Context Independent Continuous Authentication using Behavioural Biometrics

Soumik Mondal
NISlab, Gjøvik University College, Norway
soumik.mondal@hig.no

Patrick Bours
NISlab, Gjøvik University College, Norway
patrick.bours@hig.no

Abstract

In this research, we focus on context independent continuous authentication that reacts on every separate action performed by a user. The experimental data was collected in a complete uncontrolled condition from 53 users by using our data collection software. In our analysis, we considered both keystroke and mouse usage behaviour patterns to prevent a situation where an attacker avoids detection by restricting to one input device because the continuous authentication system only checks the other input device. The best result obtained from this research is that for 47 biometric subjects we have on average 275 actions required to detect an imposter where these biometric subjects are never locked out from the system.

1. Introduction

Contrary to static authentication is the goal in *Continuous Authentication (CA)* to check if the current user of a system is the genuine user who logged on to the system or if a change of a user has occurred. In the last case the system should lock to avoid any damage done by the impostor. The obvious requirements are to detect an impostor as fast as possible to control the amount of damage that he can do, while at the same time trying to avoid, to the largest possible extent, the incorrect locking out of the genuine user. Research on continuous authentication started in 1995 when Shepherd [14] showed some impressive result on CA using keystroke dynamics. These days CA is getting more popular because of the security requirements in office environments and the DARPA's Active Authentication project announced in 2012. There are many possible ways to implement a CA system, but behavioural biometrics seems promising enough to achieve a cost effective (due to no special hardware required) and unobtrusiveness solution [16]. More particular, Keystroke dynamics [2, 4, 14] and Mouse dynamics [6, 11, 13] are reasonably resilient against changes in the user's physical environment. It is known however that typing behaviour is influenced by the emotional state of a person [5]. In our research, we have used Keystroke and Mouse dynamics both as biometric modality [1, 3, 8, 15], but the proposed scheme could be useful for continuous authentication by using any biometric modality.

Continuous Authentication by analysing the user's behaviour profile on the computer input devices is challenging due to the limited information, large intra-class variations and the sparse nature of the information. As a result, most of the previous research was done as a periodic authentication, where the analysis was performed based on a fixed number of actions or fixed time period. Also, the experimental data were obtained for most of the previous research in controlled conditions, where the task and the environment were fixed. In our experiment we removed these restrictions. Our system uses the behaviour of the current user to determine the level of trust in the genuineness of this user. We do so by focussing on how a user uses the input mechanisms of keyboard and mouse. We combine both to avoid a situation where an impostor restricts to e.g. keystrokes because the system only checks the mouse dynamics or vice versa. Like in all behavioural biometric systems is the behaviour of the user compared to the template of the genuine user. The similarity or dissimilarity between these two will determine if the trust in the genuineness of the user will increase or decrease. If the trust in the genuineness has become too low, then the system will protect itself by locking out the current user.

In this paper, we show that the performance reporting measures used in biometrics (FMR and FNMR) are no longer valid for continuous biometrics and we use the *Average Number of Genuine Actions (ANGA)* and the *Average Number of Impostor Actions (ANIA)* as new performance reporting measures that can be used to compare continuous authentication systems. We show how ANGA and ANIA can be computed.

The summary of our contribution made in this paper as follows: 1) New scheme for continuous authentication which verifying the genuineness of the user for every action performed by the user; 2) The new evaluation metric has been introduced for Continuous Authentication Biometric Systems; 3) Introduction of novel mouse move trajectory related features for continuous authentication research; 4)

Consider multiple input device to avoid the situation where an attacker avoids detection by restricting to one input device because the system only checks the other input device.

The remainder of this paper will be as follows. In Section 2, we provide the data description used in this research. We are going to describe feature extraction and classification techniques in Section 3. In Section 4, we discuss the methodology followed to carry out this research. Result analysis will be provided in Section 5. Finally, we conclude this research with future work in Section 6.

2. Data Description

Due to unavailability of any database which consists of combined keystroke and mouse dynamics data in a continuous manner, we build our own dataset to conduct our research. We designed a Windows OS based behavioural logging tool. This tool continuously captures Keystroke and Mouse interaction data. Log data was stored on the participant's own device in CSV format. Privacy of the users as well as confidentiality of the sensitive data were maintained throughout the experiment. The structures of the data is as follows:

Keystroke Events: Table 1, shows the data format for keystroke events. Sequence (*i.e*. Seq.) represents the sequential occurrence of the events. The *Evt. Type* is always 'K' (*i.e*. keystroke related events). Keystroke events have only two types of *actions*, key press (*i.e*.'D') and key release ('U'). The *Value* field states which key was pressed or released with UTF-8-encoded. The *time-stamp* was recorded in milliseconds when the event occurred with a 16ms sampling interval. The *relation* attribute contains a corresponding sequence number of a previous event. *Flag* is an Integer indicating which alternate/system key was active.

Table 1. Data structure for keystroke events.

Seq.	Evt. Type	Action	Value	Time	Relation	Flag	Additional fields
n	'K'	'D' 'U'	String	ms	evt. ID	Int	n/a Count

Mouse Events: The data format for mouse events is shown in Table 2. Sequence (*i.e*. Seq.) represents the sequential occurrence of the events. The *Evt. Type* is always 'M' (*i.e*. mouse related events). Mouse events can have four types of *actions*, mouse move ('M'), mouse wheel use ('W'), mouse button press ('D') and mouse button release ('U'). The *Value* field contains the x_y mouse pointer coordinates concatenated by an '_' underscore character. In case of mouse wheel use, *Value* is the corresponding delta value indicating how much the wheel was scrolled; positive values are upward scrolls, negative are downward scrolls. The *time-stamp* was recorded in milliseconds when the event occurred with a 16ms sampling interval. The *relation* attribute contains a corresponding sequence number of a previous event. *Flag* is an Integer indicating which mouse button was pressed/released. *Additional fields* indicates the active rectangle area for mouse button press and release events.

Table 2. Data structure for mouse events.

Seq.	Evt. Type	Action	Value	Time	Relation	Flag	Additional fields
n	'M'	'M' 'U' 'D' 'W'	x_y x_y x_y Delta	ms	evt. ID	n/a Int n/a	n/a Rectangle Rectangle n/a

2.1. Data Collection

Due to a high degree of privacy concern, we managed to only get 53 volunteers to participate in our experiment.These 53 volunteers installed our software and continuously collected the logging data for 5 to 7 days. The participants were permanent staff members and students of our institute and they all were regular computer users. Contrary to other research has our dataset the following properties: 1) Data collection was done in a complete uncontrolled environment therefore, it represents the user's natural computer usage behaviour; 2) To remove the hardware change effect on the natural behavioural pattern, the participants have performed this experiment on their own system; and 3) Predefined tasks or any instructions were not imposed on the users.

Our dataset is diverse in context of nationalities of our volunteers and the keyboard layouts they used. In this dataset we have the data from different nationalities (*i.e*. Chinese, Dutch, English, French, Indian, Iranian, Norwegian etc.) with different keyboard layouts (*i.e*. QWERTY, AZERTY etc.).

Table 3 shows the quantitative and qualitative comparison between our dataset and the dataset collected for previous research on CA.

Table 3. Dataset comparison with state-of-the-art research.

Ref.	# user	Duration (Days)	Task/Applications	Environment
[1]	10	5	Fixed	Controlled
[3]	31	1	Fixed	Controlled
[8]	20	Not given	Fixed	Controlled
[15]	24	56	Web Browser	Uncontrolled
Our	53	5-7	Uncontrolled	Uncontrolled

2.2. Data Separation

We split the dataset for each participant into three non-overlapping parts. The Training Dataset M is used to build and train the system, the Validation Dataset V is used for parameter adjustment of the algorithms used in this research and finally the Test Dataset T is used to test the system performance.

Let $S = [s_1, s_2, s_3, ..., s_n]$ be the datasets of the n biometric subjects (in our research $n = 53$). Furthermore let $M = [m_1, m_2, m_3, ..., m_n]$, $V = [v_1, v_2, v_3, ..., v_n]$ and $T = [t_1, t_2, t_3, ..., t_n]$ where $m_i \approx 35\% \; of \; s_i$, $v_i \approx 10\% \; of \; s_i$, and $t_i \approx 55\% \; of \; s_i$ for $i = 1, 2, 3, ..., n$.

3. Feature Extraction and Classification

In this section we will give a description of the biometric features and classifiers that are used in this study.

3.1. Keystroke Features

We converted the keystroke events into two different actions for our analysis. These actions are as follows:

1. `Single Key Action`, where the biometric feature is the *key hold time*. We have calculated the feature for the a `Single Key Action` by taking the time difference between *key down event* and the *key up event*.

2. `Key Digraph Action`, where the biometric features are the *Total Digraph Duration* (*i.e*. from first key press until the second key release), *Down-Down Time* (*i.e*. time between first key press and second key press), *Up-Down Time* (*i.e*. time between first key release and second key press and *Up-Up Time* (*i.e*. time between first key release and second key release) of a particular key digraph [2, 4]. We have applied a constraint for *Key Digraph Action* that the latency between two consecutive keys should be below 2000ms.

3.2. Mouse Dynamics Features

We converted the mouse events into four different actions for our analysis. These actions are as follows:

1. `Mouse Single Click Action`, where the biometric feature is similar to *Single Key Action*. We calculated the feature for the a `Mouse Single Click Action` by taking the time difference between *mouse down event* and the *mouse up event*.

2. `Mouse Double Click Action`, where the biometric features are similar to *Key Digraph Action*. The two consecutive mouse clicks considered to be as double click where the threshold for the latency is 1000ms.

3. `Mouse Move Action` can be formed by the sequence of mouse move events.

4. `Mouse Drag-Drop Action` is also a very similar to the *Mouse Move Action* but for this action first there has to be a mouse click down event followed by mouse move sequences and then mouse click up event. Similar to mouse move features we have calculated *Mouse Drag-Drop Action* and also added to the *Mouse Single Click Action* feature.

Our data collection software follows an efficient compression technique where it can only records the relevant mouse move sequences. This means that we can reconstruct the mouse curve with negligible error. Based on these mouse movement data, we can compute a variety of trajectory related features for mouse move and mouse drag-drop actions [6, 10, 12, 13]. In total we extracted 18 feature attributes for *Mouse Move Actions* and 24 feature attributes for *Mouse Drag-Drop Actions*. Figure 1, shows the Empirical Cumulative Distribution Function for significant Mouse trajectory related features. From the figure we can clearly understand that the users are separable from each other. All of these features are used for the first time in continuous authentication research.

3.3. Classification

In our study, we used 2 regression models and 1 prediction model in a multi-classifier decision fusion architecture [7]. For the regression models we have used Counter-Propagation Artificial Neural Network (CPANN) [17] and Artificial Neural Network (ANN) and for the prediction model we have applied Support Vector Machine (SVM). We have build separate classifier models based on the performed actions for each user. The score vector we use for further analysis is $(f^1, f^2, f^3) = (Score_{ann}, Score_{svm}, Score_{cpann})$ (see Section 4).

Feature selection [9] was applied on the training data of each of the users (see Section 4.1) before building the classifier models for *Mouse Move* and *Mouse Drag-Drop* actions. Let, $F = 1, 2, 3, ...m$ be the total feature set, where m is the number of feature attributes. The feature subset $A \subseteq F$ are based on the maximization of S_n with *Genetic Algorithm* as a feature subset searching technique where,

$$S_n = sup\left|MVCDF(x_g^A) - MVCDF(x_i^A)\right|$$

$$MVCDF() \rightarrow Multivariate\ Cumulative\ Distribution\ Function$$

$$x_g^A \rightarrow Genuine\ user\ feature\ subset\ data$$

$$x_i^A \rightarrow Imposter\ user\ feature\ subset\ data$$

4. Methodology

We are going to discuss the methodology followed to carry out research in this section.

4.1. Verification Process

In our research, we followed the leave-one-out testing process. This means that we have 1 genuine set of test data and 52 impostor sets of test data for each user. Based on the number of imposters considered in the training phase by using training dataset and parameter adjustment phase by using validation dataset, we used two verification processes which we will describe below.

Verification Process 1 (VP-1): In this verification process, the classifiers is trained with the training dataset M

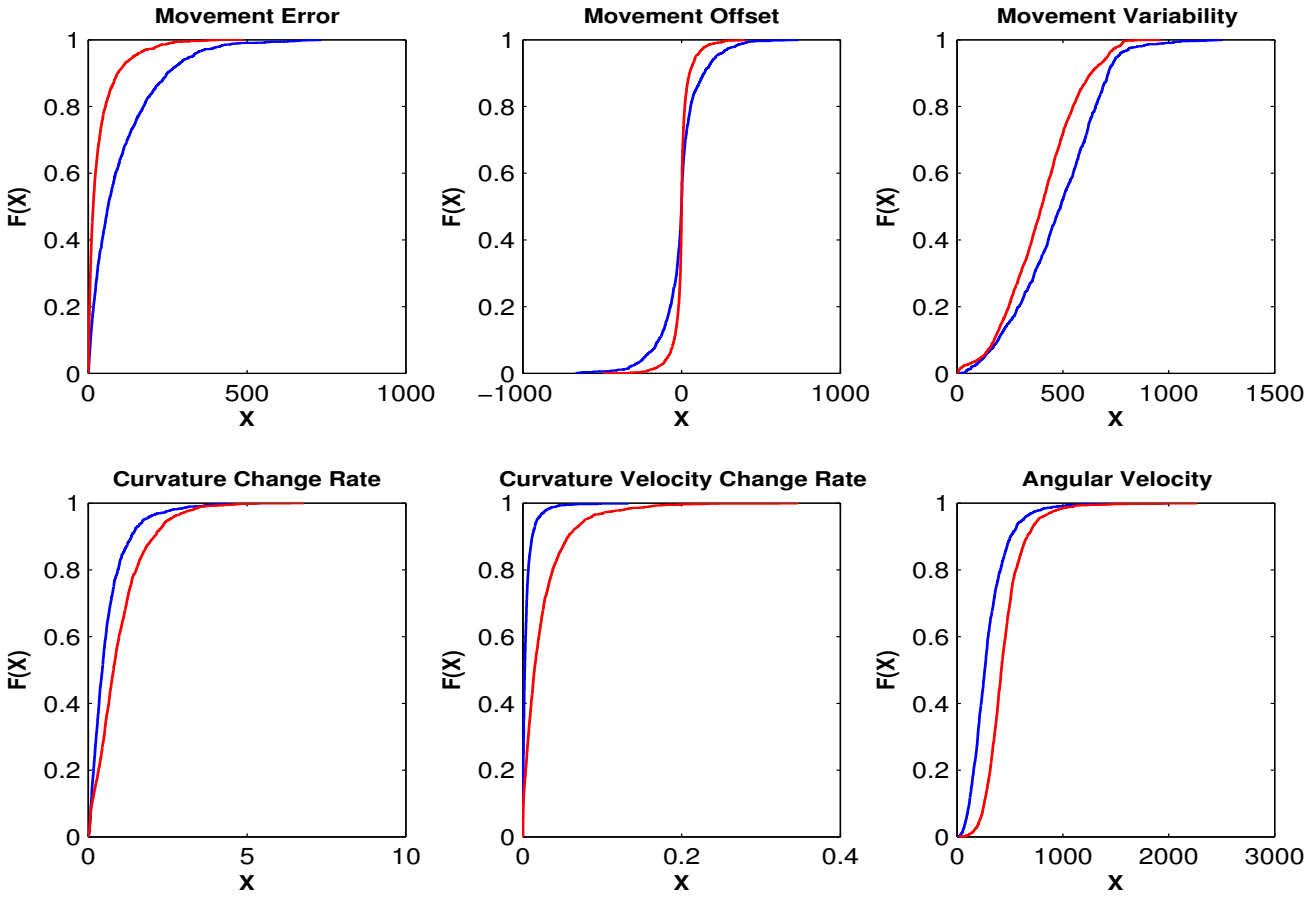

Figure 1. Empirical Cumulative Distribution Function for significant Mouse trajectory related features.

(see Section 2.2) from the genuine user, as well as data of 26 of the impostor users. In this verification process 50% of the imposter users are known to the system (*i.e.* genuine user and 26 impostors "known" to the system and 26 impostors completely "unknown" to the system.). Figure 2 explains this separation process for the first user, where $|m_1| \approx |IMP_{tr}|$. A similar technique was followed for the parameter adjustment dataset.

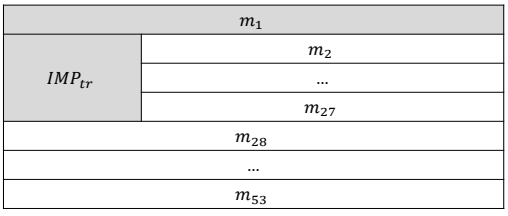

Figure 2. Data separation for *VP-1*. This is an example for User-1.

Verification Process 2 (VP-2): In this verification process is the training and testing of the system done by separate sets of impostor users. In this case we have two training datasets per genuine user. We do split the group of impostor users into 2 sets of 26 impostors. Figure 3 explains this separation process for the first user, where $|m_1| \approx |IMP^1_{tr}|$ and $|m_1| \approx |IMP^2_{tr}|$. First, we trained the classifiers with the first set of training data of the genuine user and the training data of the first set of 26 impostor users, exactly as we have done in *VP-1* (see Figure 3(a)). Next we tested this system with the testing data of the genuine user and all of the data of the second set of 26 impostor users. This process is then repeated with the second set of training data where the impostor users swapping roles, *i.e.* the data of the second set of 26 impostors will be used to train the classifiers, while it is tested with the data of the first set of impostors (see Figure 3(b)). In this verification process no imposter users are "known" to the system. A similar technique was followed for the parameter adjustment dataset.

4.2. Trust Model

Behavioural biometric data (e.g. keystroke dynamics or mouse dynamics) are generated from human motor-skills [16]. Therefore, there will be some natural variation in the user's data. This phenomena motives the researchers to use

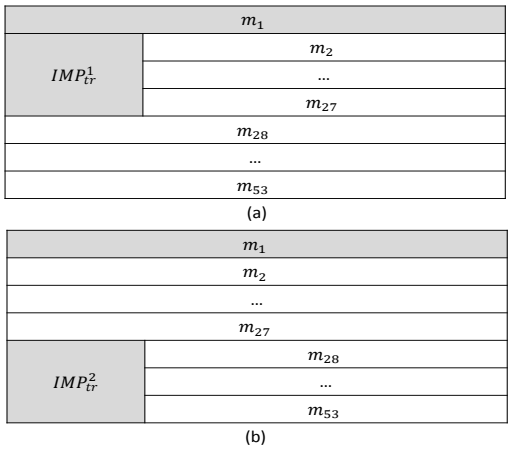

Figure 3. Data separation for *VP-2*. This is an example for User-1.

the *Trust Model* for behavioural biometric based continuous authentication [4]. We set the trust level to 100% trust in the genuineness of the user after static authentication. Now we will compute the score (similarity measure) for each and every action performed by the user (see Section 3 for the description of the performed actions) by comparing this action with the template of the true user. If the score (*i.e.* sc_i after i^{th} action) is high enough (*i.e.* $sc_i > A$) we will reward the current user (*i.e.* we increase the system trust), otherwise we will penalize the user (*i.e.* decrease the system trust). If the system $Trust$ drops below a pre-defined threshold ($Trust < T_{lockout}$) then the system locks itself and will require static authentication of the user to continue working. The amount of penalty/reward is calculated according to the Equation 1 and the current system trust is calculated according to the Equation 2. In Equation 2, we can see that the upper limit of the system trust is 100, to prevent a situation where an imposter user benefits from the high system trust obtained by the genuine user, before he/she hijacks the system.

$$\Delta_T(sc_i) = \min\{-D+D\times(\frac{1+\frac{1}{C}}{\frac{1}{C}+\exp(-\frac{sc_i-A}{B})}), C\} \quad (1)$$

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

In the Equation 1, parameter A is the threshold for penalty/reward, parameter C is the upper limit of the reward, and parameter D is the upper limit of the penalty. Finally parameter B is the width of the sigmoid (*i.e.* for which score value it will reach to the upper limit of the penalty/reward).

4.3. System Architecture

Figure 4, shows the complete system architecture of our proposed CA system. In this figure, the dotted lines represent the training phase and the solid lines represent the testing phase. We can see that the keystroke and mouse dynamics are separated because at any given time there will be either keystroke data or mouse data but not both at the same time. Not explicitly that our proposed architecture is not based on the modality fusion which is contradictory to the state-of-the-art research [1, 3, 8, 15].

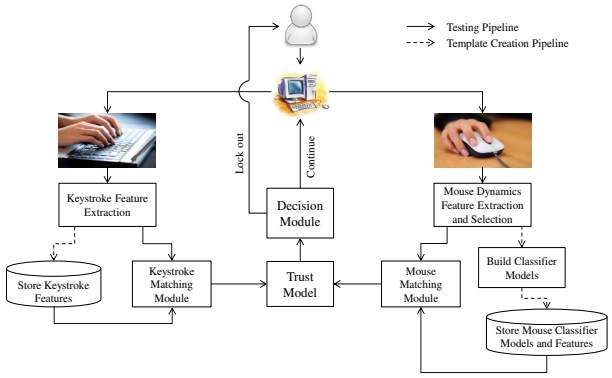

Figure 4. Proposed system architecture.

All the algorithmic parameters (*i.e.* A, B, C, and D in Equation 1) and the weights for the classifier fusion techniques dependent on the type of action and are optimized by *Genetic Algorithm* optimization techniques, where the cost function was to maximize $ANGA-ANIA$ (see Section 4.4 for description of ANGA and ANIA). In our research, we apply two classifier fusion techniques which are described below:

4.3.1 Score Fusion (SF)

In this fusion technique, from the 3 separate classifier scores, we calculate a single score that is used in Equation 1 (see Section 4.2) to calculate the penalty-reward. This is done in the following way: $sc_i = (\sum_{j=1}^{3} w_j f_i^j)/(\sum_{j=1}^{3} w_j)$ where, w_j are the weights for the weighted fusion techniques. Then the calculated penalty-reward goes to the Equation 2 to calculate the current system trust. The weights were optimized using *Genetic Algorithm* with parameter adjustment dataset.

4.3.2 Penalty-Reward Fusion (PRF)

In this fusion technique, from the 3 classifier scores we have individually calculate the penalty-reward from Equation 1. So, $\Delta_T(sc_i^1)$ represents the penalty-reward from Classifier-1, $\Delta_T(sc_i^2)$ stands for the penalty-reward from Classifier-2 and $\Delta_T(sc_i^3)$ is the penalty-reward from Classifier-3. Then we combine these with weighted fusion to calculate the system trust from Equation 2. Therefore, $\Delta_T(sc_i) = (\sum_{j=1}^{3} w_j \Delta_T(sc_i^j))/(\sum_{j=1}^{3} w_j)$ were, w_j are the weights

for the weighted fusion techniques. The weights are optimized using *Genetic Algorithm* with the parameter adjustment dataset.

4.4. Performance Measure

As mentioned before, each separate action performed by the user is taken into consideration in our testing phase. Therefore, is performance reporting by using EER not representing the system performance properly. For a CA system the objective is not simply to detect an impostor, but to dectect him or her after as few actions as possible. This will assure that an imposter can do as little damage as possible to the system. We will report our system performance in terms of *Average Number of Genuine Actions (ANGA)* and *Average Number of Impostor Actions (ANIA)* [11].

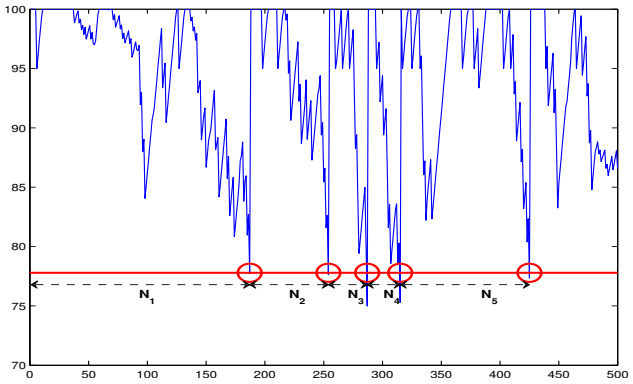

Figure 5. Change of Trust for impostor test set. X-Axis represents action number and Y-Axis represents the system trust.

In Figure 5, we see an example of how the system trust changes when we compare a given user template with test data of an impostor user. We see that the trust will drop 5 times below the lockout threshold ($T_{lockout}$ marked with a red line) within approximately 450 user actions. The value of ANIA in this example equals $\frac{1}{5}\sum_{i=1}^{5} N_i$. We can calculate ANGA in the same way, if the genuine user is locked out based on his own test data. For the experiment, whenever a user is locked out from the system, we reset the trust value to 100 and restart the computation to simulate a new session starting.

The goal is obviously to have ANGA as high as possible (idle will be ∞, that is, the genuine user is never locked out from the system) and the ANIA value must be as low as possible to make sure that an impostor user can do as little harm on the system as possible. Also, we want that all the imposters are detected by the system.

4.4.1 FNMR-FMR to ANGA-ANIA Conversion

In this section, we show how we can convert FNMR and FMR for a periodic authentication system in terms of ANGA and ANIA. Assume the $FNMR = p$ for m actions (*i.e.* the system operates on chunks of m actions). The genuine user can always do m actions and with the probability of $(1-p)$ he can continue and do m more actions. After that, again with the probability of $(1-p)^2$, he can do m more actions and so on. Therefore,

$$\begin{aligned} ANGA &= m + (1-p)\times m + (1-p)^2 \times m + \\ &(1-p)^3 \times m... \\ \text{So, } ANGA &= \frac{m}{1-(1-p)} = \frac{m}{p} \end{aligned}$$

Similarly if the $FMR = p$, then the imposter user can always do m actions and with the probability of p he can continue and do m more actions. After that again with probability p^2 he can do m more actions and so on. Therefore,

$$\begin{aligned} ANIA &= m + p \times m + p^2 \times m + p^3 \times m... \\ \text{So, } ANIA &= \frac{m}{1-p} \end{aligned}$$

Table 4. Previous research results with our conversion method.

Ref	# Users	FNMR	FMR	Blocksize	ANGA	ANIA
[1]	10	3%	3%	?	?	?
[3]	31	2.24%	2.10%	1100	49107	1124
[8]	20	17.80%	17.80%	?	?	?
[15]	24	8.21%	8.21%	40	487	44

In Table 4, we have converted the FMR and FNMR results of previous research on combined keystroke and mouse dynamics continuous authentication into ANIA and ANGA when possible. One requirement here is that the block size was reported which was not always the case.

5. Result Analysis

The results obtained from our analysis will be discussed in this section. We performed leave-one-out testing with the methodology discussed in Section 4. We will perform 2809 tests of which 53 are testing genuine user's test data against their own profile, while the remaining tests are impostor tests, *i.e.* data of imposters users is tested against the profile of a genuine user. Table 5, shows number of actions we have on average for testing for each of the users to measure our system performance.

Table 5. Average number of actions tested for each users.

Action Type	**Genuine Actions**	**Imposter Actions**
Single Key Action	17.4×10^3	92.2×10^4
Key Digraph Action	8.8×10^3	46.4×10^4
Mouse Single Click Action	6×10^3	31.7×10^4
Mouse Double Click Action	1.8×10^3	9.4×10^4
Mouse Move Action	8.2×10^3	43.3×10^4
Mouse Drag-Drop Action	0.074×10^3	0.42×10^4

We report the results in terms of the performance measure technique discussed in Section 4.4 for the person based

lockout threshold ($T_{lockout} = T_{us}$) which was optimized by using the *Genetic Algorithm* with the parameter adjustment dataset.

Understanding Tables 6 and 7: Each user can be categorize into 4 possible categories according their CA performance. The categories are divided based on genuine user lockout (*i.e.* $+$ if the genuine user is not locked out and $-$ if he is locked out) and non-detection of impostor users (*i.e.* $+$ if all impostor users are locked out and $-$ is one or more impostor is not detected).

- **All Positive (+ / +) :** This is the best category, where $EER = 0\%$ (*i.e.* the genuine user is never locked out from the system, and the system is able to detect all 52 impostors).
- **Positive vs. Negative (+ / -) :** In this category, $FMR = 0\%$ and $FNMR \neq 0\%$ (*i.e.* the genuine user is not locked out from the system but some impostors are not detected by the system).
- **Negative vs. Positive (- / +) :** In this category, $FMR \neq 0\%$ and $FNMR = 0\%$ (*i.e.* the genuine user sometimes locked out the system but the other hand are all impostors detected).
- **All Negative (- / -) :** This is the worst case category, where $FMR \neq 0\%$ and $FNMR \neq 0\%$. In this category, some of the impostors are not detected also the genuine user sometimes locked out by the system.

The column '# Users' shows the total number of users that fall within each of these categories. The 'ANGA' and 'ANIA' columns represents the Average Number of Genuine Actions (if genuine user lockout from the system) and the Average Number of Impostor Actions (if imposter user detected by the system). The '# Imp. ND' columns represents the total number of imposters were unable to detect by the system. As for example in the Table 6, the '+/-' category for SF and VP-1, we see that total 3 out of $3 \times 52 = 156$ impostor users not detected for 3 genuine users.

Table 6, shows the results we obtained from our analysis. We can see from this table that PRF performs better than SF for VP-1 where the best category users (*i.e.*'+/+' category) goes up from 44 to 46 but with a small increase of ANIA (*i.e.* 374 to 381). On the other hand, for VP-2, SF performs better than PRF where the best category users improves from 44 to 47, also with a lower value of ANIA (275 compared to 288).

We also analysed the performance of individual classifier and found that the number of users on '+/+' category vary from 25 to 32 (*i.e.* for ANN - 25, CPANN - 32 and SVM - 32) for VP-1 and in case of VP-2 these numbers vary from 34 to 36 (*i.e.* for ANN - 36, CPANN - 34 and SVM - 36). Therefore, we can say the multi-modal classifier fusion has significantly improved the system performance.

We applied the trust model proposed by Mondal *et al.* [11] on VP-1 with the score fusion technique. Table 7, shows the result obtained from this analysis. We can see that our proposed trust model performs better than the trust model used in previous research.

Table 7. Results for VP-1 with SF by using state-of-the-art trust model [11].

Category	# User	ANGA	ANIA	# Imp. ND
+/+	29		282	
+/-	3		3518	7
-/+	19	6593	349	
-/-	2	25203	1083	2

We would also like to test and compare our proposed scheme on another dataset used in the state-of-the-art research [1, 3, 8, 15] but unavailability of these datasets limits the possibility for comparison with these previous works. Table 3 and 4, show the qualitative and quantitative compression for previous research in this domain. The major limitation of the state-of-the-art research is that all the research in literature used a fixed chunk of data (*i.e.* chunks of m actions) for analysis. Due to that, even if they can achieve 0% EER rate, then the imposters can still perform m actions before getting detected by the system. In fact, they do not perform continuous authentication, but periodic authentication. This limitation can be overcome by our proposed scheme. According to our best knowledge, completely uncontrolled settings could be one of the reasons for failing to achieve the desired results, *i.e.* all 53 users in the '+/+' category.

Our major focus in this research was to develop a proper CA system, which can react on every single action performed by the user. A change of the pre-processing can influence the results (*i.e.* the different feature extraction process, different feature selection techniques and the choice of classifiers).

6. Conclusion

In this study, we presented a continuous authentication system based on behavioural biometrics. We evaluated it from a different perspective. We performed different verification process along with different fusion techniques to present quantitative and qualitative values of our research. The experimental was performed in such a way that it highly replicates a real world scenario. The proposed techniques are general enough that they can also be applied to, for example, swipe gesture data (*i.e.* for mobile devices) or any other biometric data that can be used for continuous authentication. To our best knowledge is this the first research that focuses on context independent continuous authentication with a combination of possible human computer interaction devices. We will consider soft biometrics for further inves-

Table 6. Result obtained from our analysis.

	Category	SF				PRF			
		# User	ANGA	ANIA	# Imp. ND	# User	ANGA	ANIA	# Imp. ND
VP-1	+/+	44		374		46		381	
	+/-	3		487	3	2		1241	3
	-/+	6	5322	542		5	9920	642	
	-/-								
VP-2	+/+	47		275		44		288	
	+/-	3		1580	3	5		2361	16
	-/+	3	4037	726		4	8876	405	
	-/-								

tigation in this area to improve the system performance and build a deployable security solution.

References

[1] S. Acharya, A. Fridman, P. Brennan, P. Juola, R. Greenstadt, and M. Kam. User authentication through biometric sensors and decision fusion. In *47th Annual Conf. on Information Sciences and Systems*, pages 1–6. IEEE, March 2013.

[2] A. A. Ahmed and I. Traore. Biometric recognition based on free-text keystroke dynamics. *IEEE Trans. on Cybernetics*, 44(4):458–472, 2014.

[3] K. O. Bailey, J. S. Okolica, and G. L. Peterson. User identification and authentication using multi-modal behavioral biometrics. *Computers & Security*, 43:77 – 89, 2014.

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

[5] C. Epp, M. Lippold, and R. Mandryk. Identifying emotional states using keystroke dynamics. In *The 2011 Annual Conference on Human Factors in Computing Systems (CHI'11)*, pages 715–724, 2011.

[6] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, and A. Schclar. User identity verification via mouse dynamics. *Information Sciences*, 201(0):19 – 36, 2012.

[7] T. K. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(1):66–75, Jan 1994.

[8] H. Jagadeesan and M. Hsiao. A novel approach to design of user re-authentication systems. In *IEEE 3rd Int. Conf. on Biometrics: Theory, Applications, and Systems, (BTAS'09)*, pages 1–6, Sept 2009.

[9] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. 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.

[10] I. S. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. In *The SIGCHI Conf. on Human Factors in Computing Systems (CHI '01)*, pages 9–16. ACM, 2001.

[11] S. Mondal and P. Bours. Continuous authentication using mouse dynamics. In *Int. Conf. of the Biometrics Special Interest Group (BIOSIG'13)*, pages 1–12, Sept 2013.

[12] S. Mondal and P. Bours. Continuous authentication using fuzzy logic. In *7th Int. Conf. on Security of Information and Networks (SIN'14)*, SIN '14, pages 231–238. ACM, 2014.

[13] B. Sayed, I. Traore, I. Woungang, and M. Obaidat. Biometric authentication using mouse gesture dynamics. *IEEE Systems Journal*, 7(2):262–274, June 2013.

[14] S. Shepherd. Continuous authentication by analysis of keyboard typing characteristics. In *European Convention on Security and Detection*, pages 111–114, May 1995.

[15] I. Traore, I. Woungang, M. Obaidat, Y. Nakkabi, and I. Lai. Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments. In *4th Int Conf. on Digital Home*, pages 138–145, Nov 2012.

[16] R. V. Yampolskiy and V. Govindaraju. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008.

[17] J. Zupan, M. Novič, and I. Ruisánchez. Kohonen and counterpropagation artificial neural networks in analytical chemistry. *Chemometrics and Intelligent Laboratory Systems*, 38(1):1 – 23, 1997.