Evaluating Continuous Authentication in Smartphones

Mina M. Guirguis
Department of Computer Science
Winston-Salem State University
Winston-Salem, NC, USA
guirguismm@wssu.edu

Debzani Deb
Department of Computer Science
Winston-Salem State University
Winston-Salem, NC, USA
debd@wssu.edu

ABSTRACT
With the growing popularity of smartphones, continuous and implicit authentication of such devices via behavioral biometrics such as touch dynamics becomes an attractive option. Specially, when the physical biometrics are challenging to utilize, and their frequent and continuous usage annoys the user. This paper presents a touchstroke authentication model based on several classification algorithms and compare their performances in authenticating legitimate smartphone users. The evaluation results suggest that it is possible to achieve comparable authentication accuracies with an average accuracy of 91% considering the best performing model. This research is supervised by Dr. Debzani Deb (debd@wssu.edu), Department of Computer Science at Winston-Salem State University, NC.

Keywords
Touch dynamics; Behavioral Biometrics; Continuous authentication

1. INTRODUCTION
Over the last few years, the world has witnessed the explosive growth of consumers who are increasingly using their smartphones for anytime-anywhere computing and the enhancements of their daily lives. During the Covid-19 era, smartphones are regarded as lifelines and became absolutely crucial for distance learning and working. Since these devices store a mounting quantity of user's private and sensitive information, securing these devices from adversary attacks continues to be a significant concern for both manufacturers and users. Physical biometrics (face, fingerprints, iris, etc.) has often been promoted as the most secure means for log-in authentication for smartphones. However, there is a need for additional security measures after the initial log-in, known as continuous and implicit user authentication [1]. In such authentication, the system keeps continuously monitoring the user throughout their interactions with the device. The process is implicit such as all authentication is carried out in the background without interrupting the user or requiring any active user cooperation.

Conference'10, Month 1–2, 2010, City, State, Country.

DOI: http://dx.doi.org/10.1145/12345.67890

Vital physical biometrics are not appropriate for such implicit authentication as they need either full or partial collaboration from the users at regular intervals, which results in annoying the user.

Recent research has shown promising results in using behavioral biometrics [2] to verify users implicitly and continuously on smartphones. Today's smartphones are equipped with many sensors and accessories and could be used to extract user behavioral attributes such as touch dynamics, keystroke dynamics, and gait recognition. This paper focuses on touch dynamics [3,4], which captures how a user touches a touchscreen device and its usage on continuous and implicit user authentication.

In touch dynamics continuous authentication, the system continuously monitors the raw touch data and extracts touchstroke features. These include the screen area covered by the touch stroke, touch pressure, speed, velocity, and acceleration of the x, y-positions on the screen [3]. After observing the user behavior for a while, the system learns her touch dynamics by performing statistical analysis or using machine learning. Then, at a later time, after the initial log-in by using a password/pin or physical biometric, the system continuously compares current user behavior with the learned user model to make an authentication decision. The training phase in such authentication is different from typical classification as the only training data available is merely the smartphone owner's data. It is doubtful that many users will share a smartphone, and therefore the classifier can only assume the availability of the owner's data that belongs to a single class instance. The challenge is to train a classifier with two different predictions, such as owner and attacker, where the attacker instance does not belong to a prior-learned class [5]. Most of the prior works [3,6,7] on smartphone touchstroke authentication addressed this challenge by simulating one or more random users as attackers, and the authentication problem is naturally fitted as a binary-class classification problem, where the model is trained using a particular user's touchstone data as the owner's and the others' as attacker's.

This study describes a more robust behavioral biometric authentication based on four different algorithms to address this challenge and compare the accuracy. K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random forest (RF), and deep learning algorithms are deployed for identifying a user based on their touch dynamics data in a contiguous and implicit fashion.

The rest of this paper is organized as follows. Section 2 provides methodology including the description of our dataset, classification algorithms, and evaluation metrics. Section 3 presents results and Section 4 concludes the paper.

2. METHODOLOGY

2.1. Dataset

This study adopted Touchanalytics [3] dataset for experiments. There are 21,158 touch strokes in total, belongs to 41 subjects, which are collected from four different Android phones. For each stroke, 31 functional features can be derived [3]. Since each feature does not fall in the same range, they are standardized to the range [-1,1]. The authentication scenario considers two classes only, such as owner and imposter; however, the Touchanalytics dataset collects touchstrokes data that belongs to 41 users. A user-specific dataset is created for a legitimate user by extracting all of her touch data from the primary dataset, in order to keep classes balanced, as many samples from the negative study (other users) are obtained as there are samples of the legitimate user and these samples are added to make a complete user-specific dataset for such user. This user-specific dataset is divided into training (80%), and testing (20%) sets and is utilized during training and testing.

2.2. Classification Algorithms

This study utilized four different classifiers such as k-Nearest Neighbor (kNN), Support Vector Machine (SVM), Random Forest (RF), and Deep Neural Network (DNN). The choice for these classifiers was driven by various reasons as discussed below.

kNN is robust to work with and provides a fast classification. The kNN classifier takes every single observations and locates it in feature space with respect to all training observations. The classifier identifies the *k* training observations that are closest (based on Euclidian distance) to the new observation. Then, it selects the label that the majority of the *k* closest training observations have. This procedure requires no explicit training phase and the classifier merely stores all training observations and their labels in order to make predictions.

Support vector machines are popular and powerful binary classifiers. SVMs divide the feature space by a hyperplane such that the margin between the two classes is maximized, i.e., SVMs squeeze a maximally thick layer between the boundary observations of both classes, known as support vectors. In contrast to kNN, SVM generalizes from the observed data, i.e., it does not store the individual observations once the training is performed and only saves the decision hyperplane. For more robustness against outliers, a small number of boundary observations are tolerated within the margin. A parameter C controls the trade-off between maximizing the margin and minimizing the number of such exceptions. For classes that are not linearly separable in feature space, the standard scalar products involved in the computation of the hyperplane can be replaced with 'kernels'. Kernels implicitly relocate the problem in another high-dimensional space where the classes are separable. In the same step, the kernel maps the found hyperplane back to feature space. The presented research used a Gaussian radial-basis function (*rbf*) as the kernel, parameterized by the width parameter *gamma*.

Random Forest is an ensemble tree-based learning algorithm. The RF Classifier consists a set of decision trees, each of them built over a random extraction of the observations from the dataset and a random extraction of the features. Not every decision tree in the set utilizes all the features or all the observations in the training dataset, and this guarantees that the trees are less correlated and more independent, and therefore less prone to over-fitting. Each tree uses a sequence of yes-no questions based on a single or combination of features in order to divide the training observations. At each node, the tree divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket. Therefore, the importance of each feature is derived from how "pure" each of the buckets is. The most widely used impurity measure is the *Gini impurity*, which is also utilized in this study. The classifier aggregates the votes from different decision trees to decide the final class of the test object. Random forests are one of the most popular machine learning algorithms because of the good predictive performance and their resistance to outliers.

2.3. Evaluation Metrics

The classifiers performances are evaluated using various standard evaluation metrics such as Precision, Recall, F1 and ROC score. In this study, Precision is the ratio of correctly authenticated *users* observations to the total predicted *user* observations. Recall is the ratio of correctly predicted *users* observations to all actual observations with *users* labels. In other words, Precision and Recall are all interested in predicting the true answer of the positive label. F1 score takes both Recall and Precision into account, hence can be considered as a weighted average of them, and therefore it provides a useful accuracy indicator. The ROC curve is another common tool used with binary classifiers. The ROC curve plots the *true positive rate* (another name for Recall) against the *false positive rate* (FPR). The FPR is the ratio of negative instances that are incorrectly classified as positive. To visualize the performance of the classifier, Receiver Operating Characteristics (ROC) curve and Precision-Recall (PR) curves are introduced.

3. RESULTS

Five subjects are randomly selected from the Touchanalytics dataset for experimental evaluation. Each subject's touch strokes (first column in Table 1,2,3) are extracted as legitimate data, and a same number of other users' touchstrokes are added as fraud data to make the class balanced. The training and testing are performed for each authentic user individually, and the performances for each user in terms of Precision, Recall, F1 are shown in Table 1, 2 and 3. The maximum and minimum performances achieved for each metric across all subjects are highlighted in the table, along with the mean and median of all metrics. It is evident from the tables that the random forest algorithm is performing best in authenticating users closely followed by the support vector machines. The Precision scores are relatively high for RF classifier, with mean: 0.91 and median: 0.94. The Recall and F1 values are similar for all the three classifiers. These results indicate that it is possible to achieve acceptable authentication accuracies with touch dynamics data. The results in Table 1, 2, and 3 further reveal that the proposed system comparatively performs better when there are more data available to learn and to generate from.

Table 1. Predictions Results for KNN

Subject ID	Subject instances	Precision	Recall	F1
2	1230	**.92**	**.94**	**.93**
3	759	**0.77**	0.87	**0.81**
11	445	0.90	0.92	0.91
16	382	0.83	0.95	0.88
4	241	0.91	**0.80**	0.85
Mean		**0.87**	**0.90**	**0.88**
Median		**0.90**	**0.92**	**0.88**

Table 2. Predictions Results for SVM

Subject ID	Subject instances	Precision	Recall	F1

2	1230	**.94**	.94	**.94**
3	759	**0.80**	0.89	**0.84**
11	445	0.89	0.89	0.90
16	382	0.85	**0.96**	0.90
4	241	0.93	**0.86**	0.90
Mean		**0.88**	**0.91**	**0.90**
Median		**0.89**	**0.89**	**0.90**

Table 3. Predictions Results for RF

Subject ID	Subject instances	Pre.	Rec.	F1
2	1230	**.98**	**.92**	**.95**
3	759	**0.79**	0.89	**0.83**
11	445	0.94	0.88	0.91
16	382	0.87	**0.81**	0.84
4	241	0.94	0.88	0.91
Mean		**0.91**	**0.88**	**0.89**
Median		**0.94**	**0.88**	**0.91**

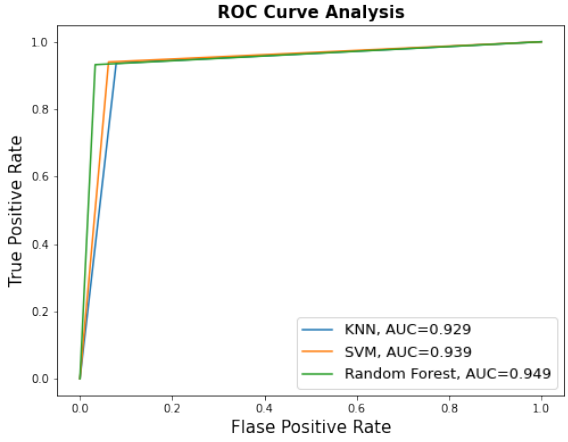

Figure 1: ROC accuracies for Three Models

ROC analysis is considered for all the classifiers. Figure 1 shows the ROC curves and the corresponding AUC values of all models. The ideal point in ROC space is the top-left corner. AUC is an important statistical parameter for evaluating classifier performance: the closer AUC is to 1, the better overall performance of established classifier. In the current work, as shown in Figure 1, the AUC value of RF classifiers is .949 for subject 2, which is higher than the other classifiers with a margin (2% or more), indicating that the RF classifiers achieves better performance than the other classifiers. Figure 2 shows the Precision-Recall curves (PR-curves) for all classifiers. PR-curve is a very widely used evaluation method in machine learning. In general, the closer the curve is to the top-right corner, the more beneficial the tradeoff it gives between precision and recall. The PR-curve in Figure 2 shows the superiority of random forest model in minimizing the number of false positives while ensuring high classification accuracy.

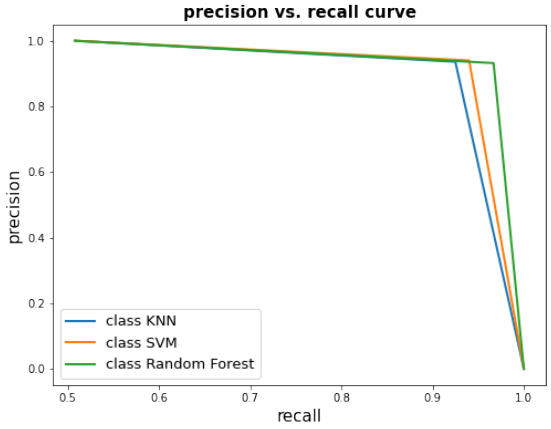

Figure 2: Precision Recall curves for Three Models

4. CONCLUSION

This paper presents a touchstroke authentication model. Given a small subset of a legitimate user's touchstroke data during training, the presented can accurately authenticate the user. This paper presents a touchstroke authentication model based on several classification algorithms and compare their performances in authenticating legitimate smartphone users. The evaluation results suggest that it is possible to achieve comparable authentication accuracies with an average accuracy of 91% considering the best performing model. The future works will focus on fine tuning the model in order to achieve better accuracies and investigating the impact of posture variation on the presented authentication.

ACKNOWLEDGMENT

We would like to acknowledge the support provided by NSF award # 1900087.

REFERENCES

[1] V. M. Patel, R. Chellappa, D. Chandra, B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges". IEEE Signal Processing Magazine, vol. 33, issue. 4, pp. 49–61, 2016.

[2] A. Mahfouz, T. M. Mahmoud, and A. S. Eldin, "A survey on behavioral biometric authentication on smartphones," Journal of Information Security and Applications, vol. 37, pp. 28–37, 2017.

[3] 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, 2013.

[4] Z. Sitova, J. Sedenka, Q. Yang et al., "HMOG: new behavioral biometric features for continuous authentication of smartphone users," IEEE Transactions on Information Forensics and Security, vol. 11, no. 5, pp. 877–892, 2016.

[5] R. Domingues, M. Filippone, P. Michiardi and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses", Pattern Recognition, vol. 74, pp. 406-421, 2017.

[6] A. Roy, T. Halevi, and N. Memon. "An hmm-based behavior modeling approach for continuous mobile authentication", In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3789–3793. IEEE, 2014.

[7] A. Serwadda, V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on, pp. 1–8, 2013