A Contribution to the Continuous Implicit Authentication Process based on Touchscreen Input

Georgios Kalantzis
Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
gkalantz@ece.auth.gr

Gerasimos Papakostas
Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
gpapakos@ece.auth.gr

***Abstract*—Nowadays the vast usage of smart-phones, along with the sensitive information these could possibly contain for an individual user, raises the need for more effective and sophisticated authentication mechanisms and not only relying on the traditional ones, such as password/pattern input authentication. So there is a direction in research attempting to build machine learning models which are trained in biometric data stemming from hardware resources of the device i.e touchscreen, camera. In this paper we attempt to build an authentication process which can be employed as a biometric system. Given as input in a heuristical algorithm a series of predictions coming from machine learning models we exploit them in order to deduce a final right prediction of the identity of the user operating the device in a predetermined number of steps.**

***Index Terms*—User Authentication, Classification, Heuristics**

I. Introduction

Currently the modern biometric authentication systems employ intelligent machine learning systems in order to authenticate the legitimate users and prevent any possible unwanted usage from another person. So, there is a need to construct classifiers able to discriminate efficiently and correctly data coming from different people. They are trained in various biometric data from the smartphone and afterwards they take as input also biometric data and use them to make decisions. The classification scheme of the machine learning models that are constructed for this purpose usually is one-vs-all or novelty detection [1], [2]. The data for this problem consists of user specific classes and in this scheme we consider one user class and the rest of them as attackers class. The biometric data that are used in this methodology come from the touch-screen swiping data [3]. This specific dataset was created with the help of a real world application which accumulated a fair amount of data from many different users. Then we are able to generate a feature vector which consists of useful quantities that are capable of offering discrimination among the swipes of different users. Specifically the feature vector we employ is similar to that of Karanikiotis et al. [4] and eventually we train and test the machine learning models in the classification scheme mentioned above. The output of the machine learning models takes a value from the set $\{-1, 1\}$ where the value -1 declares that the observed swipe belongs to an attacker while 1 belongs to the original user. Hence, the metrics that are employed for the evaluation of the models essentially quantify the fraction of the false predictions to the total predictions. Thus, it is natural that a probability exists for the model to make a misprediction.

In this paper we make use of a series of these predictions which are contaminated in the sense that contain an unknown number of mispredictions and also there is no prior information about when a misprediction will occur. For this purpose we construct a parametric heuristic to update the values of a sequence where its short term tendency will determine the final prediction of the authentication. That is to say, we consider a decision sequence which its value depends on past values and input prediction. Also, we let two border values for the sequence that control the outcome of the authentication. Since the input to the algorithm is probabilistic we study the expected outcome of the decision sequence and we compute the desired parameters of the algorithm in order to ensure correct outcome in the authentication. With this methodology what we achieve is, given a machine learning model, in this classification scheme, we can conclude the correct identification of the mobile user in a predetermined number of swipes even if the efficiency of the model is not ideal.

The rest of this paper is organized as follows. At the section Related Work we make a brief overview of the research efforts considering the subject of building machine learning models for this specific problem and also others who attempt to authenticate users correctly given a collection of predictions. In the Methodology section we present analytically the authentication process as well as the decision sequence and the parameter computation. In Evaluation we employ sample machine learning models and we conduct several experiments to test the efficiency of the authentication process. Finally,in Conclusion we discuss several possible extensions and alternations of this approach.

II. Related Work

In this paper we extend the already done methodology of Karanikiotis et al. [4] in the field of continuous implicit authentication based on touch screen inputs. In particular in [4] the authors employ a big dataset coming from a real world application and not from a controlled environment which makes the data less biased and more random. Then they

extract features from touch swiping analytics and constructed a machine learning classification model. Finally, they employed the confidence level system in order to simulate the real life scenario of using the smartphone, essentially the value of confidence decreases or increases based on the predictions of the classifier. We also employ the same dataset with a slightly different feature vector. In general there are several studies that explore the continuous implicit authentication based on the interaction of a user with his touch screen. Feng et al [5] collected data from 40 diffrenet users with 53 features extraced for each one.They also used 3 different classification algorithms namely, Decision Trees, Random Forest and Bayes Net and achieved values of FAR and FRR 4.6% 0.13% respectively. Xu et al. [6] recruited 32 different users and by giving them specific instructions, acquired data for various gestures with touchscreen and extracting a big number of features. The authors employed multi-class classification with the use of Support Vector Machine(SVM) classifier. The results were pretty satisfying in the case were the number of users was quite small. Frank et al. [7] used kNN and SVM classifiers in a balanced dataset and achieved Equal Error Rate of 13%. The authors also combined the classification outputs of multiple strokes in order to boost their decisions and reduce EER significantly. Antal and Szabó [8] achieved very low EER specifically 4% and by using the average of sequences of swipes predictions scores they reduced EER even lower. A recent study [9] used deep learning models for the classification and combined data from different motion sensors, achieving very satisfying values on EER and Accuracy metrics. In the direction of using a series of predictions to deduce a final decision a more sophisticated approach instead of just averaging or combining predictions the authors in [10] employed a methodology based on the theory of DE-CumSum algorithm [11] from change detection theory in order to detect a possible intruder. So we observe that very efficient models have been built for the problem of classification based on touchscreen data and also there are attempts to make use of a collection of predictions to output a final prediction. The methodology of this paper falls in the context of the last approach.

III. Methodology

In this paper we build a system which takes as input a fraction of the touchscreen data(swipes) for training a machine learning model. Afterwards the rest of the swipes are also given as input to the system and the model makes predictions upon them and as a result metrics of the efficiency of the model are calculated. Then given these metrics we make a choice in advance of the number of the swipes we desire the authentication process to end and we compute the parameters of the heuristics. Finally, we pass series of predictions into the authentication algorithm and the system determines whether the swipes belong to an original user or to an attacker. Also, one notable fact is that the methodology is independent of the characteristics of the machine learning model and the only requirement is its efficiency metrics. This section is divided in two chapters where in the first one is the presentation of the authentication algorithm and its specifics and in the second one is the analysis of the algorithm and the parameter computation.

A. Authentication Algorithm

In this section we present the process we follow in order to classify a user given a series of swipes predictions. Basically, we define a decision sequence where its values are updated with the help of parametric heuristics which their logic was inferred experimentally after observing how a series of predictions containing false ones could be used in order to conclude a final right prediction. More analytically, let $s \subset \mathbb{Z}^{0+}$ represent the indexing of the swipes and p_s the sequence of predictions of given swipes which its value range is the two elements set $\{-1, 1\}$. Also, we define the important set $W = \{3w + 1 : w \in s\}$ which basically represents an indexing of the swipes with a step of length three. The reason for the choice of length three will be clarified later on this subsection. Then we can define the decision sequence which consists of a pair of two strictly positive parameters (a, b) as follows:

$$D_{s+1} = \begin{cases} D_s - a(2^{-fs}) + \frac{b}{2} I\{P\} e^{-s+1}, & \text{if } p_s = -1 \\ D_s + b(2^{-fs}) - \frac{a}{2} I\{Q\} e^{-s+1}, & \text{if } p_s = 1 \end{cases} \quad (1)$$

,where $I\{*\}$ represents the indicator function and it gives the value of 1 when the condition inside the brackets is satisfied and 0 otherwise and the arguments of I are two logical statements as $P = (D_{s+1} < D_s < D_{s-1} : s \in W)$ and $Q = (D_{s+1} > D_s > D_{s-1} : s \in W)$. The decision sequence consists of two important terms which essentially define the parametric heuristics mentioned in the beginning. The first one is the *update term* which is in one case $-a(2^{-fs})$ and the other $b(2^{-fs})$ and is responsible for increasing or decreasing the value of the sequence based on the prediction p_s . The contribution of this term slowly vanishes, in order to obtain fast decisions and to avoid long term sudden changes. The parameter f is just a constant given a fixed value of order 10^{-3} to ensure a slowly vanishing term. The second one is the *confirmation term*, i.e. $\frac{b}{2} I\{P\} e^{-s+1}$ and $-\frac{a}{2} I\{Q\} e^{-s+1}$ which essentially changes the value of the sequence in the opposite direction when two consecutive decreases or increases are observed. The idea behind this term is to avoid quick false decisions based on the occurrence of false predictions. The contribution of this term decreases exponentially and after some terms is considered negligible. The confirmation term takes place every three swipes, since it is based on the set W, and only if the logical statements P, Q are satisfied in this positions. The reason for the three length step of W is because we want to avoid having confirmations happening continuously and thus hindering the contribution of the updating term. Also, we want the confirmation term to occur due to consecutive increases or decreases only from the update terms and not from previous confirmation terms since the P and Q look out to the two previous values of decision sequence. Then we choose an initial value D_0 for the sequence. The final decision is made by choosing two border values D_{low}

and D_{high}, such that $D_{low} \leq D_0 \leq D_{high}$, and if the value of the sequence at some point surpasses the D_{high} value, i.e. $D_{s+1} \geq D_{high}$, then the final prediction is that the swipes belong to the original user, otherwise if $D_{s+1} \leq D_{low}$ at some point then the swipes belong to an attacker. The pseudocode for the algorithm is given in (Algorithm 1). Furthermore, there exist two important conditions governing the parameters (a, b), except being positive. The first one is derived by the fact that we do not want the algorithm to terminate in one step, i.e. $D_1 \geq D_{high}$ or $D_1 \leq D_{low}$, because that could lead to a false decision due to a possible misprediction positioned at first and also the confirmation term will never take place, since it needs at least two previous realizations of the sequence. The condition is given as

$$\begin{aligned} a &< D_0 - D_{low} \\ b &< D_{high} - D_0 \end{aligned} \tag{2}$$

The second condition ensures that the algorithm will not terminate, i.e. surpass a border value, after a confirmation term. Because then the confirmation is going to lose its meaning and its function in the logic of the algorithm. Namely the conditions is given as

$$\frac{4(D_{low} - D_0)}{3 + 2^{1-f}} < b - a < \frac{4(D_{high} - D_0)}{3 + 2^{1-f}} \tag{3}$$

The derivations of these inequalities can be found in the Appendix A

Algorithm 1 Decision Sequence(p_s)

for s **in** $0 \ldots length(p_s)$ **do**
  if $p_s = -1$ **then**
    $D_{s+1} \leftarrow D_s - a \cdot 2^{-fs}$
    if $s \in W$ **and** $D_{s+1} < D_s < D_{s-1}$ **then**
      $D_{s+1} \leftarrow D_{s+1} + \frac{b}{2} \cdot e^{-s+1}$
    end if
  else
    $D_{s+1} \leftarrow D_s + b \cdot 2^{-fs}$
    if $s \in W$ **and** $D_{s+1} > D_s > D_{s-1}$ **then**
      $D_{s+1} \leftarrow D_{s+1} - \frac{a}{2} \cdot e^{-s+1}$
    end if
  end if
  if $D_{s+1} \geq D_{high}$ **then**
    Original user identified!
    break
  end if
  if $D_{s+1} \leq D_{low}$ **then**
    Attacker identified!
    break
  end if
end for

B. Probabilistic Analysis and Parameter Computation

The input to our algorithm is probabilistic since a machine learning classification model makes false or correct predictions. Also we do not possess information for the moment of a right or a false prediction occurrence. But by the frequentist view, considering that we tested our machine learning model in a big number of swipes, we can estimate the probability of having a false prediction. Specifically, the probability of false prediction has two different values for two different cases. The first one is the case where the predictions are made from the original user's swipes and the probability is given by:

$$FRR = \frac{\text{original user rejected swipes}}{\text{original user total swipes}} \tag{4}$$

And the second one is the case where the predictions are from the attacker's swipes with false prediction probability:

$$FAR = \frac{\text{attacker accepted swipes}}{\text{attacker total swipes}} \tag{5}$$

So, now we are able to analyze the behavior of our algorithm in two separate cases. The first one is when the algorithm is getting as input a series of predictions from the user's swipes and the second one is the case where is getting input from an attacker.

1) User case: Since FRR (4) is the estimated probability of having a false prediction on the original user swipes, i.e. the probability that the prediction p_s takes the value of -1 on a swipe s ,we can define two sets of random variables which are related with the update and confirmation terms in every swipe. Since the swipes are considered independent from each other the random variables are also independent. Specifically let

$$U_s = \begin{cases} b, & 1 - FRR \\ -a, & FRR \end{cases} \tag{6}$$

be the identical and independent random variables associated with the update term in every swipe s which has the form of scaled bernoulli trials. And

$$C_s = \begin{cases} -\frac{a}{2}, & (1 - FRR) \cdot (1 - FRR) \\ \frac{b}{2}, & FRR \cdot FRR \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

be the i.i.d random variables associated with the confirmation term which has the form of two scaled independent consecutive bernoulli trials. Since the expression of the decision sequence (8) depends on the realization of the prediction p_s ,we can rewrite it in terms of the random variables we declared above, as follows:

$$D_{s+1} = D_s + U_s \cdot 2^{-f_s} + C_w \cdot e^{-w+1} \tag{8}$$

where $w \in W$. Now we can compute the decision sequence after k swipes in function of the initial value D_0. The formula occurs by induction as:

$$D_k = D_0 + \sum_{s=0}^{k-1} U_s \cdot T_s + \sum_{s \in W} C_s \cdot e^{-s+1} \tag{9}$$

where $T_s = 2^{-f_s}$. The reason for the above formulation of the decision sequence is that now we are able to compute the expected value of it after k swipes, which yields

$$E[D_k] = D_0 - G \cdot E[U_s] + N \cdot E[C_s] \tag{10}$$

where G is the result of the geometric series $\sum T_s$ and N is an exponent sum, with their exact expressions presented in Appendix B together with the derivations of all the above results. After the expected value has been computed we desire it to be greater than or equal to D_{high} because in the case of $D_k \geq D_{high}$ the algorithm will decide that the swipes belong to the original user.

2) Attacker case: Similarly in the case of an attacker FAR is the estimated probability of having a false prediction. Respectively we define similar i.i.d random variables related with the terms in the decision sequence in every swipe. Namely let

$$A_s = \begin{cases} -a, & 1 - FAR \\ b, & FAR \end{cases} \tag{11}$$

be associated with the update term. And

$$V_s = \begin{cases} \frac{b}{2}, & (1 - FAR) \cdot (1 - FAR) \\ -\frac{a}{2}, & FAR \cdot FAR \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

be associated with the confirmation term. In a similar fashion the expression for the decision sequence takes the form of the random variable:

$$D_{s+1} = D_s + A_s \cdot 2^{-f_s} + V_w \cdot e^{-w+1} \tag{13}$$

where $w \in W$. In order to evaluate the expected value after k swipes, we follow the same procedure as before with the difference that in the case of the random variable we substitute U_s with A_s and C_s with V_s. The expression is

$$E[D_k] = D_0 - G \cdot E[A_s] + N \cdot E[V_s] \tag{14}$$

In this case we desire the expected value to be less than or equal to D_{low} because if $D_k \leq D_{low}$ the algorithm decides that the swipes belong to the attacker.

Our goal is to calculate the parameter pair (a, b) for every user. So by letting $E[D_k] = D_{high}$ in the user's case and $E[D_k] = D_{low}$ in attacker's case we are able to create a 2×2 linear system of the form $Ax = d$ where $x = [a, b]^T$. Since we got the exact expressions for $E[D_k]$ in each case, with simple calculations we can derive the formulation of the linear system. By letting a function $f : \mathbb{R} \to \mathbb{R}$ as $f(x) = x \cdot G + \frac{N}{2} \cdot (1-x)^2$ the explicit form of the system is:

$$\begin{bmatrix} -f(FRR) & f(1 - FRR) \\ -f(1 - FAR) & f(FAR) \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} D_{high} - D_0 \\ D_{low} - D_0 \end{bmatrix} \tag{15}$$

So, we have the ability to compute the vector x in an inexpensive way, since we are dealing with a linear system in matrix form and lots of computer algebra systems exist for solving this efficiently. But, the solution vector will still be a function of G , i.e. $x = [a(G), b(G)]^T$. So, we can choose a value for k_{th} swipe we desire the process to terminate and easily transform it to the value G with the equation (20) in Appendix B and plug it in the solution vector to obtain the parameter pair (a, b). Also naturally this linear system has a condition of the existence of the solution which is provided from the determinant of A matrix. Since the FAR, FRR values are in the range $[0, 1]$ the only condition that applies is the obvious one $FAR + FRR \neq 1$. In practise the condition is not an obstacle in the search of solutions since the models satisfying this condition are impractical and not used. The general scheme of the system is depicted in 1.

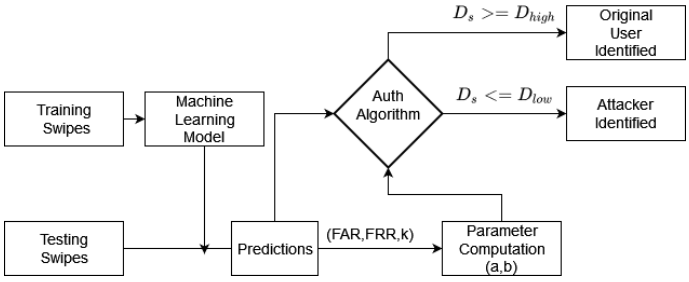

Fig. 1. Methodology Scheme

IV. Evaluation

In this section we present the results of experiments conducted to test the methodology of this paper. Initially, from the dataset [3] we extract data from twenty users in total with a sufficient number of swipes in order to obtain a better estimation of the misprediction probabilities and also to conduct as many experiments as possible. Based on the results of FAR, FRR we choose the k_{th} swipe that the algorithm will terminate. We let $D_0 = 0.55$, $D_{high} = 0.8$ and $D_{low} = 0.3$. These values were chosen so that $|D_0 - D_{low}| = |D_0 - D_{high}|$ in order to ensure fairness in the decision sequence. Also, the conditions (2,3) simplify. Then we divide the series of predictions into smaller segments which we will refer as sessions in order to simulate a real life scenario where someone makes a use of a phone in order to complete a particular task. Approximately, we consider the number of swipes required for completing a task in a smart phone is about 15 to 20, so the sessions will consist of twenty swipes for every experiment. The metric that we use for the evaluation is the percentage of sessions in which we correctly identify the user. The evaluation process has the following three stages.

A. Models Evaluation

In the first stage we employ a sample machine learning model for novelty detection, namely a Local Outlier Factor(LOF), and present the results of the algorithm in I and also we make use of other machine learning models as base models in order to show the independence of our methodology regarding the underlying model. Namely, the models are an Autoencoder for novelty detection and an ensemble model with One-Class Support Vector Machines combined with LOFs, their respective results are presented in II. In the first three columns is the indexing of the users, the FAR and FRR metrics. The column $\widehat{k}$ represents the estimated number of the swipes we chose in order for the algorithm to terminate. The k_U, k_A columns are the actual mean number of swipes of the

sessions that the algorithm terminated for the user and attacker respectively. The $U(\%)$, $A(\%)$ columns are the percentages of the sessions in which the algorithm correctly identified the user and attacker respectively. Finally, the last column $\#Exps_{U/A}$ represents the number of experiments conducted, specifically the left hand side of the slash$(/)$ is the number of experiments for the user while the right is the number for the attackers.

TABLE I
LOCAL OUTLIER FACTOR

ID	FAR	FRR	$\widehat{k}$	k_U	k_A	U(%)	A(%)	#Exps$_{U/A}$
1	0.46	0.15	8	8.27	7.48	96.6	87.64	30/1190
2	0.12	0.12	4	4.86	5.03	100	99.75	15/1205
3	0.28	0.15	5	6.15	5.7	100	97.59	13/1207
4	0.44	0.11	8	8.36	7.93	100	91.48	11/1209
5	0.52	0.16	8	8.1	6.33	90	86.61	10/1210
6	0.25	0.15	6	6.3	6.47	100	98.4	10/1210
7	0.45	0.16	6	6.6	5.15	90	87.19	10/1210
8	0.35	0.16	5	4.75	5.5	88.8	94.13	9/1211
9	0.34	0.14	6	5.88	6.51	100	97.27	9/1211
10	0.21	0.15	5	4.88	5.87	100	99.42	9/1211
11	0.55	0.12	10	10.09	8.01	91.3	83.56	23/919
12	0.57	0.12	10	9.3	7.79	90.9	82.39	22/920
13	0.37	0.12	8	8.82	7.97	100	94.7	17/925
14	0.26	0.14	7	7.25	7.48	100	97.19	16/926
15	0.4	0.14	8	7.61	7.48	100	92.35	13/929
16	0.4	0.15	7	7.16	6.73	100	91.82	12/930
17	0.41	0.17	8	7.91	7.48	100	91	12/930
18	0.41	0.1	7	6.09	7.2	100	89.68	11/931
19	0.43	0.16	9	10.09	8.29	100	89.9	11/931
20	0.07	0.16	5	5.8	5.75	100	100	10/931

As we can see from I the maximum deviation from the expected number of swipes is 2.21 which happens in the user with ID 12 who has the worst prediction capability since its pair of FAR, FRR metrics is very high, specifically FAR is equal to 57%. Generally, when the pair of FAR, FRR is getting lower the deviation also decreases. For instance, the user with ID 20 has very satisfying FAR, FRR metrics and its deviation from the expected number of swipes is less than one and its success rate is perfect in both cases. Furthermore the user success rate in most cases is ideal but for this result also plays role the fact that the number of experiments for the users is small in comparison with that of the attackers which is huge. Finally, we observe that for a mediocre model such as the user with ID 3 has very satisfying results, namely 100% success rate in the user experiments and 97.59% in the attacker experiments. In II we employed an Autoencoder for the users with the ID 1 to 8 and ensembling learning with Local Outlier Factors models and One-Class Support Vector Machines models for the rest.

TABLE II
AUTOENCODER-ENSEMBLE

ID	FAR	FRR	$\widehat{k}$	k_U	k_A	U(%)	A(%)	#Exps$_{U/A}$
1	0.38	0.06	7	7.46	7.79	100	96.76	15/1205
2	0.67	0.08	7	7.5	5.2	88.88	75.14	9/1211
3	0.53	0.16	9	7.22	7.17	100	82.98	9/1211
4	0.54	0.14	8	7.77	6.61	95.65	84.65	23/919
5	0.68	0.07	7	7.31	5.93	86.36	77.6	22/920
6	0.66	0.09	8	7.84	5.68	100	77.6	13/929
7	0.7	0.07	8	7.1	5.88	90.9	77.87	11/931
8	0.09	0.10	4	4.4	4.64	100	100	10/931
9	0.14	0.02	5	5	5.56	100	100	15/1205
10	0.29	0.03	5	5	5.78	100	99.33	9/1211
11	0.02	0.03	4	4.6	4.16	100	100	10/931

Initially, in II some users are missing due to the fact that for their particular FAR, FRR metrics the conditions 2,3 are not met since the values of the metrics are very high. In general we let the Autoencoder to be a non efficient model, in particular it is biased towards the positive predictions(user). So we observe while the FRR metric is very low the user percentage of success is also affected since a good model is defined by having both FAR, FRR metrics low. Furthermore, we employed a collection of classifiers, as mentioned above, in order to construct very efficient models and as we can see in this case correct identification is achieved in every session.

B. Model Efficiency Improvement

In the first stage the main evaluation metric of our methodology is the percentage of the sessions in which the algorithm correctly identified the user, both in the case of an attacker and the original user. These success rate metrics can be transformed back again to FAR and FRR metrics. Specifically, our methodology in the experiments makes predictions in every session, which consists of twenty swipes. So, in the case of the original user if a session does not recognize him as the user or recognizes him as attacker then the prediction in this session is false. Respectively, in the case of an attacker, the prediction in a session is considered false, if in these twenty swipes the algorithm does not recognize him. Thus, we can extend the FAR and FRR metrics, by letting the predictions to be represented by the final prediction in a session. Analytically, the FAR metric equals the quantity $100 - A(\%)$ and the FRR metric is $100 - U(\%)$. The main goal of this paper is to effectively enhance the prediction capability of the machine learning models. In order to show the improvement of the efficiency this methodology achieves, we make plots from the data of table I, which are the output of Local Outlier Factor model and the new results in comparison with the initial ones are depicted below. Figure 2 depicts the improvement of the FAR metric and figure 3 depicts the improvement of FRR.

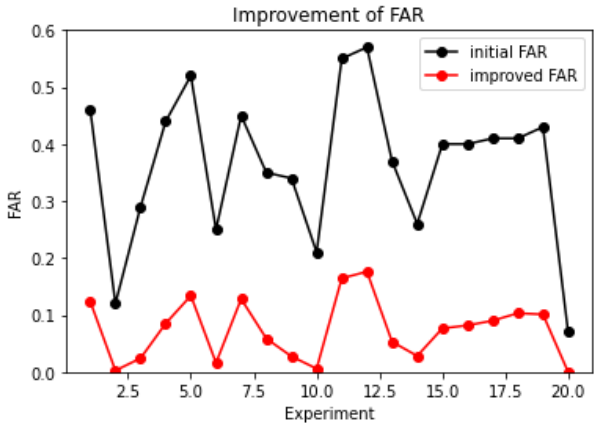

Fig. 2. FAR Improvement

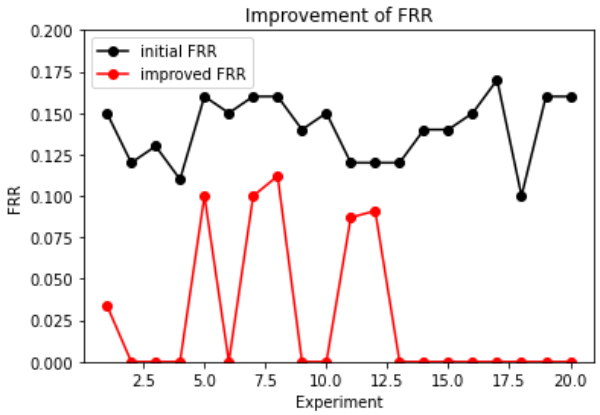

Fig. 3. FRR Improvement

We measure the improvement quantitatively as the mean absolute difference $\overline{\mathrm{d}}$ of the data points in 2,3. The improvement in the FAR metric is $\overline{\mathrm{d}} = 29\%$ while in FRR is $\overline{\mathrm{d}} = 11\%$. The reason for the lesser improvement in FRR is the fact that the initial values are already low in the scale of $10 - 20\%$.

In table III we include the mean FAR,FRR metrics computed in our experiments from the authentication algorithm with the final results of the same metrics of models created in other related studies. The mean of the metrics comes from I where the base model is just a sample Local Outlier Factor model.

TABLE III
COMPARISON TO OTHER STUDIES

Approach	FAR	FRR	# of Subjects
Yang et al. [12]	15%	8%	200
Draffin et al. [13]	14%	2.2%	13
Feng et al. [5]	9%	7%	23
Shen et al. [14]	5.01%	6.85%	48
Lee et al. [15]	2.8%	0.09%	35
Karanikiotis et al. [4]	-	4.7-5.7%	2,221
Ours with sample model	7.39%	2.62%	20

As can be seen from III our approach achieves very satisfying results also in comparison with other related studies and given the fact that the focus of this study is not to construct an efficient machine learning model, the results of III occured from a sample untuned model. Obviously, if the base model is replaced with a better model the final results will also be outstandingly improved.

C. Frequency of Attacker Accepted Swipes

In the first stage we considered the mean number of swipes that the algorithm terminated in the experiments among the sessions. In this stage, we present the explicit distribution of the attackers swipes for 4 users from I with IDs $\{2, 3, 11, 20\}$ where their corresponding FAR, FRR are depicted as labels in the histogram

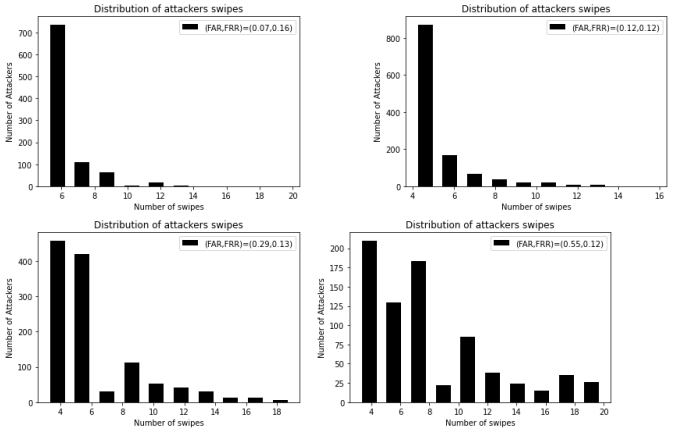

Fig. 4. Histograms of Attackers Swipes

The top two histograms in Figure 4, which belong to users with satisfying FAR, FRR have heavy left tail which is a positive result since almost $\sim 80\%$ of the attackers has completed less than 5 swipes. In the bottom two histograms in which FAR, FRR are not satisfying but still most of the attackers are distributed in the lowest number of swipes.

Following, we explicitly present in table IV the exact frequencies of the accepted attacker swipes for all the users from I to show that the most attackers were recognized in less than six swipes.

TABLE IV
FREQUENCY OF ATTACKER ACCEPTED SWIPES

Number of swipes	Frequency
≤ 6	11,678
7-9	3,734
> 9	4,403

We acknowledged that a potential attacker in a real scenario will not be able to cause any harm in less than six swipes using the smartphone. So the results of IV are very satisfying, since explicitly quantify the total number of accepted swipes across all the attackers and the overall tendency is that the attackers are recognized from the algorithm in a small number of swipes.

V. Conclusions and Future Work

In this paper we attempted to identify the imposter and the user of a smartphone device based on a sequence of swipes and experiments were conducted for a variety of FAR and FRR metrics on purpose in order to show that our methodology is efficient even for mediocre models. Specifically, we achieve more than 90% success rate even for a model with FAR $\sim 40\%$. As a future work we can consider also the variance of the decision sequence in both cases and attempt to minimize it and compute the parameters. Also then we will be able to study the concentration bounds of the sequence and thus ensure a better convergence to the expected behaviour. But this approach adds a lot of complexity to the system since variances create non-linear equations and finding a common solution is a cumbersome task also without simple guarantees or conditions for the existance of such solution.

Appendix A

We want the decision sequence to satisfy $D_{low} < D_1 < D_{high}$. We have two possible realizations of the term D_1 that is why we derive and two inequalities. For the first one, let $p_0 = -1$ then

$$\begin{aligned} D_{low} < D_1 < D_{high} &\Rightarrow D_{low} < D_0 - a < D_{high} \\ &\Rightarrow D_{low} - D_0 < -a < D_{high} - D_0 \\ &\Rightarrow D_0 - D_{high} < a < D_0 - D_{low} \end{aligned}$$

Since $a > 0$ and $D_0 < D_{high}$ we are only keeping the left hand side inequality, so the first condition is derived. In a similar way, if $p_0 = 1$ then

$$\begin{aligned} D_{low} < D_1 < D_{high} &\Rightarrow D_{low} < D_0 + b < D_{high} \\ &\Rightarrow D_{low} - D_0 < b < D_{high} - D_0 \\ &\Rightarrow D_{low} - D_0 < b < D_{high} - D_0 \end{aligned}$$

Similarly, $b > 0$ and $D_{low} < D_0$ we keep only the right hand side inequality.

The confirmation terms should not force the decision sequence to terminate the algorithm. We begin by constraining the first confirmation and then we will show that it is sufficient for constraining the rest. In the first case we assume that $p_0 = -1$ and $p_1 = -1$ then

$$\begin{aligned} D_{low} < D_2 < D_{high} &\Rightarrow \\ \Rightarrow D_{low} < D_0 - a - a \cdot 2^{-f} + \frac{b}{2} &< D_{high} \\ \Rightarrow D_{low} - D_0 < \frac{b}{2} - a \cdot (1 + 2^{-f}) &< D_{high} - D_0 \\ \Rightarrow 2 \cdot (D_{low} - D_0) < b - a \cdot (2 + 2^{1-f}) &< 2 \cdot (D_{high} - D_0) \end{aligned} \tag{16}$$

In the case where $p_0 = 1$ and $p_1 = 1$ we derive

$$\begin{aligned} D_{low} < D_2 < D_{high} &\Rightarrow \\ \Rightarrow D_{low} < D_0 + b + b \cdot 2^{-f} - \frac{a}{2} &< D_{high} \\ \Rightarrow D_{low} - D_0 < -\frac{a}{2} + b \cdot (1 + 2^{-f}) &< D_{high} - D_0 \\ \Rightarrow 2 \cdot (D_{low} - D_0) < b \cdot (2 + 2^{1-f}) - a &< 2 \cdot (D_{high} - D_0) \end{aligned} \tag{17}$$

By summing (16) and (17) we have

$$\frac{4(D_{low} - D_0)}{3 + 2^{1-f}} < b - a < \frac{4(D_{high} - D_0)}{3 + 2^{1-f}} \tag{18}$$

Appendix B

Initially we need to find an expression for the decision sequence after k in function with the constant initial value D_0 to easily calculate the expected value afterwards. So, by computing inductively the sequence (8) we have:

$$\begin{aligned} &D_1 = D_0 + U_0 \cdot T_0 \\ &D_2 = D_1 + U_1 \cdot T_1 + C_1 \\ &... \\ &D_k = D_0 + \sum_{s=0}^{k-1} U_s \cdot T_s + \sum_{s \in W} C_s \cdot e^{-s+1} \end{aligned}$$

In order to calculate the expected value we take the expectation operator in equation (9).

$$E[D_k] = E[D_0 + \sum_{s=0}^{k-1} U_s \cdot T_s + \sum_{s \in W} C_s \cdot e^{-s+1}]$$

By the linearity of expected value operator

$$\begin{aligned} E[D_k] &= E[D_0] + E[\sum_{s=0}^{k-1} U_s \cdot T_s] + E[\sum_{s \in W} C_s \cdot e^{-s+1}] \\ &= D_0 + \sum_{s=0}^{k-1} E[U_s \cdot T_s] + \sum_{s \in W} E[C_s \cdot e^{-s+1}] \\ &= D_0 + E[U_s] \cdot \sum_{s=0}^{k-1} T_s + E[C_s] \cdot \sum_{s \in W} e^{-s+1} \end{aligned} \tag{19}$$

Considering that $T_s = 2^{-fs}$ the sum $G = \sum T_s$ in equation (19) is a geometric series of the form $\sum_{k=0}^{n} ar^k$ with $a = 1$ and $r = 2^{-f}$ without the last element. So to calculate G :

$$\begin{aligned} \sum_{s=0}^{k-1} 2^{-fs} = \frac{1 - 2^{-fk}}{1 - 2^{-f}} \Rightarrow \\ G = \frac{1 - 2^{-fk}}{1 - 2^{-f}} \end{aligned} \tag{20}$$

Also in equation the term $N = \sum_{s \in W} e^{-s+1}$ is an exponent sum indexed with the set W which represents integers with a jump of three. The terms of the sum are approaching zero very fast so in our computations we consider only the first three terms and N takes the form $N = 1 + e^{-3} + e^{-6}$. Finally the equation (19) yields (14).

References

[1] C. M. Bishop, "Novelty detection and neural network validation," *IEE Proceedings-Vision, Image and Signal processing*, vol. 141, no. 4, pp. 217–222, 1994.

[2] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," *NASA STI/Recon Technical Report N*, vol. 93, p. 24043, 1993.

[3] M. D. Papamichail, K. C. Chatzidimitriou, T. Karanikiotis, N.-C. I. Oikonomou, A. L. Symeonidis, and S. K. Saripalle, "Brainrun: A behavioral biometrics dataset towards continuous implicit authentication," *Data*, vol. 4, no. 2, p. 60, 2019.

[4] T. Karanikiotis, M. D. Papamichail, K. C. Chatzidimitriou, N.-C. I. Oikonomou, A. L. Symeonidis, and S. K. Saripalle, "Continuous implicit authentication through touch traces modelling," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, 2020, pp. 111–120.

[5] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen, "Continuous mobile authentication using touchscreen gestures," in *2012 IEEE conference on technologies for homeland security (HST)*. IEEE, 2012, pp. 451–456.

[6] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*, 2014, pp. 187–198.

[7] 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 transactions on information forensics and security*, vol. 8, no. 1, pp. 136–148, 2012.

[8] M. Antal and L. Z. Szabó, "Biometric authentication based on touchscreen swipe patterns," *Procedia Technology*, vol. 22, pp. 862–869, 2016.

[9] S. Mekruksavanich and A. Jitpattanakul, "Deep learning approaches for continuous authentication based on activity patterns using mobile sensing," *Sensors*, vol. 21, no. 22, p. 7519, 2021.

[10] P. Perera, J. Fierrez, and V. M. Patel, "Quickest intruder detection for multiple user active authentication," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1341–1345.

[11] P. Granjon, "The cusum algorithm-a small review," 2013.

[12] L. Yang, Y. Guo, X. Ding, J. Han, Y. Liu, C. Wang, and C. Hu, "Unlocking smart phone through handwaving biometrics," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1044–1055, 2014.

[13] B. Draffin, J. Zhu, and J. Zhang, "Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2013, pp. 184–201.

[14] C. Shen, T. Yu, S. Yuan, Y. Li, and X. Guan, "Performance analysis of motion-sensor behavior for user authentication on smartphones," *Sensors*, vol. 16, no. 3, p. 345, 2016.

[15] W.-H. Lee and R. B. Lee, "Sensor-based implicit authentication of smartphone users," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2017, pp. 309–320.