Does Context matter for the Performance of Continuous Authentication Biometric Systems? An Empirical Study on Mobile Devices

Soumik Mondal and Patrick Bours
Norwegian Information Security Laboratory (NISLab)
Gjvik University College
firstname.lastname@hig.no

Abstract: In this paper we will show that context has an influence on the performance of a continuous authentication system. When context is considered we notice that the performance of the system improves by a factor of approximately 3. Even when testing and training are not based on exactly the same task, but on a similar task, we see an improvement of the performance over a system where the context is not included. In fact, we proof that the performance of the system depends on which particular kind of task is used for the training.

1 Introduction

Access control on a mobile device (*i.e.* smart phone or tablet) is generally implemented as a one-time proof of identity during the initial log on procedure [CSW$^+$13]. The validity of the user is assumed to be the same during the full session. Unfortunately, when a device is left unlocked, any person can have access to the same sources as the genuine user. This type of access control is referred to as static authentication. On the other hand we have *Continuous Authentication* (sometimes also called *Active Authentication*), where the genuineness of a user is continuously verified based on the activity of the current user operating the device. When doubt arises about the genuineness of the user, the system can lock, and the user has to revert to the static authentication access control mechanism to continue working.

Due to its novelty, little research was done in this area [FBM$^+$13, RHM14]. *Continuous Authentication (CA)* by analysing the user's behaviour profile on mobile input devices is challenging due to the limited amount of information and the large intra-class variations. Most of the previous research was actually done as periodic authentication, where the analysis was based on a fixed number of actions or fixed time period.

In our research we address a fundamental question, whether context really matters for the performance of a biometric CA system. We look at how much the performance changes if a user's behaviour profile is created by performing a particular task in a specific applications on the mobile device compared to test data coming from various application with different tasks. We use a CA biometric system proposed by [MB15], which checks the genuineness of the user during the full session.

2 Background Knowledge

Classifier(s)

We used two classifiers in a *Multi Classifier Fusion (MCF)* architecture to achieve an acceptable system performance [KHDM98]. Due to the nature of the data we found that one prediction model and one regression model gave us a better learning accuracy. We used *Support Vector Machine (SVM)* as a prediction model and *Counter-propagation Artificial Neural Network (CPANN)* as a regression model.

Trust Model

In our analysis we look at every single action performed by the user and we have used the *Trust Model* [Bou12] in our analysis. The basic idea of the *Trust Model* is that the trust of the system in the genuineness of the current user depends on the deviations from the way this user performs various actions on the system. If a specific action is performed in accordance with how the genuine user would perform the task (*i.e.* as stored in the template), then the systems trust in the genuineness of this user will increase. We call this a *Reward*. If there is a large deviation between the behaviour of the genuine user and the current user, then the trust of the system in that user will decrease, which is called a *Penalty*. If the trust of the system in the genuineness of the user is too low, then the user will be locked out of the system. In particular if the trust drops below a pre-defined threshold $T_{lockout}$ (global or user specific), then the system locks itself and will require static authentication of the user to continue working. In our research, we use the Dynamic Trust Model [MB15] where, the penalty/reward is calculated based on the resultant classifier score sc_i according to

$$\Delta_{Trust}(sc_i) = \min\{-D + D \times (\frac{1 + \frac{1}{C}}{\frac{1}{C} + \exp(-\frac{sc_i - A}{B})}), C\}. \tag{1}$$

Let the trust value after i actions be denoted by $Trust_i$, and let the i^{th} action have a classification score sc_i. Then we have the following relation between $Trust_{i-1}$ and $Trust_i$:

$$Trust_i = \min\{\max\{Trust_{i-1} + \Delta_{Trust}(sc_i), 0\}, 100\} \tag{2}$$

The trust level never exceeds 100 to assure that an impostor cannot profit from a longer period of time where the genuine user behaved according to his own profile.

Performance Measure

In this research, we focus on actual CA that reacts on every single action from a user. Therefore, we use the *Average Number of Genuine Actions (ANGA)* and *Average Number of Impostor Actions (ANIA)* as a performance evaluation metric [MB15]. A detailed explanation of the performed actions is given in Section 3.

Our goal is obviously to have ANGA as high as possible, while at the same time the ANIA value must be as low as possible. The last is to assure that an impostor user can do as little harm as possible, hence he/she must be detected as quick as possible. In our analysis, whenever a user is locked out, we reset the trust value to 100 to simulate a new session starting after the set of actions that lead to this lockout.

3 Data Description and Feature Extraction

In our research, we used a publicly available continuous mobile touch gesture dataset [FBM$^+$13]. To the best of our knowledge is this the only dataset publicly available and the structure is suitable for our analysis. A brief description of the dataset is given below.

Data Description

During the data collection process a custom application was deployed on 5 different Android mobile devices and touch gestures from 41 volunteers with 5 to 7 session per participants was collected [FBM$^+$13]. Data was collected in 7 different tasks, *i.e.* 4 different Wikipedia Reading articles and 3 different Image Comparison Games.

We noticed in the data that all 41 users completed tasks 1-4 (*i.e.* 3 different reading tasks and one image comparison task). Task 5 (image comparison) was completed by 40 users, while tasks 6 and 7 (reading article and image comparison) were completed by only 14 users. These last 14 users were a subset of the 40 users that completed task 5.

Feature Extraction and Selection

In our analysis, we divided the sequence of consecutive tiny movement data into actions (*i.e.* strokes). From the raw data 31 different features were calculated. [FBM$^+$13] shows the details of these feature extraction process.

Before building the classifier models we applied the feature selection technique proposed by [VK09]. We also analysed another feature selection technique [LTM$^+$12], but found that the learning accuracy of the SVM dropped from 97.7% to 52.4% and the CPANN learning accuracy dropped from 97.5% to 89.9%. We have also observed that in some cases the SVM models of some users were biased, meaning that the models always classified one particular class.

4 Methodology

Verification Process

In our research, we used three verification processes. We split the data of each of the users into a part for training and a part for testing. In all cases the classifiers are trained with genuine and impostor (training) data. The amount of training data of the genuine user is 50% of the total amount of data of that user. The training data of the impostor users is taken such that the total amount of impostor data equals the amount of training data of the genuine user. This is done to avoid bias towards either the genuine or the impostor class. The three verification processes described below might be seen to correspond with an "Internal System" *(VP-1)* where all the impostor users are known to the system, an "External System" *(VP-3)* where impostor users during testing are not known to the system and a combination of these (*VP-2*) where 50% of the impostor users are known to the system.

Classifier Fusion

Below we describe the two fusion techniques that we applied in this research. The score vector we use for further analysis is $(f^1, f^2) = (Score_{svm}, Score_{cpann})$. We have applied *Score Fusion (SF)* and *Penalty-Reward Fusion (PRF))*.

In SF a single score is calculated from the 2 classifier scores and this score is used in Equation 1 (see Section 2) to calculate the penalty-reward. The final score is the weighed sum of the two scores, where f^1 gets weight w and f^2 gets weight $1 - w$. The calculated score is used in Equation 2 to calculate the updated system trust level. The weights are optimized using linear search.

In case of PRF we individually calculate the penalty-reward from Equation 1 for the 2 classifier scores. Let $\Delta_{Trust}(sc_i^1)$ represents the penalty-reward from Classifier-1 (*i.e.* SVM) and $\Delta_{Trust}(sc_i^2)$ stands for the penalty-reward from Classifier-2 (*i.e.* CPANN). We combine these with weighted fusion to calculate the system trust from Equation 2 in the same way as above for SF. As before are the weights optimized using linear search.

System Architecture

The system is divided into two basic phases, the training phase and the testing phase. In the training phase, the training data is used to build the classifier models and the models are stored in a database for use during the testing phase. Each genuine user has his/her own classifier models and training features.

In the testing phase, we use the test data which was separated from the training data for comparison. In the comparison, we use the models and training features stored in the database and obtain the classifier score (probability) of each sample of the test data according to the performed action. This score will then be used to update the trust value $Trust$ in the trust model (see Section 2). Finally, the trust value $Trust$ is used in the decision module, to determine if the user will be locked out or can continue to use the device. This decision is made based on the current trust value and the lockout threshold ($T_{lockout}$).

5 Result Analysis

In this section, we analyse the results that we obtained from our performance analysis. We divide our analysis into three major parts based on the verification process (see Section 4). We use one-hold-out cross validation testing. The total number of data sets of genuine users is 41 for tasks 1 to 4 and hence, the total number of data sets of impostor users for these tasks is 1640 (41 × 40). For task 5 these numbers are 40 resp. 1560 (40 × 39), while for tasks 6 and 7 these numbers are 14 and 182 (14 × 13). We report the results for user specific lockout thresholds (T_{us}), where each threshold satisfies $50 \leq T_{us} < \min(Trust_{genuine})$ and is optimized using linear search.

Interpretation of the tables: The results from our analysis are divided into 4 possible categories. The categories are divided based on genuine user lockout ($+$ if the genuine user is not locked out and $-$ in case of lock out) and non-detection of impostor users ($+$ if an impostor user is locked out and $-$ otherwise). Each of the genuine users can thus be classified of 4 categories: (+/+), (+/-), (-/+), and (-/-), where the first sign is related to the genuine user and the second to the impostor users. In our analysis are genuine users never locked out, so in Table 1 only the categories +/+ and +/- are shown.

The column *# Users* shows how many users fall within each of the 4 categories (*i.e.* the values sum up to 41). In the column ANGA a value will indicate the Average Number of Genuine Actions in case indeed genuine users are locked out by the system. If the genuine users are not locked out, then we actually cannot calculate ANGA, which is indicated by ∞. The column ANIA will display the Average Number of Impostor Actions, and is based on all impostors that are detected. The actions of the impostors that are not detected are not used in this calculation, but the number of impostors that is not detected is given in the column *# Imp. ND*. This number should be seen in relation to the number of users in that particular category. For example, in the `VP-3`, `PRF` '+/-' category described in Table 1, we see that *# Users* equals 3, *i.e.* there are $3 \times 40 = 120$ impostor test sets, and 4 impostors within these 120 impostors are not detected by the system as being an impostor. In this particular case for the full system, 4 out of 1640 of the impostors are not detected, hence we have a 0.24% Impostor Non-Detected Rate (INDR).

5.1 Result Analysis Without Context

We first considered the CA system performance irrespective of the context. Table 1, shows the result we have obtained for this analysis. We clearly observe from the table that *Score Fusion (SF)* performs better than *Penalty-Reward Fusion (PRF)* for each of the verification processes. For this reason we only used *Score Fusion (SF)* for our further analysis. Note that the summary for each verification process shows the system ANGA and ANIA values, as well as the INPR value.

Table 1: Results obtained from our analysis for the context independent evaluation.

	Categories	SF				PRF			
		# User	ANGA	ANIA	# Imp. ND	# User	ANGA	ANIA	# Imp. ND
VP-1	+/+	41	∞	11		40	∞	13	
	+/-					1	∞	435	14
	Summary		∞	11	0%		∞	24	0.85%
VP-2	+/+	40	∞	17		38	∞	23	
	+/-	1	∞	88	1	3	∞	275	39
	Summary		∞	19	0.06%		∞	47	2.38%
VP-3	+/+	40	∞	22		38	∞	19	
	+/-	1	∞	286	1	3	∞	233	4
	Summary		∞	29	0.06%		∞	35	0.24%

Comparison with Previous Research

We compared our results with previous results based on the same dataset in [FBM$^+$13]. Table 2, shows the previous research results in terms of ANIA/ANGA by using the conversion technique described in [MB15]. We see that our methods outperform the previous research for `VP-1`, while for `VP-2` and `VP-3` our ANIA values are higher. Note that the other researches that are based on the same dataset have done the analysis in the same manner as we have done for `VP-1`.

Table 2: Comparison with Previous Research.

Reference	# Users	FNMR	FMR/INDR	Blocksize	ANGA	ANIA	*P-value*
[FBM$^+$13]	41	3%	3%	12	400	12	
[RHM14] - Horizontal	41	1.75%	1.75%	11	629	11	
[RHM14] - Vertical	41	2.8%	2.8%	11	393	11	
Our (VP-1 with SF)	41	0%	0%	NA	∞	11(±9)	0.79
Our (VP-2 with SF)	41	0%	0.06%	NA	∞	19(±13)	0.98
Our (VP-3 with SF)	41	0%	0.06%	NA	∞	27(±15)	0.93

5.2 Result Analysis With Context

The main objective of this paper is not to find a better CA method, but to determine if including the context in the analysis has an impact on the performance. In this section we present the results we obtained from our analysis by considering this context. We measure the CA system performance by training the system using the data obtained from a particular task and then testing the system with the data from the various tasks performed by the users. Tables 3, 4, and 5 show the results we obtained from the VP-1, VP-2 and VP-3 verification processes respectively, using the *Score Fusion (SF)* technique. In all cases we noted that genuine users were never locked out, hence we found that ANGA=∞, so these values are not reported. The tables only contain the ANIA values, as well as the INPR values between brackets. The values on the diagonal are displayed in bold to signify that training and testing is done using the same task, while for all off-diagonal values the training and testing is done with 2 different tasks.

Table 3: Results obtained from our analysis for context dependent evaluation for VP-1.

Test Train	Task-1	Task-2	Task-3	Task-4	Task-5	Task-6	Task-7
Task-1	**3 (0.1%)**	3 (0.1%)	16 (4.1%)	30 (3.1%)	30 (2.8%)	11 (4.9%)	42 (1.1%)
Task-2	7 (0.6%)	**3 (0.1%)**	15 (3.6%)	28 (2.7%)	29 (2.4%)	12 (4.9%)	33 (0.5%)
Task-3	14 (2.8%)	19 (4.4%)	**4 (0.2%)**	30 (2.6%)	32 (2.8%)	13 (7.7%)	53 (3.8%)
Task-4	22 (4.5%)	20 (3.6%)	20 (4.4%)	**3 (0.1%)**	6 (0.1%)	15 (4.4%)	15 (0.5%)
Task-5	16 (2.3%)	27 (4.9%)	19 (4.4%)	7 (0.6%)	**4 (0.0%)**	17 (9.3%)	11 (0.5%)
Task-6	33 (19.2%)	30 (12.6%)	22 (9.3%)	54 (17.0%)	43 (13.7%)	**4 (0.0%)**	42 (1.6%)
Task-7	32 (19.2%)	47 (25.8%)	48 (33.0%)	13 (3.8%)	12 (2.7%)	15 (8.8%)	**4 (0.0%)**

Recall that tasks 1 to 4 had 41 participants, while task 5 had only 40 participants and the last 2 tasks had only 14 participants. When training with task i and testing with task j we only considered those participants that participated in both tasks. Also recall that tasks 1, 2, 3, and 6 are article reading tasks and tasks 4, 5, and 7 are image comparison games, which will help to understand the results and it's impact based on the type of the tasks.

Table 4: Results obtained from our analysis for context dependent evaluation for VP-2.

Test Train	Task-1	Task-2	Task-3	Task-4	Task-5	Task-6	Task-7
Task-1	**6 (0.2%)**	21 (4.0%)	17 (3.8%)	31 (3.3%)	29 (1.9%)	20 (12.1%)	51 (2.7%)
Task-2	11 (1.7%)	**5 (0.2%)**	18 (3.2%)	31 (3.5%)	28 (2.5%)	18 (11.5%)	46 (2.7%)
Task-3	26 (7.7%)	24 (6.7%)	**10 (1.6%)**	35 (4.7%)	35 (3.7%)	20 (13.2%)	48 (6.0%)
Task-4	27 (6.8%)	30 (4.4%)	33 (8.3%)	**10 (1.1%)**	15 (1.5%)	19 (10.4%)	14 (1.6%)
Task-5	22 (4.2%)	27 (4.6%)	24 (4.6%)	8 (0.7%)	**6 (0.3%)**	18 (6.6%)	18 (2.2%)
Task-6	16 (4.4%)	16 (1.6%)	21 (2.7%)	34 (0.0%)	38 (0.5%)	**7 (2.2%)**	39 (2.2%)
Task-7	22 (0.4%)	26 (1.2%)	30 (2.7%)	18 (0.0%)	17 (1.1%)	14 (4.4%)	**7 (0.5%)**

Table 5: Results obtained from our analysis for context dependent evaluation for VP-3.

Test Train	Task-1	Task-2	Task-3	Task-4	Task-5	Task-6	Task-7
Task-1	**9 (1.3%)**	18 (3.9%)	20 (5.2%)	28 (2.3%)	23 (2.1%)	19 (10.4%)	36 (4.4%)
Task-2	11 (1.6%)	**7 (0.9%)**	19 (5.0%)	30 (3.5%)	27 (1.9%)	15 (10.4%)	40 (2.2%)
Task-3	17 (4.5%)	20 (0.9%)	**7 (0.9%)**	24 (2.7%)	22 (1.3%)	14 (5.5%)	33 (2.2%)
Task-4	22 (4.2%)	21 (2.1%)	20 (5.1%)	**6 (0.5%)**	8 (0.7%)	15 (7.1%)	21 (1.1%)
Task-5	17 (2.4%)	24 (4.5%)	19 (4.1%)	9 (0.9%)	**8 (0.6%)**	12 (1.6%)	7 (0.0%)
Task-6	14 (4.9%)	13 (1.1%)	32 (8.8%)	39 (1.6%)	38 (0.5%)	**11 (4.4%)**	30 (1.1%)
Task-7	23 (8.2%)	33 (12.6%)	30 (7.1%)	16 (0.5%)	19 (1.6%)	15 (7.6%)	**6 (0.0%)**

On average we find that ANIA equals 3.6/7.3/7.7 for `VP-1`/`VP-2`/`VP-3` when training and testing with the same task, while it equals on average 23.9/25.1/21.8 when using different tasks for training and testing. The next values are split according to the type of task (reading or image comparison) and we found that if the task is different, but the type of task is the same, then the ANIA equals 14.4/17.7/16.3, while if also the type of tasks differs, then the ANIA values are much higher: 31.0/30.7/25.9. Finally, if we consider a reading task for training then the system ANIA for any of the other tasks is 26.7/28.0/24.3 and for an image comparison tasks these values are 20.1/21.2/18.4.

5.3 Discussion

We have to stress that the amount of samples in the dataset allowed for Linear Search optimization of the parameters in the algorithm. The amount of data was not sufficient to be used for more advanced optimization methods like Genetic Algorithm.

The context dependent analysis performs better than context independent analysis for all the verification processes (3.6/7.3/7.7 vs. 11/19/29 for `VP-1`/`VP-2`/`VP-3`). This finding is in line with the findings from [KH14]. We do note that the INDR goes up slightly when including context in the analysis, but the average values stay low when the same task is considered. Training with an image comparison task gave slightly better results when testing with an arbitrary other task.

6 Conclusion

In this research we looked at the performance of a system when including context in the analysis. We saw that the performanced improved (i.e. a lower ANIA value) when context was considered. The optimal performance was obtained by training and testing with the data of the exact same task. Even when testing is done with a similar kind of task is the performance still better than when no context is used, but the differences are not as significant.

Our results show that we cannot train and test a system with a specific task and then assume that the full system will perform according to the results that are found. Our future work will include training with multiple sessions of similar and different kind of tasks and see how this affects the performance of the system.

References

[Bou12] P. Bours. Continuous keystroke dynamics: A different perspective towards biometric evaluation. *Information Security Technical Report*, 17:36–43, 2012.

[CSW$^+$13] Zhongmin Cai, Chao Shen, Miao Wang, Yunpeng Song, and Jialin Wang. Mobile Authentication through Touch-Behavior Features. In *Biometric Recognition*, volume 8232 of *Lecture Notes in Computer Science*, pages 386–393. Springer, 2013.

[FBM$^+$13] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Trans. on Information Forensics and Security*, 8(1):136–148, 2013.

[KH14] Hassan Khan and Urs Hengartner. Towards Application-centric Implicit Authentication on Smartphones. In *15th Workshop on Mobile Computing Systems and Applications*, pages 10:1–10:6. ACM, 2014.

[KHDM98] J. Kittler, M. Hatef, R. P W Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[LTM$^+$12] Cosmin Lazar, Jonatan Taminau, Stijn Meganck, David Steenhoff, Alain Coletta, Colin Molter, Virginie de Schaetzen, Robin Duque, Hugues Bersini, and Ann Nowe. A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 9(4):1106–1119, 2012.

[MB15] Soumik Mondal and Patrick Bours. A computational approach to the continuous authentication biometric system. *Information Sciences*, 304:28 – 53, 2015.

[RHM14] A. Roy, T. Halevi, and N. Memon. An HMM-based behavior modeling approach for continuous mobile authentication. In *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 3789–3793, 2014.

[VK09] D. Ververidis and C. Kotropoulos. Information Loss of the Mahalanobis Distance in High Dimensions: Application to Feature Selection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(12):2275–2281, 2009.