Continuous authentication with a focus on explainability

Rodrigo Rocha[a], Davide Carneiro[a,b], Paulo Novais[b]

[a] *CIICESI/ESTG, Polytechnic Institute of Porto, Portugal*
[b] *ALGORITMI/Department of Informatics, University of Minho, Portugal*

Abstract

Traditional explicit authentication mechanisms, in which the device remains unlocked after the introduction of some kind of password, are slowly being complemented with the so-called implicit or continuous authentication mechanisms. In the latter, the user is constantly monitored in one or more ways, in search for signs of unauthorized access, which may happen if a third party has access to the phone after it has been unlocked. There are some different forms of continuous authentication, some of which based on Machine Learning. These are generally black box models, that provide a decision but not an explanation. In this paper we propose an approach for continuous authentication based on behavioral biometrics, machine learning, and that includes domain-dependent aspects for the user to interpret the actions and decisions of the system. It is non-intrusive, does not require any additional hardware, and can be used continuously to monitor user identity.

Keywords: Continuous authentication, Behavioral Biometrics, Mobile Devices, Classification, Explainable AI

1. Introduction

Mobile devices in general, and smartphones in particular, have grown significantly in computational power and functionality in recent years, as well as in their relevance in our daily living. Due to their pervasiveness, they now contain or give access to a significant amount of our sensitive information, from social networks to bank accounts.

In [1], the authors conclude that 4 in each 10 users of a smartphone store information that they deem *secret* on their smartphone, and that 1 in each 3 smartphone owners had accessed or used a smartphone that was not theirs. Indeed, the easiness with which we store sensitive information in these mobile devices makes it easier for unauthorized individuals to gain access to it [2].

Recently, Symantec conducted a social experiment in which 50 smartphones without any authentication mechanism were left on the street, while the devices were collecting data about usage patterns. Results show that 96% of the devices were accessed by at least one person, and in 86% of these the users accessed personal information, such as social networks or e-mail accounts [3].

Smartphones are traditionally secured using information-based authentication mechanisms, notably passwords or visual patterns. In the last years, biometric approaches started to be incorporated such as face recognition or fingerprints. Section 2 analyzes these forms of authentication.

This work proposes a novel authentication mechanism that relies on features of the user's interaction with the screen of the device. An individual interaction model is created for each user, which is then used to continuously monitor the interaction and determine when it deviates from the user's known model. Moreover, and as opposed to other approaches, the system is able to generate visualizations that allow the user to interpret and understand the decisions of the system[4].

2. Existing Authentication Mechanisms

Authentication methods can be characterized as explicit or implicit/continuous. Explicit methods require the input of some identifying information at a specific moment (e.g. fingerprint, password). The device remains indefinitely unlocked afterwards. Implicit authentication, on the other hand, continuously monitors the device in search for clues of unauthorized access, a case in which an explicit authentication method is generally requested.

The most frequent methods are information-based passwords, which may in-

clude text, numbers or visual patterns. These methods are straightforward and computationally efficient. However, it is also easy for a third party to observe the device being unlocked and learn its unlock code/pattern. This is especially true in public spaces, and more so in spaces that have video surveillance [5]. Users also tend to rely on codes/patterns that are easy to memorize, but also easy to guess. In [6], the authors show that in more than 9% of the times, an unauthorized user manages to gain access to a device by "password-guessing" in less than 3 attempts. In the case of visual patterns, it is often as easy since the repeated use of the pattern leaves visible markings on the screen [5].

In order to solve some of these problems, and supported by recent technological developments, the use of biometrics became the new standard. There are two main categories: physiological and behavioral. The former relies on the use of sensors such as video cameras or fingerprint scanners. The latter relies on the analysis of user behavior while interacting with the device (e.g. movement habits, network context) and in the detection of abnormal patterns.

Some example of the latter include the analysis of specific gestures of the user on the screen (e.g. drag, flick, pinch). In [7], a specifically designed glove is used to acquire data. In [8], on the other hand, the authors analyze the way the user types text using the virtual keyboard, using a so-called Typing Authentication and Protection mechanism.

The method proposed in this paper is different from the existing ones in the sense that: 1) it does not require specific hardware or software; 2) it is based on the user interaction with the screen but interaction features are acquired in a transparent way; and 3) it is independent of the application. This approach is based on previous results of the authors, in research work conducted to assess mental fatigue and stress in computer users [9, 10].

3. Architecture

This section describes the main components that implement the proposed continuous authentication mechanism for mobile devices. It considers a central

server, a back-office application, and the users' smartphones (Figure 1). The server implements a REST API [11] that provides services that are used by both the users' smartphones and the backoffice.

The users' smartphones use the endpoints of the API to send interaction data to the server. Client applications need to implement their own logic to capture user interaction, which is dependent on several aspects, including OS. However, the API is completely independent of the client and can be implemented in any device that has a touch screen, internet connection and an API provided by the OS to collect data from user interactions. Smartphones are encouraged to use the local storage to temporarily store interaction data, and send it to the server in batches at regular intervals in order to optimize battery consumption and network usage.

Finally, the backoffice includes a set of services aimed at managing the data and the continuous authentication mechanism (e.g. adjusting sensitivity or thresholds, adjusting model re-training intervals). It also includes tools for generating explanations about the user classification and accompanying intuitive data visualizations.

In the empiric study detailed in Section 4, an Android application was developed to implement the API in Android devices. In the server-side, MongoDB database was used to store data. Profiling scripts and the generation of visualizations were implemented in R. The lifecycle for model training was implemented using the H2O API. Finally, the REST API was implemented using Node and Express. However, the proposed architecture is generic enough to be implemented with equivalent technologies.

Whenever there was interaction with the application, it collected data describing touch events. In order to reduce variability, the application aggregates interaction data using a first-in first-out sliding window of size 30. The average of each variable in the sliding window is calculated and stored locally until it is sent to the server.

We propose the use of 12 interaction features. The first 9 describe the maximum, average, and minimum values of touch duration, area and intensity.

In order to calculate the other 3, we fit a quadratic curve to the intensity of each touch over time. The curve obtained represents the touch pattern of the user, and represents a composite feature: it combines the duration and intensity dimensions. The remaining 3 variables are thus the coefficients of the quadratic curve, respectively x^2, x and n. This data is aggregated and processed locally in the device, and is then sent to a central server where it is stored.

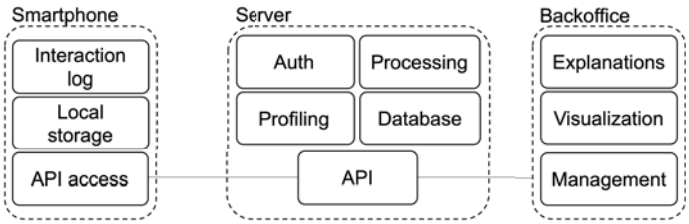

Figure 1: Main modules of the proposed system.

4. Methodology

To validate this approach, an experiment was setup in which users interacted regularly with their mobile devices, using a specifically developed application. This was a game-like application, developed for Android devices, in which the users needed to complete relatively simple mathematics tasks and navigate through menus. The main goal of the application was to collect interaction data from each user.

Data was collected from 30 users, 15 women and 15 men, with ages ranging between 10 and 67 ($\bar{x} = 35.95$, $\sigma = 14.96$). Users were individually brought in to a laboratory, in which they played the game. Users were not instructed regarding the nature of the experiment beforehand. This resulted in the collection of 1665 instances of data (each representing the result of aggregating over a sliding window of 30 touches). After collection, data was normalized.

Initially, data was visually and statistically analyzed in search for significant inter-user differences. As Figures 2 and 3 depict, there are indeed differences between users. These Figures show, respectively, how touch duration and touch

intensity are distributed for each user. Differences are clearly visible, not only in terms of the median value but also of the distribution of the data.

This provides some support to the hypothesis that users may have different interaction patterns. Similar differences are observed in the remaining variables.

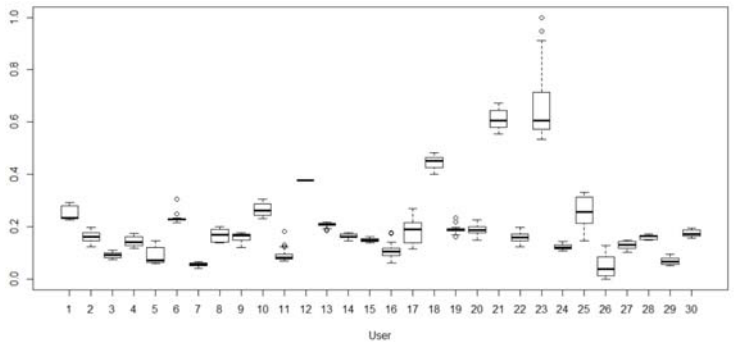

Figure 2: Distribution of average touch duration for all the users.

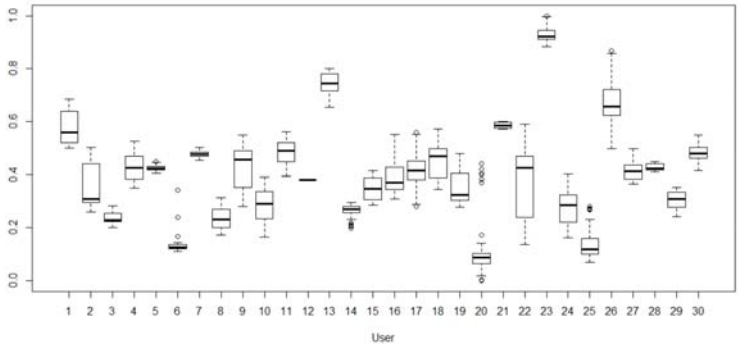

Figure 3: Distribution of average touch intensity for all the users.

4.1. User Interaction Model and Classification

This section describes the steps taken to create a so-called individual *interaction model*, based on the proposed interaction features, suitable to be used for identity classification. To this end, the following approach was implemented. For each participant a dataset was built that contains all the interaction data of that participant and, in the same proportion, data selected randomly from

other participants. This provides us with a balanced dataset that contains data from the actual participant associated to the dataset and from other random participants. A new binary column was added to the dataset that encodes if each instance contains data of the participant associated to the dataset (true/1) or from other participants (false/0).

For each of these 30 datasets a Neural Network [12] with 2 layers (of 200 neurons each) was trained. Thus, there is an interaction model for each participant. These neural networks have thirteen inputs (the twelve interaction features plus the boolean predictor variable) and two outputs (the probability of the classification result being true or false). The interaction model of each user is updated at regular intervals, when new interaction data is available in the central server. Models are stored in a central database, in which there is one instance for each user.

The continuous authentication service thus works as follows. The server's API accepts requests from mobile devices containing an instance of processed interaction data and the identifier of the owner of the device in which the data was collected. The service then retrieves that user's interaction model from the database, provides it with the interaction data, and obtains a classification result. Then, if this percentage is above a specific threshold, it is assumed that the current user of the device is the authorized user. Otherwise, it is assumed that an unauthorized user gained access to the device. The server does this for each packet of interaction data and returns to the mobile device a classification result for each one .

If a packed of data is classified as belonging to the authorized user, it is added to the database and is later used to update the interaction model of that user, guaranteeing that it is always up to date, namely to reflect small variations that happen over time.

Alternatively, if the packet is classified as an identity breach, the mobile device may choose to lock the screen and immediately request an explicit authentication mechanism, such as a password or a fingerprint. Two cases can occur: 1) The user succeeds in unlocking the device, the event is marked as a

false positive, and the instance of data is saved in the database nonetheless; 2) The user does not unlock thee device, the event is considered a true positive, and the data is discarded as it represents an interaction from a third party.

4.2. Experimental Results

To validate the proposed approach, data was split into two groups: the train split holding 75% of the data (1248 instances) and the test split holding the remaining 25% (417 instances). The train split was used to train one interaction model for each user, as detailed in Section 4.1.

Both splits were also transformed in order to simulate authentication breaches since no unauthorized accesses took place during the data collection. To emulate these unauthorized accesses, the users of 67% of the instances were randomly changed and the instances marked as a positive. The remaining 33% were left unchanged.

After the training of the models, each instance of data in the test split was then submitted to the classifier service, which classified it as *breach* (positive) or *no breach* (negative) according to the supposed owner of the device and using a threshold of 0.5.

From the validation of each model the confusion matrix and the ROC curve were created. However, their size prevents us from including them in this paper. Instead, Table 1 summarizes the main metrics of each model.

In average, the accuracy of the 30 models for the test data is of 98.94%. The average precision is 0.943 and the average recall is 0.9. In general, these are promising results that, albeit the relatively small number of users, indicates that this approach may indeed be a suitable way to classify user identity in mobile devices.

4.3. Explainability

Most of the existing authentication methods based on Machine Learning can be seen as a black box, that is, they do not provide an explanation or

Table 1: Summary of the metrics of each model.

Model ID	Accuracy	Precision	Recall	AUC
1	1	1	1	1
2	0.9706	0.842	0.64	0.99
3	1	1	1	1
4	0.9363	1	0.278	0.955
5	1	1	1	1
6	0.9975	0.90	1	0.995
7	1	1	1	1
8	1	1	1	1
9	0.9829	1	0.696	0.989
10	0.9975	0.966	1	0.999
11	0.9510	0.368	0.467	0.924
12	1	1	1	1
13	1	1	1	1
14	0.9975	1	0.877	0.999
15	0.9975	1	0.889	0.999
16	0.9559	0.923	0.706	0.98
17	0.9668	0.956	0.796	0.982
18	1	1	1	1
19	1	1	1	1
20	0.9975	1	0.90	0.999
21	1	1	1	1
22	0.9877	0.789	0.938	0.988
23	1	1	1	1
24	0.9828	0.74	0.895	0.992
25	0.9775	1	0.958	0.999
26	1	1	1	1
27	1	1	1	1
28	1	1	1	1
29	0.9951	0.917	0.917	0.999
30	0.9975	0.889	1	0.999

justification for the result of the user classification mechanism. This significantly decreases the transparency of the whole system, and the user is limited to accepting its decision without clearly understanding why. In the proposed approach, explanations exist in two different ways.

The first is based on the importance of each feature in the model. In the software package used, variable importance is calculated using the Gedeon method and it provides the user with information regarding how important each variable is for predicting user identity. Depending on the users' interaction patterns, the variables in each model will have different relative importance. This information can be used per se, or can be used to select the variables to show in the visualizations described next. The second is based on a visual analysis that compares

the interaction profile of the user against the data being classified, that allow the user to understand in which ways the interaction was different and why a given decision was made by the system.

In order to build this kind of explanations for a given user we calculate the quartiles and the interquartile range (IQR) of each feature, considering the data of that user in the central database. Based on these values, we define the *normal* upper and lower limits for each feature, as proposed by [13] and as defined by equations $lower_{u,i} = Q1_{u,i} - 1.5 * IQR_{u,i}$ and $upper_{u,i} = Q3_{u,i} + 1.5 * IQR_{u,i}$. These limits represent the boundaries between which each user normally interacts.

Using these limits we can then build graphical visualizations such as those in Figure 4. The left image simply shows the average values of the features for three different users, allowing to perceive *how* the interaction is different. For instance, User 1 has significantly higher values of minimum of touch duration and of maximum touch intensity, which clearly separate him from the others. The right image shows the upper and lower limits of User 1, compared against an instance of his own interaction (dotted line) against that of another user (User 2, dashed line). Visualizations such as these allow to perceive why a given instance may be classified as a positive or as a negative.

5. Conclusions and Future Work

This paper proposed an approach for continuous authentication using behavioral biometrics. It is completely non-intrusive and transparent, does not require any specific hardware, and is based on 12 interaction features. These features describe several modalities of interaction such as time, intensity and area. Individual interaction models are built that allow to determine if a given instance of interaction belongs or not to the supposed owner of the device.

An online authentication service was developed that can be easily integrated by mobile app and device developers. The service is also able to generate visual explanations about the differences observed between the interaction and the

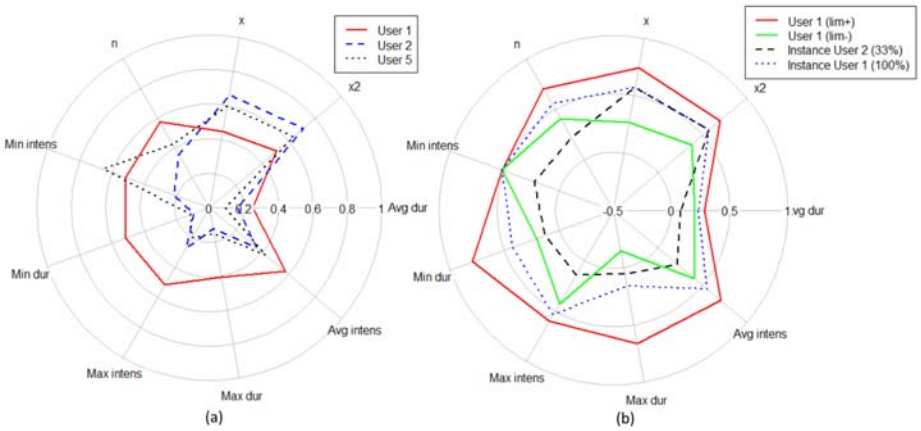

Figure 4: (a) Graphical representation of the average values of each interaction feature, for three users. (b) Graphical representation of the interaction profile of User 1, and comparison with two instances for classification, one from the same user and another from a different one.

model.

Current work focuses on scaling the system to the magnitude of hundreds of users, to assess the suitability of the approach, not only in terms of computational efficiency but also in terms of accuracy. Once this step is complete, we will compare the pros and cons of this approach with existing ones, namely in terms of accuracy, usability and easiness of implementation/integration. We will also improve the Explanations module to generate text that can be used together with the visualizations.

Acknowledgments

This work has been supported by national funds through FCT – Fundação para a Ciência e Tecnologia through project UIDB/04728/2020.

References

[1] J. L. Boyles, A. Smith, M. Madden, Privacy and data management on mobile devices, Pew Internet & American Life Project 4 (2012) 1–19.

[2] A. K. Jain, D. Shanbhag, Addressing security and privacy risks in mobile applications, IT Professional 14 (5) (2012) 28–33.

[3] S. Wright, The symantec smartphone honey stick project, Symantec Corporation, Mar (2012) 1–17.

[4] W. Samek, Explainable AI: interpreting, explaining and visualizing deep learning, Vol. 11700, Springer Nature, 2019.

[5] L. Zhou, Y. Kang, D. Zhang, J. Lai, Harmonized authentication based on thumbstroke dynamics on touch screen mobile phones, Decision Support Systems 92 (2016) 14–24.

[6] L. Lu, Y. Liu, Safeguard: User reauthentication on smartphones via behavioral biometrics, IEEE Transactions on Computational Social Systems 2 (3) (2015) 53–64.

[7] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, N. Nguyen, Continuous mobile authentication using touchscreen gestures, in: Homeland Security (HST), 2012 IEEE Conference on Technologies for, Citeseer, 2012, pp. 451–456.

[8] T. Feng, X. Zhao, B. Carbunar, W. Shi, Continuous mobile authentication using virtual key typing biometrics, in: Trust, security and privacy in computing and communications (TrustCom), 2013 12th IEEE international conference on, IEEE, 2013, pp. 1547–1552.

[9] A. Pimenta, D. Carneiro, J. Neves, P. Novais, A neural network to classify fatigue from human–computer interaction, Neurocomputing 172 (2016) 413–426. doi:10.1016/j.neucom.2015.03.105.

[10] D. Carneiro, P. Novais, J. M. Pêgo, N. Sousa, J. Neves, Using mouse dynamics to assess stress during online exams, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2015, pp. 345–356.

[11] M. Masse, REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces, O'Reilly Media, Inc., 2011.

[12] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, C. L. Giles, Neural network classification and prior class probabilities, in: Neural networks: tricks of the trade, Springer, 1998, pp. 299–313.

[13] J. W. Tukey, Exploratory data analysis, Vol. 2, Reading, Mass., 1977.