Machine Learning Assisted Brute-force Attacks on Touch Input Biometric

Faith C. Obasi
fobasi@uoguelph.ca
University of Guelph
Guelph, ON, Canada

Danyal Namakshenas
dnamaksh@uoguelph.ca
University of Guelph
Guelph, ON, Canada

Abstract

Providing the best methods of authentication and continuously improving them is of the utmost importance in the pursuit of security and usability. There has been chatter about touch biometrics being integrated into touchscreen devices to provide a hidden layer of security alongside passwords. It is also being theorized to serve as a method for continuous authentication. However, these methods have been proven to possess some vulnerabilities like smudge attacks, brute-force attacks, and spoofing attacks. Therefore, we propose an experiment to investigate the viability of touch biometrics as a form of authentication through the process of applying machine learning methods to classify the touch patterns of different users and performing a brute-force attack against each user. The proposed method comprises of different classifiers which yielded results indicative of the fact that touch biometrics is not a secure enough method for primary or secondary authentication.

Keywords:

Authentication, biometrics, touchscreen, brute force, machine learning

1 Introduction

The typical methods of authentication on mobile devices involve some sort of entry point into the system. The user is usually required to provide the correct password or pin to be granted access to the system. While the bulk of authentication schemes are governed by these entry point-based methods, they possess some weaknesses in terms of security and utility. In the matter of utility, the traditional methods of authentication have proven to be a bit cumbersome and inconvenient in the sense that users need continuously authenticate themselves rather frequently when using their hand-held touch screen devices, which may compel users to use easy passwords and pins which poses a security threat to them [1]. Furthermore, regarding security, some studies have shown that the usual methods of authentication are vulnerable to smudge attacks and the device cannot differentiate between a fake and real user when the authentication step is completed successfully [2].

There has been some buzz about the use of behavioral biometrics like touch patterns to bridge the usability and security gap in traditional authentication schemes. The methods have been proposed as a passive authentication method to provide an extra

layer of security in conjunction with passwords and pins. It is also supposed to provide continuous authentication by monitoring the user's interaction with their smart phones. This solves the utility problem of user's being frequently locked out of their devices by having an authentication scheme running in the background that continuously authenticates the user based on the unique behavior of the user interacting with their mobile devices. However, this approach to authentication poses threats of its own like mimicry and being vulnerable to brute-force attacks.

In this paper, we propose an experiment to investigate and analyze the credibility of touch inputs as a method of authentication. We classify touch input data of 31 users using artificial learning. We also attempt to perform an attack in which an attacker can delude the trained machine learning models by feeding the models the input of a different user chosen from 10 of the entire 41 users in the database to test if the model will accept the touch behavior of an illegitimate user as a legitimate user. The aim of this is to determine whether brute force methods could make the trained AI interpret a real user as a fake user and ultimately determine if touch inputs is enough to provide security to mobile devices. However, using brute-force methods is time consuming and costly to implement also, in the real world, an attacker will not go through the rigors of performing a brute force attack. He or she will simply try to mimic the victim's patterns. Therefore, we propose a dictionary attack instead.

2 Related Work

The use of human behaviors and characteristics as a way of authentication and verification has been ongoing research field for quite some time now [3]. Typically, it comprises of two categories which are the behavioral and physiological biometrics. The physiological biometrics are the wildly adopted authentication method among the two mentioned, it includes fixed physical attributes like fingerprints, facial features, DNA etc. Behavioral biometrics on the other hand, is still a developing method, theoretical if you may, it involves the unique way a user performs different activities, like typing, walking, speaking, etc.

As far back as the 90's, experiments to determine whether behavioral biometrics could be used as a form of authentication to supplement the conventional authentication methods (passwords, pins). One of the earliest research projects was carried out by Clark et al. in [3] conducted an experiment to explore the feasibility of using keystroke dynamics on a mobile handset. The data used included features like the entry of a four-digit number, series of varying phone numbers and entry of a particular phone number from 50 subjects. This yielded an overall error rate of 15%, the results were not great, but it did prove that each user could be distinguished from other users to an extent. In [4] similar research was conducted by Saevanee et al., using SMS texting activities and messages on mobile devices and achieved an overall error rate of 20%. These are all based on the use of keystroke biometrics which is less popular today as nearly all devices have adopted the touchscreen technology.

In more recent and relatable studies, Temper et al. in [5] proposed a method for continuous biometric authentication on Android devices using features like the

touchscreen gestures and related posture information to utilize a fuzzy classifier and scoring model. The researchers achieved an Equal Error Rate (ERR) of 11.5%. Another study in [6] conducted by De Luca et al. Based on touchscreen patterns at the point of entry into the device using the popular gesture-based authentication method used to unlock Android devices, they attempted to distinguish one user from the other. Similar research conducted in [7] by Sae-Bae achieved over 90% accuracy rate using a multi-touch or gesture approach, where 22 gestures were used to authenticate 34 users on an iPad.

However, the two most significant papers that relate to ours are [8] and [9]. The former, although based on keystroke biometrics, proposed a method to classify typing behaviors of different users using some artificial intelligence algorithms and performed a controlled brute force experiment on the trained model based on the observation of the users. The experiment achieved a 0.67 success rate. The latter utilized touchscreen input as a behavioral biometric for continuous authentication. The researchers used touch data that involved users performing up-down and left-right scrolling and developed an AI model that classified users based on their respective behaviors interacting with their mobile devices. The results show equal error rates between 0% and 4%, which hinged on the application scenario. The two papers reached the same conclusion that behavioral biometrics using keystroke and/or touch patterns might not be the best option for accurate authentication.

3 Attack Design

The main idea of this study is to investigate whether touch biometrics is viable enough to replace or augment the traditional authentication methods. This begins with developing AI models for about 75% of the users in the dataset. The model for each user is trained based on the behavioral characteristics provided in the dataset.

To ensure a broad and robust experiment we employed both supervised and unsupervised machine learning approaches. The supervised algorithms used were the Support Vector Machine (SVM), the K-Nearest Neighbors (KNN) and the Multi-Layer Perceptron (MLP) and the unsupervised learning algorithm used was the K-Means algorithm. We divided the data into training and attacking sets, using the attacking set to attack the trained AIs of each user. The aim to determine if any user in the attacking set can pass as even one user in the training set.

To further strengthen our design, we set a dynamic threshold for the number of similar patterns that each attacker can have with the legitimate users and found those users in the attacking set that can deceive the AI of the users in the training set.

4 Dataset

4.1 A Brief Information about Dataset

The dataset utilized in our project was obtained from a study carried out by Mario Frank in 2013. The data acquisition involved the use of Android phones where users assigned unique anonymous IDs were asked to read a particular text document and compare two images. The data was collected in a controlled environment that may or may

not have influenced the data and its applicability to real world events. For instance, the users were told the purpose of the experiment was to analyze their touch behaviors which may have made them more conscious of their typing behaviors. There is also the case of adaptability to the same tasks over time which does not leave room for evaluation of the user performing other tasks apart from the stipulated tasks.

4.2 Feature Extraction

The data was divided into individual strokes where each stroke is a pattern of the user touching the screen and lifting their finger. A single stroke is a trajectory that is encoded in a sequence of vectors, that consisted of the location, time stamp, pressure on the screen, the area covered by finger and the orientation of the mobile phone. Thirty-four features were generated from one stroke.

For our research, we cleaned the data by dropping some of the features that were not useful to our research, like the "phone id" and "doc id". We also dropped 310 patterns because they had null values as seen in the Figure 1 below.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21158 entries, 0 to 21157
Data columns (total 32 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   user id                                     21158 non-null  int64
 1   inter-stroke time                           21119 non-null  float64
 2   stroke duration                             21158 non-null  float64
 3   start $x$                                   21158 non-null  float64
 4   start $y$                                   21158 non-null  float64
 5   stop $x$                                    21158 non-null  float64
 6   stop $y$                                    21158 non-null  float64
 7   direct end-to-end distance                  21158 non-null  float64
 8   mean resultant lenght                       21158 non-null  float64
 9   up/down/left/right flag                     21158 non-null  int64
 10  direction of end-to-end line                21158 non-null  float64
 11  20\%-perc. pairwise velocity                21158 non-null  float64
 12  50\%-perc. pairwise velocity                21158 non-null  float64
 13  80\%-perc. pairwise velocity                21158 non-null  float64
 14  20\%-perc. pairwise acc                     20895 non-null  float64
 15  50\%-perc. pairwise acc                     20899 non-null  float64
 16  80\%-perc. pairwise acc                     20902 non-null  float64
 17  median velocity at last 3 pts               21157 non-null  float64
 18  largest deviation from end-to-end line      21158 non-null  float64
 19  20\%-perc. dev. from end-to-end line        21158 non-null  float64
 20  50\%-perc. dev. from end-to-end line        21158 non-null  float64
 21  80\%-perc. dev. from end-to-end line        21158 non-null  float64
 22  average direction                           21158 non-null  float64
 23  length of trajectory                        21158 non-null  float64
 24  ratio end-to-end dist and length of trajectory  20896 non-null  float64
 25  average velocity                            21158 non-null  float64
 26  median acceleration at first 5 points       20901 non-null  float64
 27  mid-stroke pressure                         21158 non-null  float64
 28  mid-stroke area covered                     21158 non-null  float64
 29  mid-stroke finger orientation               21158 non-null  float64
 30  change of finger orientation                21158 non-null  int64
 31  phone orientation                           21158 non-null  int64
dtypes: float64(28), int64(4)
memory usage: 5.2 MB
```

Figure 1. Features of the Dataset after Cleaning

5 AI Models

The choice of the machine learning models performs our experiment was a robust selection method that was based on the time it took for each model to train the model and the corresponding accuracies obtained as well as the compatibility with the research we were trying to perform. For instance, we tried the using the Support Vector Machine via the Radial Basis Function (RBF) kernel, but we obtained a 100% accuracy which meant that we could not use that kernel, as it had no weak points to exploit to deceive the AI.

We finally settled on 3 supervised learning algorithms, the Support Vector Machine (SVM) via the sigmoid kernel, the K-Nearest Neighbors (KNN) using the default parameters defined by the Sklearn package (i.e., K=5) and the Multi-Layer Perceptron (MLP) setting the maximum iteration to 300 and one unsupervised learning algorithm, the K-Means algorithm using the Elkan variation.

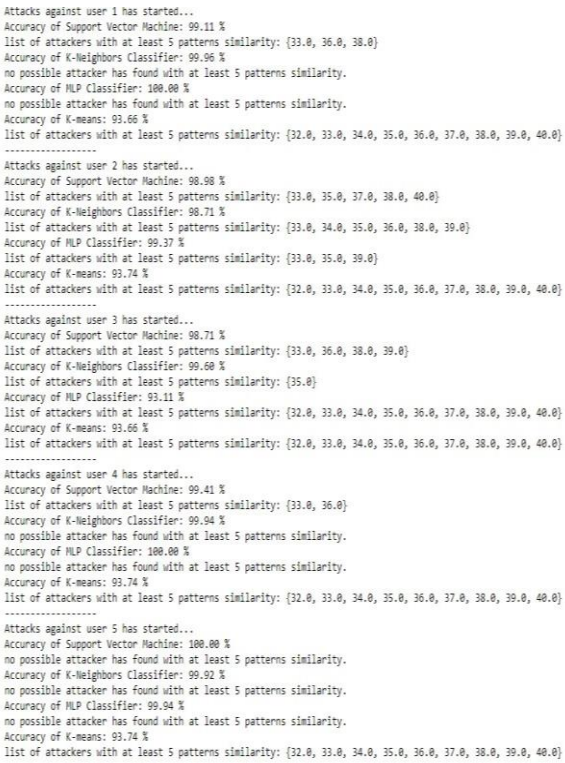

Figure 2. Experiment Results for the First Five Users

6 Experiment and Results

The training process began with splitting the data into training and attacking sets. We trained 31 AIs for the first 31 users in the dataset using all the four algorithms. We then attacked each trained AI using 10 different user patterns (i.e., from user number 32 to 41) from the dataset. We then set a dynamic threshold for the number of similar patterns that each user can have with each user. This will also show the attacker(s) that had similar patterns with a particular user from the training set. For our experiment, we set the threshold to 5 (i.e., 5 is the least number of similarities that an attacker can have with a user from the training set), this could be adjusted to any desired number.

The results as seen in the Figure 2 above, showed that the unsupervised learning algorithm (K-Means clustering classifier) was the weakest algorithm with results barely above 93% on each AI. While the other 3 algorithms (SVM, KNN and MLP classifiers), yielded more competitive results ranging from 98% to 100%. Regardless, these algorithms still proved vulnerable to the attacks from the attacking sets with SVM coming in as the second weakest algorithm. Where K-Means recorded almost 9 attackers out of the 10 defined attackers having similar patterns as a particular user from the training set (legitimate user), SVM had at least 2 attacking users presenting similar patterns as the real user at each experiment stage. KNN and MLP proved to be stronger than K-Means and SVM algorithms with varying number of attacking users having at least 5 similarities with the real users from the training sets. However, for a good number of iterations and users, KNN and MLP presented no 0 or 1 attacker that had similarities with the real users. The Table 1. below depicts the failure rates of each algorithm

Training Model/Algorithm	Failure Rate (%)
KNN	35.48
MLP	70.96
SVM	90.32
K-MEANS	100

Table 1. The Training Model/Algorithm and the Corresponding Failure Rates.

7 Discussion and Limitations

The result from our experiment suggests that it is indeed possible to distinguish one user from another based on the way each user interacts with their smartphones. Depending

on the algorithm selected to classify the behaviors of users, the strength of this argument is variable. However, from our results, we can ascertain that the K-Means algorithm is out of the question. As stated earlier, using the SVM algorithm via the RBF kernel yielded 100% accuracy which was quite surprising, could it mean that using SVM via the RBF kernel is the way to go to yield an impenetrable touch behavioral biometrics system? That is still yet to be decided.

The focus of our work was to perform a brute force attack on the touch behavioral biometrics. We have succeeded in proving that touch input behavioral biometrics is not secure enough to replace the current authentication methods. We encountered some limitations in proving this fact to be true, especially with the dataset. For instance, the fact that our dataset only comprised of 41 subjects which is not enough to carry out robust research, the controlled circumstances in which the data was obtained could also be called into question as the users were only allowed to perform a limited number of tasks on the phone that mostly involved up-down and left-right scrolling motions. This does not give the full range of behaviors that could be exhibited by a user when interacting with a mobile phone.

8 Conclusion and Future Work

We performed a successful investigation into the viability of touch inputs as a behavioral biometric for authentication on mobile phones using machine learning methods. To accomplish this, we simulated an environment that consisted of 41 users, 31 real users and 10 threat actors. Training 31 AIs for the first 31 users and using the left over 10 users to attempt to deceive the AIs into believing that they were the real users. The results showed that the AIs could be tricked, proving that touch input biometric is not secure enough as a stand-alone authentication method or even as a method for continuous authentication based on the overwhelming number similarities two or more users could have relative to the training algorithm employed.

In subsequent research, more machine learning methods could be explored to ascertain if there exists an algorithm that would prove to be unbreakable. The dataset could also be more diversified to yield more accurate and relatable real-world results. Different authentication methods could be explored like the eye movements as a user interacts with their screen or research could just remain on already provided research methods and improving them. These methods comprise of using physiological features, some form of password, pin, etc.

9 Contribution

The contribution of each member of the research are outlined below.

Faith Obasi

1. Training of the three supervised learning models,
2. Report writing, and
3. PowerPoint presentation preparation

Danyal Namakshenas

1. Preparation of the dataset,
2. Feature extraction,
3. Training of unsupervised learning model and general linkage of the code,
4. Readme File preparation and report writing contribution

The final submission of the paper and code was reviewed and edited by both parties of the research.

References

[1] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in Proc. 4th USENIX Conf. Offensive technologies (WOOT'10), Berkeley, CA, 2010, pp. 1–7 [Online]. Available: https://dl.acm.org/citation.cfm?id=1925004.1925009,%20USENIX%20Association

[2] A. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *IEEE Trans. Inf. Forensics Security, vol. 1, no. 2, pp. 125–143, Jun. 2006.*

[3] Clarke, N. L., Fumell, S. M., & Reynolds, P. L. (2002). Biometric Authentication for Mobile Devices. *3rd Australian Information Warfare and Security Conference*. https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=7759&context=ecuworks&httpsredir=1&referer=#page=69

[4] Saevanee, H., Clarke, N. L., & Furnell, S. M. (2012). *Multi-modal Behavioural Biometric Authentication for Mobile Devices*. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-642-30436-1_38?error=cookies_not_supported&code=bce682cc-ca6f-4f9b-aa68-8fb2f0ef3b26

[5] Touch to Authenticate — Continuous Biometric Authentication on Mobile Devices. (2015, July 1). *IEEE Conference Publication | IEEE Xplore.* https://ieeexplore.ieee.org/document/7812943/;jsessionid=mNvaayV7VMaFshov2yiCTDahh-e9WjIWDiSmIpTZG02I20_KTYmC!-1256091585?arnumber=7812943

[6] De Luca, A., Hang, A., Brudy, F., Lindner, C., & Hussmann, H. (2012). Touch me once and i know it's you! *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2207676.2208544

[7] Sae-Bae, N., Ahmed, K., Isbister, K., & Memon, N. (2012). Biometric-rich gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2207676.2208543

[8] Zhu G. and Ismail T. (2021). AI-Assisted Brute-Force Attacks on Keystroke Biometric Authentication Systems. CIS*6510 Final Project Writeup.

[9] Frank, M. (2013, July 30). Touchalytics. Retrieved December 2, 2022, from http://www.mariofrank.net/touchalytics/