the time series into the range between 0 and 1.

We split the stream of data sampled from the sensors into segments. We call instance to the vector which includes samples from space directional components for each of the sensors taking into account.

4.2 Feature extraction

Machine learning models learn from past data. In some cases, captured features may be irrelevant what may influence the effectiveness results of the approach. Finding a meaningful representation of the data may help to improve the accuracy rate of the system and reduce the computational burden.

Metric learning is a technique which pretends to learn a distance function from the input data that preserves the distance relation among the training samples. Our approach uses a Siamese CNN to learn an arbitrary transformation of the data based on a pairwise constraint.

4.2.1 Pairwise Constraints. Given a list of training samples \( \{x^{(i)}, y^{(i)}\} \) where \( x^{(i)} \) is the input feature vector and \( y^{(i)} \) a label which represents the class of the observation, common supervised feedforward networks are trained to minimize the prediction error i.e. the difference between the prediction of the model \( \hat{y}^{(i)} \) and the label \( y^{(i)} \).
Differently, a distance metric based model is trained based on similarity constraints between two or more input samples. The list of training samples is given on pairs and a label associated to the pair such that:

\[
\{(x^{(i)}, x^{(j)}, y^{(i)}) \mid i = 1, \ldots, m \text{ and } j = 1, \ldots, m\}
\]

where \(m\) equal to the number of training samples.

The label express the similarity between the pair of observations:

\[
y^{(i)} = \begin{cases} 
0 & \text{if } x^{(i)} \text{ and } x^{(j)} \text{ are similar}, \\
1 & \text{otherwise} 
\end{cases}
\]

The equivalence constraint states that a particular pair of samples are similar and should be close each other in the learned metric. On the other hand, the inequivalence constraint expresses that the pair of training samples are dissimilar and they should not be close to each other in the learned metric.

4.2.2 Siamese architecture. A Siamese network is a neural network which consists in two identical sub-networks. The objective of this approach is to find the relationship between two input samples. Fig. 2 shows the architecture of a Siamese network trained with a pairwise constraint. The two sub-networks share the parameters of the model (weights \(w\) and biases \(b\)). Thus, physically there is a unique sub-network which computes observations in different time steps. The parameters are updated based on the label associated to the pair.

One way to learn the parameters of the Siamese network which gives us a good encoding for the input vector is to define the pairwise loss function and apply gradient descent on it. For each pair of samples, the distance between the output vectors of the two networks is fed into the contrastive loss function. The contrastive loss penalizes small or large distances, depending on the similarity label \(y^{(i)}\).

If we call \(f_{(w,b)}(x^{(i)})\) the encoding of the input vector \(x^{(i)}\); and \(f_{(w,b)}(x^{(j)})\) the encoding of the input vector \(x^{(j)}\), we can define the parameterized distance function to be learned \(d_{(w,b)}(x^{(i)}, x^{(j)})\) such as the euclidean distance between the encoding vectors \(f_{(w,b)}(x^{(i)})\) and \(f_{(w,b)}(x^{(j)})\) [11] [6], such that:

\[
d_{(w,b)}(x^{(i)}, x^{(j)}) = ||f_{(w,b)}(x^{(i)}) - f_{(w,b)}(x^{(j)})||^2
\]

Being the loss function in its most general form[6]:

\[
\ell(w, b) = \sum_{i,j=1}^{m} L_{(w,b)}(x^{(i)}, x^{(j)}, y^{(i)})
\]

\[
L_{(w,b)} = (1 - y^{(i)}) \frac{1}{2} d_{(w,b)}^2 + (y^{(i)}) \frac{1}{2} \max(0, \alpha - d_{(w,b)})^2
\]

where \(\alpha > 0\) is called the margin.

5 EXPERIMENTS

In this section, we introduce the dataset used to evaluate our approach and we present the effectiveness results.

5.1 Dataset

To evaluate our approach, we use a publicly available dataset of behavioural data on smartphones [12]. The dataset contains samples from 100 volunteers, which includes multiple modalities: movement, orientation, touch, gesture and pausality. The data were collected under three task scenarios (reading, writing, and map navigation) and two body motion conditions (sitting and walking).

Between all the volunteers, 90 of them performed 24 sessions (8 reading sessions, 8 writing sessions, and 8 map navigation sessions), each session lasting about 15 minutes. We will use data from this group of 90 volunteers.

We split the group of 90 users into two subgroups, of 60 and 30 users. We used data from the first subgroup to train the feature extractor model. We used data from the second subgroup to train and test the verification model based on the one-class SVM model (the testing dataset includes samples not seen in the training phase).

To train the feature extraction model, we generate a training dataset combining samples from each of the 60 users. From each user, we sample data from 18 different sessions, 6 sessions for each task-motion condition activities (reading–sitting, reading–walking, writing–sitting, writing–walking, navigating–sitting and navigating–walking).
Based on the deep features, we train the one-class SVM model with observations from one of the 30 users -i.e. the legitimate user-and we evaluated the model with a testing dataset including samples from the same user and from one different user, i.e. the fraudulent user. Thus, we evaluate the ability of the approach classifying unseen observations as either, belonging to the owner of the device or not. The training dataset includes samples from 18 sessions, 6 sessions for each task-motion condition activity (from the legitimate user). The testing dataset includes observations from 6 sessions, one for each task-motion condition activity from each of the users (from the legitimate and fraudulent users). The observations from the legitimate user are different from those used to train the model. We repeat the same procedure for all possible pairs of legitimate-fraudulent users (in total, 870 scenarios), and we show the average of the results.

To train the feature learning model and the classification model, we use 6750 observations from each user taken into account, i.e. 270 instances when the window size is 1 second. The number of instances in scenarios with different window size, it will be proportionally to the resizing factor.

5.2 Results

To evaluate the effectiveness of our proposed approach, first we train a Siamese CNN with a pairwise constraint with observations from 60 users picked up randomly from the HMOG dataset [12], which represent the user population.

To train the model, we generate the same percentage of positive and negative pairs. We generate the pairs in the following way:

- positive pairs: both samples of the pair are from the same user being equal to the percentage of positive pairs for each of the users.
- negative pairs: we generate negative pairs matching samples from one user with another sample from a different user. We repeat the process and use the same number of negative pairs for each singular user.

The architecture of the Siamese CNN consists of 4 convolutional layers and 4 max-pooling layers connected alternatively. A 2D block of motion data of size window size by 9 channels (one channel for each directional axis of each sensor taken into account, i.e. accelerometer, gyroscope and magnetometer) is given to a convolutional layer with 32 filters of size $7 \times 7$. The configuration of the second and third convolutional layers are 64 filters of size $5 \times 5$ and 128 filters of size $3 \times 3$, respectively. We change the number of filters of the fourth convolutional layer with size $3 \times 3$ depending on the window size of the scenario. Thus, we can compare the accuracy rate for different sampling frequencies when the re-authentication time is the same. We reshape the output of the top layer of the network approximately to a 64-dimensional feature vector.

As noted, the resulting feature maps from the convolutional layers are fed to a max-pooling layer which takes the max over $2 \times 2$ spatial neighborhoods (for all four pooling layers).

To show the distribution of the learned features, we apply dimensionality reduction using principal component analysis to the extracted features of the testing dataset, including samples from the group of 30 users. Figure 3 shows the first two principal components. We can observe that samples of the same user are predominantly grouped together, and groups of observations are separate between them.

Then, using the Siamese model, we transform the observations of the training dataset of the group of 30 users and we train a one-class SVM model with a radial basis function kernel. We vary the gamma parameter (the inverse of the radius of influence of samples selected as support vectors) with values between 0 and 0.0001, and 0.0001 and 100, and we show the best accuracy rate obtained between all the results.

We have tested our approach with the following set ups:

- different sequence lengths: 0.5, 1 and 2 seconds.
- different sampling frequencies: 25 Hz and 100 Hz.

Furthermore, to show the importance of the feature extraction process, we also show the accuracy when the one-class SVM model is trained:

- with raw data.
- with the set of angles $\alpha(x, y, z)$, $\phi(x, y, z)$ and $\beta(x, y, z)$ describing the orientation of the three sensors (accelerometer, gyroscope and magnetometer) and their magnitudes $|a|$, $|g|$ and $|m|$ [9] (we call these features engineered features).
- with statistics extracted from each window: mean, median, standard deviation, variance, max, min, mean of the module,
Table 1: Confusion matrix of the approach when the feature extraction is done by a Siamese CNN with a sampling frequency of 25 Hz and window size of 1 second. We show the summation of the results of all the scenarios including different legitimate and fraudulent users.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>231683</td>
<td>3217</td>
</tr>
<tr>
<td>-1</td>
<td>7047</td>
<td>227853</td>
</tr>
</tbody>
</table>

mean of the set of angles for each axes of the three sensors. In total, each instance includes 66 features.

- with a 64-dimensional features vector extracted by a CNN (with the same architecture of the Siamese sub network).

Figure 4 shows the averaged effectiveness results for the different training datasets. We observe that the model trained with raw data has an accuracy rate of 86.9%. Observations are very noisy, and observations from the legitimate and fraudulent users are not separated in the data space; thus, the boundary which calculates the one-class SVM model cannot separate the group observations from both classes (we can see that false acceptance rate and false rejection rate are quite high). When we train the model with the group of features we call engineered (sets of angles describing the orientation and the magnitudes of the three sensors), the accuracy rate increases slightly to 87.5%, and it reached to 88.4% when we train the model with statistics (mean, median, variance, etc.) of the window. Experimental results suggest feature engineering improve moderately the performance accuracy.

On the other hand, the accuracy when we perform feature extraction using a conventional CNN is as high as 87.9%. However, we see that, when we transform the input data using the Siamese CNN model, accuracy results are significantly better. Using the same sampling frequency and window size, the accuracy rate increases to 96.3%. We can see within the figure that the false acceptance rate decreases significantly (from 7.3% to 3%). Thus, intuitively, the Siamese CNN model can learn representations of the observations from each of the users who are closer each other in the embedded space whatever is the activity which is performing the user.

On the other hand, we have evaluated the accuracy of the Siamese CNN model when using different sampling frequencies and window sizes. We can see that, when the sampling frequency is 100 Hz the accuracy rate decreases slightly when increasing the window size, from 96.3% when the window length is 0.5 seconds to 95.8% when it is 2 seconds. When the window size is bigger, the observations will include samples from a higher number of different movements which could generate representations of the observations which are more widely spread in the embedded data space.

When the input to the model has been sampled with a lower frequency, the accuracy rate is higher whatever is the window size. When the sampling frequency is 25 Hz, the accuracy is higher when the length of the window is 1 second and lower otherwise. This is the case with the highest accuracy (97.8%) between all the different scenarios (Table 1 shows the confusion matrix of the classification results). Nevertheless, these observations indicate that our method is robust regarding the sampling rate and window length, which can reduce the computational burden significantly.

6 CONCLUSION

We propose a continuous authentication biometric system for a smartphone which can detect fraudulent use by exploiting the user’s specific motion patterns. We have evaluated the effectiveness of the system with a comprehensive case study.

Throughout the experiments, we have shown that the learning feature extraction process using a Siamese CNN model improves the verification rate even using a simple novelty detection model (one-class SVM).

Furthermore, we have shown that, by adjusting the parameters of the scenario, sampling frequency and window size, we may be able to improve the effectiveness or efficiency of the authentication system.

In the future, we plan to apply this framework to an unconstrained environment, e.g., based daily activities.

Acknowledgments. This work was supported by the Engineering and Physical Sciences Research Council, Centre for Doctoral Training in Cloud Computing for Big Data [grant number EP/L015358/1]

REFERENCES


