An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices

Margit Antal
Sapientia University
Faculty of Technical and Human Sciences
Department of Mathematics and Informatics
Tirgu Mures, Romania
Email: manyi@ms.sapientia.ro

László Zsolt Szabó
Sapientia University
Faculty of Technical and Human Sciences
Department of Electrical Engineering
Tirgu Mures, Romania
Email: lszabo@ms.sapientia.ro

***Abstract*—In this paper we study keystroke dynamics as an authentication mechanism for touchscreen based devices. The authentication process decides whether the identity of a given person is accepted or rejected. This can be easily implemented by using a two-class classifier which operates with the help of positive samples (belonging to the authentic person) and negative ones. However, collecting negative samples is not always a viable option. In such cases a one-class classification algorithm can be used to characterize the target class and distinguish it from the outliers. We implemented an authentication test-framework that is capable of working with both one-class and two-class classification algorithms. The framework was evaluated on our dataset containing keystroke samples from 42 users, collected from touchscreen-based Android devices. Experimental results yield an Equal Error Rate (EER) of 3% (two-class) and 7% (one-class) respectively.**

***Keywords* —keystroke dynamics, touchscreen, one-class classification, mobile authentication, biometrics**

I. INTRODUCTION

An authentication system accepts or rejects the identity of a user based on the sample provided by the user. Several authentication schemes have been developed for sensitive data protection on mobile devices. However, the most widely used authentication mechanism is the PIN/password based one. A user enters his/her password into a mobile device by tapping buttons corresponding to the password characters. Keystroke dynamics uses the timing information related to the pressed keys and enables a two-level authentication scheme. At the first level the typed password is compared to the stored one, and then at the second level the typing rhythms are compared. This second level allows rejecting users with stolen passwords as typing rhythm is difficult to imitate. Touch sensitive displays (touchscreens) enable us to enhance keystroke dynamics. These devices are capable of measuring the extent to which the user presses the screen and the size of the pressed surface.

The authentication problem can be formulated as a two-class or as a one-class classification problem. In both cases the classifiers are trained separately for each user. The training data for two-class classification contain data from two classes, the positive and the negative class. Positive class is represented by the samples derived from the user and negative class can be formed by selecting some samples from the other users. In the case of the one-class classifier only samples from the given user are considered at the training stage. Both systems (one- and two-class classifier) are evaluated using both positive and negative samples.

In this paper, we exploit the new features deriving from touchscreens and based on these, we propose an authentication system which we then evaluate using both two-class and one-class classifiers. Results are presented using Receiver Operating Characteristics (ROC) and Detection Error Trade-off (DET) curves, indicating the confidence interval for both the Area Under Curve (AUC) and EER.

Most studies evaluate their authentication mechanism using both positive and negative samples for classifier training, or by using only positive samples. In this study we evaluate both, moreover we compare them in order to draw conclusions.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the methods used for our authentication schemes. Section 4 describes the datasets used in the experiments as well as experimental results and analysis. Section 5 concludes and discusses limitations of the study. All data collected for this paper are available online [1].

II. RELATED WORK

Several research papers are dedicated to the study of keystroke dynamics, moreover a few reviews are available [1], [2], [3], [4]. In the last few years we have seen a huge increase in the number of studies dealing with keystroke biometrics. The majority of research papers deals with keystroke dynamics on desktop computers. Some of the more recent papers report evaluation on datasets collected from touchscreen based devices.

In [5] the authors drew attention to using finger pressure for keystroke dynamics biometrics. As mobile touchscreen technology was not yet mature, the study used a touchscreen

[1] http://www.ms.sapientia.ro/ manyi/keystroke.html

978-1-4799-1780-8/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCS.2015.16

TABLE I
KEYSTROKE DYNAMICS AUTHENTICATION STUDIES CONDUCTED ON TOUCHSCREEN DEVICES. UD: UP-DOWN TIME; DD: DOWN-DOWN TIME; P: PRESSURE; PMIN: LOWEST PRESSURE; PMAX: HIGHEST PRESSURE; FA: FINGER AREA; D: DISTANCE; S: SPEED; A: ACCELERATION

Study	**#Users** **#Samples**	**Password**	**Features**	**Best result(s)(%)**
[5]	10 30	10 digits	P	EER: 1
[6]	80 25	4-8 digits	H+UD+FA+A	EER: 3.65
[7]	152 10	17 digits	H+UD+P+FA	FAR: 4.19 FRR: 4.59
[13]	20 12	7q56n5ll44	H+UD+D+S	FAR: 12.5 FRR: 39.4
[11]	10 100	4 digits	H+Pmin+Pmax	FAR: 14.1 FRR: 14.1
[8]	42 51	.tie5Roanl	H+DD+UD+P+FA	EER: 12.9

based laptop. Zheng et al. [6] were the first to study keystroke dynamics using touchscreen data from mobile devices. Besides touchscreen features, other features from accelerometer sensors were used. Only the typing rhythm of numerical passwords (PINs) was studied.

The effectiveness of touchscreen features was studied by Trojahn et al. [7] and Antal et al. [8]. Both papers conclude that these features increase keystroke based authentication system performance.

Draffin et al. [9] studied continuous authentication on touchscreen based devices. Impostor identification accuracy was 86% using the features extracted from 15 consecutive keys. Kambourakis et al. studied keystroke dynamics on Android devices. They collected a dataset from 20 users, proposed two new features, distance and speed, and then evaluated their authentication system using two-class classifiers.

Kang and Cho [10] studied keystroke dynamics authentication system using several input devices: PC keyboard, soft keyboard typed with a stylus pen and a touch keyboard. As they wanted to compare the performances of these input devices, they could use only time-based features. The PC keyboard based system performed the best. Several one-class classification algorithms, such as Gauss density estimator, Parzen window density estimator, k-nearest neighbor description, support vector data description (SVDD) and others were used for performance evaluation. Sen and Muralidharan [11] studied the typing of 4-digit PIN codes using touchscreen-based mobile devices. Authentication measurements were conducted on a dataset collected from 10 users and used both pressure and time-based features. Besides authentication, attack scenarios were also evaluated.

Xu et al. [12] studied continuous and passive authentication on touchscreen based mobile devices. They investigated the effectiveness of each feature separately and concluded that the touch area (finger area) was significant. Table I summarizes the results of the research papers studying keystroke dynamics related to password typing on touchscreen devices.

III. METHODS

The usage of a biometric authentication system supposes the earlier enrollment of the user. In the enrollment phase the system captures some samples which are used as the training data for the enrolling user. In our case, the user enters his/her password several times using the touchscreen device and the system creates a profile for the user. In the authentication (verification) phase, the user enters his/her password again. If the stored and the entered password are the same, a second decision is made based on the typing rhythm of the user. In order to verify the typing rhythm of the entered password, the features are initially extracted from the typing data. Then, based on the extracted feature vector and the user profile, the authentication system accepts or rejects the identity claim of the user.

A. Feature extraction

Let us suppose that a user presses n keys while typing a password. For each key the press and release timestamps are recorded resulting in the following time sequence

$$\{Pr_1, Re_1, Pr_2, Re_2, \ldots Pr_n, Re_n\} \tag{1}$$

From these timestamps we computed three types of time based features given by the equations 2, 3 and 4.

$$H = \{H_i | i = \overline{1,n}, H_i = Re_i - Pr_i\} \tag{2}$$

$$DD = \{DD_i | i = \overline{1,n-1}, DD_i = Pr_{i+1} - Pr_i\} \tag{3}$$

$$UD = \{UD_i | i = \overline{1,n-1}, UD_i = Pr_{i+1} - Re_i\} \tag{4}$$

At the moment of key press, the pressure and the finger area were also recorded (see Fig. 1) so we could use these data as features.

$$P = \{P_i | i = \overline{1,n}\} \tag{5}$$

$$FA = \{FA_i | i = \overline{1,n}\} \tag{6}$$

Besides the above first order features, we computed three second order features: the mean of the hold times, the mean of the pressures and the mean of the finger areas.

B. Authentication

Keystroke dynamics based authentication is a typical outlier detection problem. Given the keystroke data of a typed password the system has to decide whether the data belongs to the genuine user. This problem can be formulated as a two-class and as a one-class classification problem.

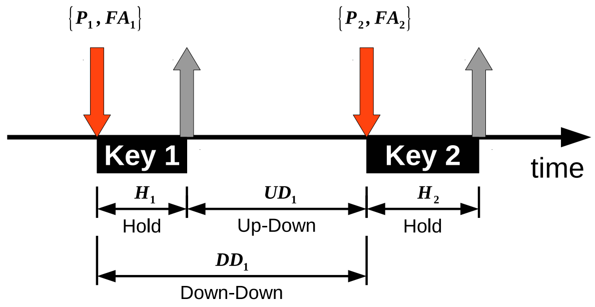


Fig. 1. Data acquisition. H: hold time; UD: Up-Down time; DD: Down-Down time; P: Pressure; FA: Finger Area

```
procedure MEASUREMENT($data, nFolds, nRuns$)
    for $run \leftarrow 1, nRuns$ do
        $randData \leftarrow randomize(data)$
        for $n \leftarrow 1, nFolds$ do
            $trainData \leftarrow trainCV(randData, n)$
            $testData \leftarrow testCV(randData, n)$
            train two-class classifier for each user
            evaluate the trained classifiers
        end for
    end for
end procedure
```

Fig. 2. Two-class classification measurement algorithm using n-fold cross validation

1) Two-class classification: In the case of two-class classification we call the data from the legitimate user positive samples and that from impostors we call negative samples. As our dataset contains data from several users and as each user typed the same password, one can easily select negative data for each user.

Various classifiers were used previously on keystroke dynamics datasets, including: statistical methods, decision trees, neural networks, fuzzy methods, support vectors, for review see [1], [2], [3], [4]. For this paper we selected some well-known algorithms implemented in Weka [14], covering various machine learning methods.

Bayesian Network is a probabilistic model that represents a set of random variables and their conditional dependencies using a directed acyclic graph. The nodes of the graph are the random variables and the edges represent conditionally dependent variables.

Nearest neighbors (kNN) is an instance based classification algorithm where a new instance label is decided by the k closest neighbors. This algorithm requires no explicit training as it stores all training samples and their labels.

The Random Forests classifier [3] is an ensemble learning method. This method combines several decision trees as weak learners to form a strong learner. Random forests run-times are fast and they work well with unbalanced and missing data.

Fig. 2 shows the general algorithm used for two-class classification measurements. n-fold cross-validation data partitioning was used and the measurement was repeated $nRuns$ times [15]. After the data was partitioned into training ($trainData$) and testing data ($testData$), for each user a separate classifier was built using the positive samples from the user and selecting 2 negative samples from each of the other users from the training part of the data. Evaluation of the trained two-class classifiers was carried out in order to obtain the data for the error curves. The testing set for each user consisted of the positive samples for that user and 2 negative samples from each of the other users from the testing part of the data.

Scores for positive and negative test samples were computed forming two sets, one for genuine users and one for impostors. Then a user independent threshold was scanned through the two sets of scores and the False Negative (FN) and False Positive (FP) rates computed for each threshold. Plotted as error curves, these values show the system performance.

2) One-class classification: One-class classifiers or anomaly detectors can provide a data description based only on the positive class. This approach agrees with our task: in a real application no one would hand his/her password to others in order to collect negative data. An extensive review of anomaly detection methods is presented in [16].

We used density and boundary method based one-class classifiers from Dd_Tools Data description toolbox for MATLAB, version 2.1.1 [17]. The nearest-neighbor data description (knndd) algorithm is based simply on the distance to the kth nearest neighbor, or some more advanced methods use averaged distances. The Parzen density estimator (parzendd) fits a Gaussian model around each of the training objects. The smoothing parameter is calculated by a maximum likelihood estimation. The mixture of Gaussians (mogdd) data description creates the model for the positive class by using k Gaussians. The parameters are optimized by the expectation maximization algorithm. Detailed presentation of the methods is provided in [18].

To perform one-class classification we used the same datasets and cross-validation folds as for two-class classification. Because of the class unbalanced test sets the error rate on the target and negative class were recorded distinctly for the positive and negative class. Although in the training phase some of the methods, such as the mogdd, can use some negative data efficiently, by all of these methods we used only the positive samples in the training phase, and both of the classes in the testing phase.

C. Evaluation metric

With respect to classifier performance evaluation we used error rates on the positive and negative class and metrics of detection performance as follows.

As the purpose of the task was authentication, measures of capturing the relation between the false acceptance rate (a negative sample is accepted) and the false rejection rate (a positive sample is rejected) were also used. Consequently we used classifiers which, in addition to the classification decision, yield comparable likelihood values (scores), and which indicate (by greater values) the measure of being part of the positive class. Scores permit the calculation of ROC curves (plotting the true positive rate against the false positive rate) and a variant of performance curves, which plot the two types of error rates: the false positive rate (miss) against the false negative rate (false alarm). These types of DET curves were introduced by Martin et al. [19] (except that we omit the normal deviate scale when plotting the curve).

The curves are plotted by varying the decision threshold of the classifier across all the ordered classifier output scores. Hence in the case of DET curves the error rate by which the miss and false alarm rates are equal (EER) can be determined. The calculated EER is considered the most important performance measure of our classifiers for the detection task.

The ROC and DET curves are not biased by skewed class distributions (we are limited to a small number of positive, and can select from a large number of negative samples). Therefore to compare classifiers by scalar values we used measures derived from these curves, the area under the ROC curve (AUC) and the EER.

10-fold cross-validation test runs were performed on two class datasets based on distinct users as positive users, resulting in a large source of variance. For this reason, we averaged ROC and DET curves across cross-validation folds and users, in order to present estimated mean values of performance. The mean of ROC and DET curves were calculated by using the $perfcurve$ function of the MATLAB (The MathWorks, Inc., Natick, Massachusetts) Statistics Toolbox. The calculations performed by this function are based on the methods presented in [20] called vertical and threshold averaging.

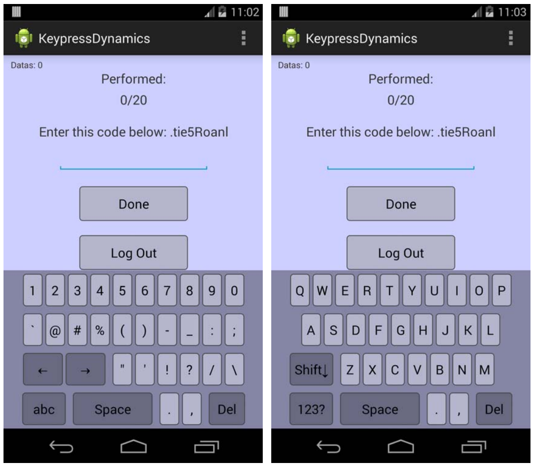
Vertical averaging takes samples for fixed X values of the curves, and calculates the estimated mean in a single dimension. Threshold averaging averages the samples in two dimensions separately for a fixed threshold applied to classifier scores (according to the algorithms presented in [20]). The $perfcurve$ function calculates pointwise confidence bounds for the estimated mean values, measuring the variance in our dataset (in one dimension for vertical averaging and two dimensions for threshold averaging). We used both of the averaging methods, and prefer vertical averaging since threshold averaging is sensitive to incommensurable classifier scores between models.

IV. Experiments

A. Data acquisition

Data collection was performed in a controlled environment using a specially designed Android application. The application used a software keyboard having three parts: lowercase letters, uppercase letters and numbers with special characters. The details of our software keyboard are shown in Fig. 3a and 3b.

Personal information such as gender, birth date and touchscreen experience level were collected. For touchscreen experience a scale containing 10 levels was defined. Table II provides the details of the data collection process and Table III contains the touchscreen experience levels of the users.



Performed:
0/20

Enter this code below: .tie5Roanl

Performed:
0/20

Enter this code below: .tie5Roanl

(a) (b)

Fig. 3. Screenshots from data collection

TABLE II
Details of data acquisition.

Information	Description
Users	42
Sample size	2142 (51 samples per user)
Number of sessions	2
Devices	Nexus 7 tablet Mobil LG Optimus L7 II P710
Password	.tie5Roanl
Password keystrokes	.tie<123?>5<abc> <Shift>R<Shift>oanl
Typing errors	Not allowed
Controlled aquisition	Yes
Age range	20-46 (average: 22.2)
Gender	24 male, 18 female
Touchscreen experience	levels shown in Table III

B. Datasets and features

1) Correlation of features: The correlation coefficients of all pairs of features are shown in the correlation matrix (Fig. 4). Dark red represents strongly correlated features and the green color indicate no correlation between features. The figure shows that the 14 hold time features are the most strongly correlated, followed by the 14 pressure features and the 14 finger area ones.

2) Informativeness of features: Weka is also capable of performing feature selection (Select attributes). If we combine an attribute evaluator with a Ranker search method, we obtain the ranking of the features, i.e. feature relevance with respect to the class. We used the GainRatioAttributeEval evaluator. This specific evaluator computes a value between 0 and 1 for each feature (gain ratio with respect to class [14]). The

TABLE III
Touchscreen experience levels of the users participated in data collection.

Level	0	1	2	3	4	5	6	7	8	9
#Users	1	1	0	4	2	8	8	10	6	2

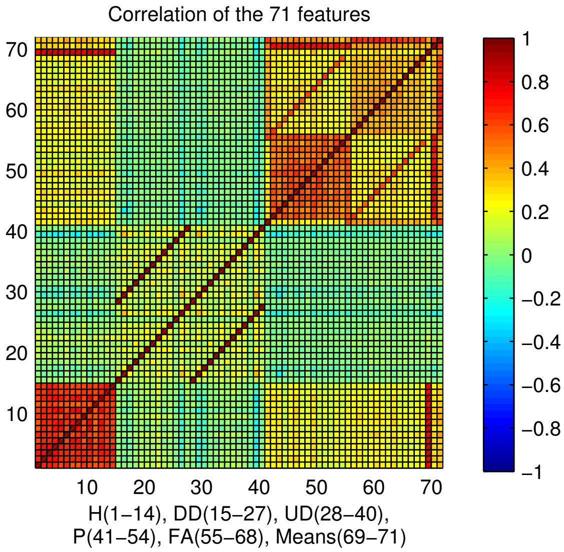


Fig. 4. Correlation matrix of the 71 features. H -hold time (14); UD - Up-Down time (13); DD - Down-Down time (13); P -Pressure (14); FA - Finger Area(14); Means - mean hold time, mean pressure, mean finger area (3)

TABLE IV
INFORMATIVENESS OF THE TOP 12 FEATURES

Information gain	Feature description
0.563	meanholdtime
0.543	meanfingerarea
0.512	meanpressure
0.482	fingerarea6
0.469	pressure9
0.466	holdtime12
0.458	pressure6
0.455	fingerarea2
0.452	pressure3
0.45	updown12
0.449	holdtime10
0.448	downdown3

larger this value, the more it determines the user. Using the GainRatioAttributeEval evaluator resulted in the ranking shown in Table IV.

3) Datasets: The datasets used in the evaluation measurements are summarized in Table V. The first dataset (dataset1) contains all features. The second dataset (dataset2) is used for historical reason, hold time features are the most frequently used features and they performed the best in comparison to the other time-based features [21]. We augmented the 14 hold time features with the first three best performing features (see Table V). The third dataset (dataset3) contains only the three mean values: mean hold time, mean pressure and mean finger area, as these were the most informative features (see Table IV). All datasets were normalized.

TABLE V
DATASETS AND THEIR DESCRIPTION

Name	Description
dataset1	71 features
dataset2	17 features: 14 hold time + 3 means
dataset3	3 features: 3 means

C. Experimental results

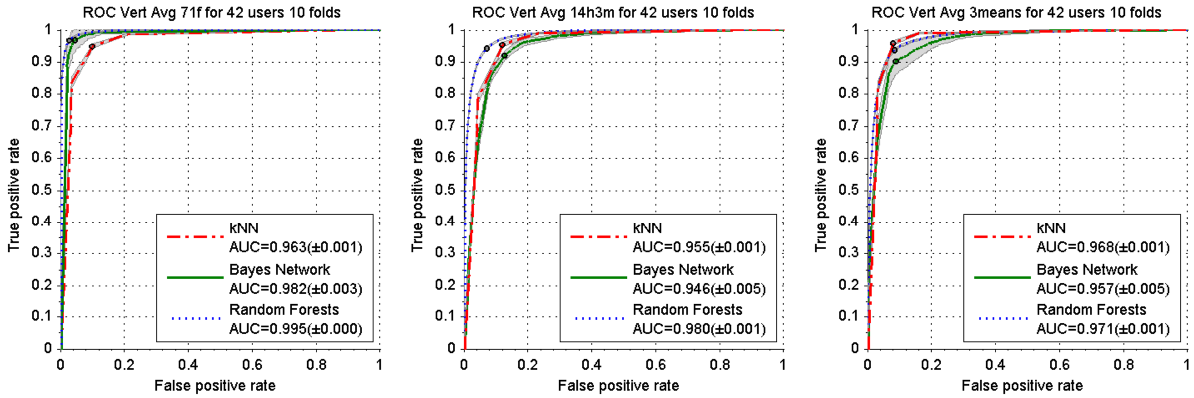
We used 10-fold cross-validation evaluation. The two-class classification algorithms were trained using 45-46 positive samples (90% of the positive samples) and 82 negative ones. The test files were composed of 5-6 positive samples (10% of the positive samples) and 82 negative ones. Results obtained for the three classifiers and the three datasets are shown in Fig. 5a – 5c and in Fig. 6a – 6c. In order to have a better visibility on the figures, error bars for pointwise confidence intervals were replaced by transparent boundary curves. The parameters of the classifiers were as follows: default value for the Bayes Network, $k = 3$ for the kNN and 100 trees for the Random Forests.

The same 10-fold cross-validation evaluation was applied in the case of one-class classifiers. The algorithms were trained using 45-46 positive samples (90% of the positive samples) and no negative ones. The test files were composed of 5-6 positive samples (10% of the positive samples) and 82 negative ones. Results are shown in Fig. 7a – 7c and in Fig. 8a – 8c. The parameters of the classifiers were as follows: default optimization for the width parameter of the parzendd, $k = 3$ for the knndd and 2 mixtures for the mogdd. The fraction of rejected objects on the positive class was set to 0.1.

The best EER (3.1%) was obtained by the Random Forests classifier for dataset1 (71 features). This was followed by the Bayes Network classifier (4.3% EER). K-NN proved to be the weakest among these three classifiers (8.3% EER). The same three classifiers produced EERs between 6.6-10.9% for dataset2 (17 features). It is worth noting the good performances obtained for dataset3 (7.1-9.8%), which contains only 3 features. Whats more, these three features are password independent (meanholdtime, meanpressure and meanfingerarea) and reflect the users' individual characteristics.

The detection performances (EER) of the selected one-class classifiers are similar. Results in higher feature spaces suffer from the dimensionality problem of one-class classifiers [18], causing poor results and larger confidence bounds. The EER is between 18-19% for all three classifiers on the full feature set and 7% on the means set with 1% confidence bound. Consequently, creating sets with reduced number of features is beneficial when training one-class classifiers for the keystroke dataset. The mogdd classifier performs slightly, but not significantly better on all feature sets.

Almost all one-class methods have better performance on classifying the negative class than the positive class (see Table VI). This follows from the robustness to outliers of one-class classifiers in Dd_tools, and not from the skewed class distribution (negative samples were not used during training).

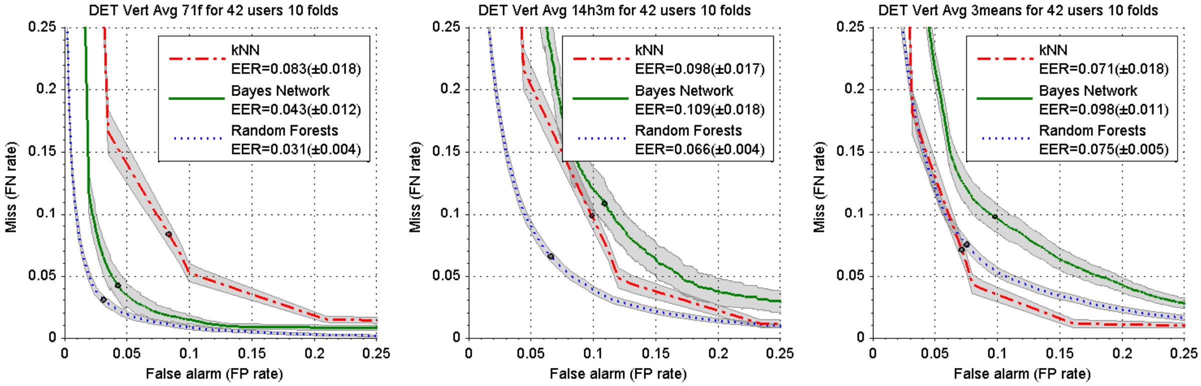


(a) Dataset1 – 71 features

(b) Dataset2 – 17 features

(c) Dataset3 – 3 features

Fig. 5. ROC curves for different two-class classification algorithms and different datasets.



(a) Dataset1 – 71 features

(b) Dataset2 – 17 features

(c) Dataset3 – 3 features

Fig. 6. DET curves for different two-class classification algorithms and different datasets.

TABLE VI
ONE-CLASS CLASSIFICATION. MEAN VALUES OF FN AND FP RATES ± SD.

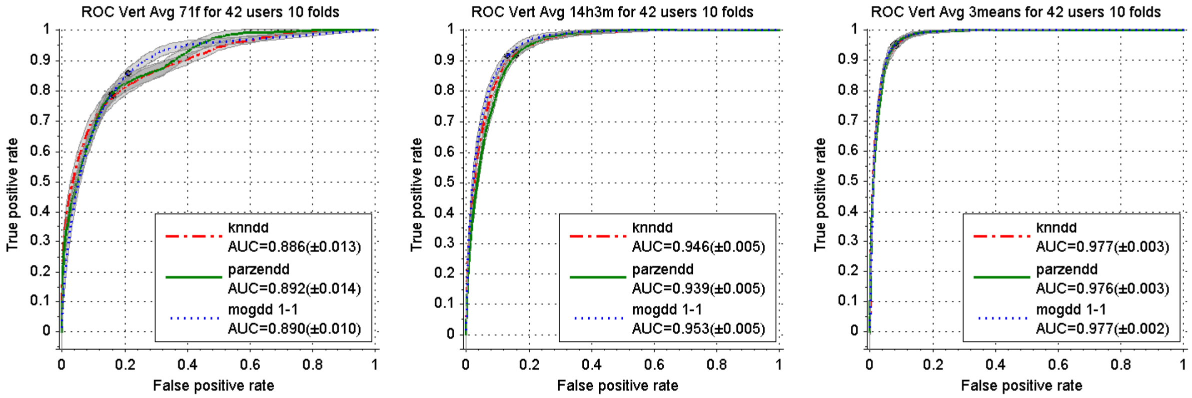
Dataset	Classifier	FN rate	FP rate
dataset1	knndd	0.11±0.15	0.20±0.23
	parzendd	1.00±0.00	0.00±0.00
	mogdd	0.31±0.28	0.06±0.07
dataset2	knndd	0.11±0.15	0.13±0.10
	parzendd	1.00±0.00	0.00±0.00
	mogdd	0.53±0.24	0.02±0.03
dataset3	knndd	0.11±0.15	0.05±0.05
	parzendd	0.27±0.21	0.02±0.03
	mogdd	0.16±0.16	0.04±0.03

V. CONCLUSION

In this paper we have presented the evaluation of our authentication framework with one- and two-class classification algorithms. The measurements were performed using our keystroke dataset collected from touchscreen devices. Authentication results were presented using ROC and DET curves, specially computed by vertical averaging of the test scores. Besides AUC and EER performance values we have consistently indicated the confidence interval.

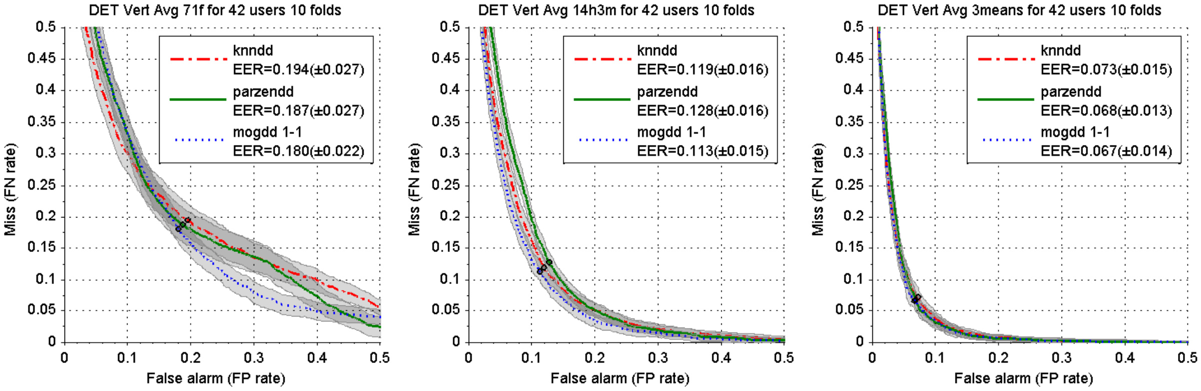
Our experimental results suggest that it is possible to reasonably authenticate users based on their typing rhythm and the way they press the touchscreen. Depending on the authentication scenario, there is 3% to 7% chance that the genuine user will be rejected or the impostor user will be accepted. As we expected, two-class classification algorithms performed better than one-class algorithms. However, one-class density methods perform well on small feature sets. Generally, negative samples are not available in the enrollment phase, therefore one-class classifiers are more suitable for use in real world authentication systems.

The main limitation of our study is that the subjects of our experiment were mostly students with touchscreen experience ranging from moderate to advanced. Another limitation of this study is the small sample size, which did not permit to some of the methods, such as the Parzen and Gaussians mixtures to have the best possible performance.



(a) Dataset1 – 71 features (b) Dataset2 – 17 features (c) Dataset3 – 3 features

Fig. 7. ROC curves for different one-class classification algorithms and different datasets.



(a) Dataset1 – 71 features (b) Dataset2 – 17 features (c) Dataset3 – 3 features

Fig. 8. DET curves for different one-class classification algorithms and different datasets.

Acknowledgment

The research and publication has been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TAMOP–4.2.2.C–11/1/KONV–2012–0004–National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

We would also like to thank all the students who have participated in the creation of the keystroke dynamics database.

References

[1] R. Giot, M. El-Abed, C. Rosenberger *et al.*, "Keystroke dynamics authentication," *Biometrics*, 2011.

[2] S. P. Banerjee and D. L. Woodard, "Biometric authentication and identification using keystroke dynamics: A survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012.

[3] N. Ahmad, A. Szymkowiak, and P. A. Campbell, "Keystroke dynamics in the pre-touchscreen era," *Frontiers in human neuroscience*, vol. 7, 2013.

[4] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, 2013.

[5] H. Saevanee and P. Bhattarakosol, "Authenticating user using keystroke dynamics and finger pressure," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE.* IEEE, 2009, pp. 1–2.

[6] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," Tech. Rep. WM-CS-2012-06, Tech. Rep., 2012.

[7] M. Trojahn, F. Arndt, and F. Ortmeier, "Authentication with keystroke dynamics on touchscreen keypads-effect of different n-graph combinations," in *MOBILITY 2013, The Third International Conference on Mobile Services, Resources, and Users*, 2013, pp. 114–119.

[8] M. Antal, L. Z. Szabó, and I. László, "Keystroke dynamics on android platform," 2014.

[9] B. Draffin, J. Zhu, and J. Zhang, "Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction," in *Mobile Computing, Applications, and Services*. Springer, 2014, pp. 184–201.

[10] P. Kang and S. Cho, "Keystroke dynamics-based user authentication using long and free text strings from various input devices," *Information Sciences*, 2014.

[11] S. Sen and K. Muralidharan, "Putting pressure on mobile authentication," in *Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference on*. IEEE, 2014, pp. 56–61.

[12] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *Symposium On Usable Privacy and Security, SOUPS*, vol. 14, 2014, pp. 187–198.

[13] G. Kambourakis, D. Damopoulos, D. Papamartzivanos, and E. Pavlidakis, "Introducing touchstroke: keystroke-based authentication system for smartphones," *Security and Communication Networks*, 2014.

[14] I. H. Witten, E. Frank, and A. Mark, *Hall (2011)." Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco. Retrieved, 2011.

[15] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, no. 2, 1995, pp. 1137–1145.

[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[17] D. Tax, "Ddtools, the data description toolbox for matlab," July 2014, version 2.1.1.

[18] D. M. J. Tax, "One-class classification; concept-learning in the absence of counter-examples," Ph.D. dissertation, 2001.

[19] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," DTIC Document, Tech. Rep., 1997.

[20] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[21] P. S. Teh, S. Yue, and A. B. Teoh, "Feature fusion approach on keystroke dynamics efficiency enhancement," *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 1, no. 1, pp. 20–31, 2012.