Effective Touch Dynamics Features

Carlos Leon, Karshan Arjun and Subhrajyoti Pradhan

Abstract- The purpose of this research is to determine which touch dynamics features would be more optimal to use from the list of all possible features in touch dynamics. By finding the most efficient features to use, researchers can then use this information to create a better model from this biometric. The goal of this paper is to examine the procedure that was taken to retrieve the optimal and efficient features from the list of features in touch dynamics. Given the list of optimal features, the glimpse of using these features will allow the study of mobile biometrics to consider using the touch dynamics biometric modality more.

I. Introduction

As number of sensors in a mobile device increase and the performance of these sensors in mobile devices improve, the more optimal the data gathered from these mobiles will be. The need of finding optimal data from these sensors are needed to improve the uniqueness of each mobile device from another one. This will allow the mobile device to be more secure and this security feature will help remove the threat of gathering data from a mobile device.

Of the many biometrics modalities that have been gathered from mobile devices, touch dynamics is an interesting modality that is used constantly by users of mobile devices. The data that is captured by this biometric is however very costly for algorithms to process, has a high energy consumption for the battery of the device, this leads to low accuracy rates in finding imposter and genuine scores and also has a factor of user adaptation to their touch dynamics which must be factored in when using this biometric.

With these current challenges, the data that is collected from touch dynamics has a large amount features which have could have been a reason for all the problems that is faced with this biometric. Perhaps, finding the best of all the features that is collected would help alleviate the challenges and allow for this biometric to be more optimal for future use.

In this paper, the overview of touch dynamics and the challenges of this biometric is discussed in Section 2. In Sections 3 and 4, the features of touch dynamics will be examined as well as the process of how to consider effective features. Lastly, Section 4 and 5 will discuss the results of the implementation and the datasets which were used in the process and the conclusion of the research.

II. Touch Dynamics

Of the many biometric modalities, touch dynamics, which is the study of measuring and assessing user interaction with a touch screen device, is considered promising in the field of biometric authentication as the main interaction of the touch screen is constantly used as the user interacts with the device. This biometric allows a user to not require any physically body parts such as the iris or face to gather information needed for the results of the biometric.

In the early conception of this biometric, was the use of keystrokes from a computer keyboard to gather information of the user habits when typing. The birth of keystroke dynamics was considered, after the usage of its information was proposed to be useful in the field of authentication.

As modern devices have used less keyboard input for user interaction, the rise of touch screens have made progress to take over the mobile device landscape. The amount of data this is gathered from the touch screens can as well be used for biometric authentication as the same features of keystroke dynamics are present in the touch screen. Even more features were captured when a user interacts, however the amount of features have led to a surplus of data that may or may not be effective to gather.

The challenges of touch dynamics are:

- Reducing energy consumption - Mobile phones, unlike computers, run on batteries and hence power supply is limited. Therefore, the less the application takes memories, the more the device can be operated. So it is essential to have a system that consumes minimum battery so that it does not majorly impact the phone usage time. Some measures, such as reducing the sampling rate (Niu and Chen, 2012) or performing complex computation only when a device is being recharged (Crawford et al., 2013)
- Reducing computational cost- Computational cost in mobile phones are generally inferior than computer devices. Hence criterions such as criteria such as algorithm complexity, communication cost, and authentication delay are important and should be considered in the design of touch dynamics authentication solutions. In other words, algorithm and communication costs introduced as the result of deploying this authentication means should be minimal.(Teh and Zhang, 2016)
- Adaptation Capability: Human behavioral characteristics are susceptible to change over time, and more frequently than physiological characteristics. A user's touch dynamics patterns can gradually change as the user gets more familiar with the passcode, input method, device, and other external factors. A touch dynamics authentication system should be capable of adapting itself to any changes in a user's touch dynamics pattern.(Teh and Zhang, 2016)

Throughout this paper we try to create a model which addresses these issues.

III. Features of Touch Dynamics

We used two dataset in our experiments. The first was created by Dr. Neil Tempest on a Nexus 7. The dataset contained 71 features which were all numerical along with 2856 records. The measured attributes are:

- Hold (H)
- Up-Down (UD)
- Down-Down (DD)
- Pressure (P)
- Finger-Area (A)
- Averages of Hold (AH)
- Pressure (AP)
- Finger Area (AFA)

The second dataset used was the MOBIKEY Keystroke Dynamics Password Database. We limited ourselves to the evoline1 subsection of it. The dataset had 14 features and ~15,000 records. One of the columns was the UserID i.e. the label we intend to predict using KNN in this case.

IV. Implementation of the process

Our implementation was divided into feature selection and the decision making. We employed the use of a Python library, scikit-learn, to perform the following types of feature selection algorithms:

- *Chi square*, this statistical algorithm that measures the frequency of expected and observed data, was used to filter the amount of features of touch dynamics. This implementation was done by getting the p-value of the features which value was less than a given alpha value.
- *Information gain*, using entropy in this algorithm we were able to also filter out the best features which had a high information gain.
- *Recursive*, for this algorithm, our goal was to select features by recursively considering smaller and smaller sets of features. The estimator was defined as a support vector machine Then, the least important features were removed from current set of features.That procedure was recursively repeated on the removed set until the desired number of features to select is eventually reached.
- *Tree selection*, in this algorithm, we used tree based estimators. It employs the process of tree selection to determine feature importance. It then uses the average of the importance to remove unneeded features
- *Variance threshold*, in this algorithm, we remove all features whose variance doesn't meet the declared threshold. By default, the code removed all zero-variance features, i.e. features that have the same value in all samples. We used the formula for variance V= p(1-p), to achieve this.

The script breaks up the dataset using 5 fold cross validation. The chunk meant to be used for testing was left alone, and the rest had outliers removed. Outlier detection was done using isolation forest in combination with leave-one-out-validation. Thus every row was tested as being an outlier against every other row. In practice we found that only a small percentage of rows were dropped; about 5%. The remaining training rows post outlier detection were used for the KNN training. Also, the training and query chunks are scaled using a MinMaxScaler after outlier detection. We chose k=31 since that yielded passable results during debugging with a partial database, but later tested other values for k.

Lastly, the main script was adapted into another script called a validation script. The purpose of this was to test our method on another database - the second database. This would serve to confirm or reject out method. The script initially used all the same code with only minor changes to account for different location of the dataset and type of file to read. The script, however, did not work. More specifically the method used for the first dataset did not work when applied to the second dataset. Essentially this inherently rejects our method, however, we went further in our attempts to make it work. First we noticed that there were simply too much data, ~15,000 rows vs ~2,500 for the first dataset. Out script never finished even when given several hours. To remedy this we switched to taking a random sample of 2,000 rows. Secondly, we noticed the data had text data and our script was meant to work with numerical. To remedy this we applied a label encoder. Thirdly, we noticed that the recursive feature elimination algorithm took far too long. To remedy this we changed the SVM estimator for a Logistic Estimator that would perform much faster. However, the estimator would not converge thus the dataset can not be fit to a logistic curve thus a different estimator was needed. Then, an SGD Estimator was used. It worked considerably faster and did converge. Next we hit a problem we could not solve: the feature selection algorithms did not agree on a set of features to keep after the intersection of them were taken. For the machine learning algorithm to work two or more feature columns were needed. We saw that after 5 attempts to get selected features at best 1 feature was chosen by all 5 algorithms.

V. Results

The first set of results are the ones required when using only a small portion of the first dataset. We used ~350 rows of the first dataset when building our script. When the script was done we observed that with k for KNN being 31 for 12 runs we averaged 65.7% accuracy. The number of features chosen were between 3 - 6, collectively they were

- Hold i
- Hold Shift
- Hold Caps
- Hold a
- Hold n
- Hold l

The next set of results come from testing on the entire dataset. Below are the ROC, DET, and Score distributions along with the D-prime value and the EER value. The accuracy for these results, were very low, strangely low, for integrity they were 5.6% to 7%. With the 7% being reached with k=75 and the lowest coming from k=81.

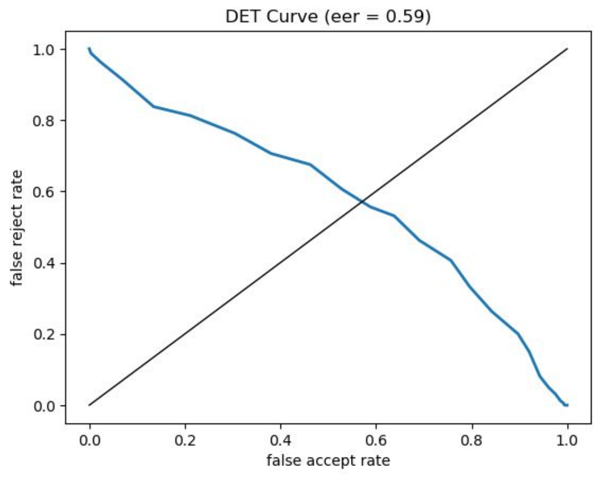

Figure 1. The DET curve with the EER value.

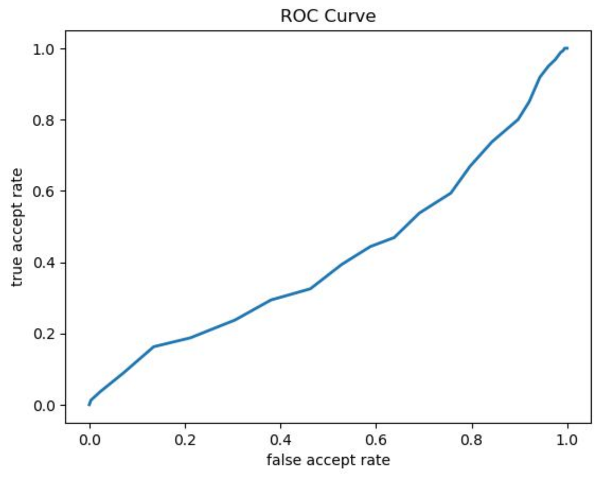

Figure 2. The ROC curve with the D-Prime value

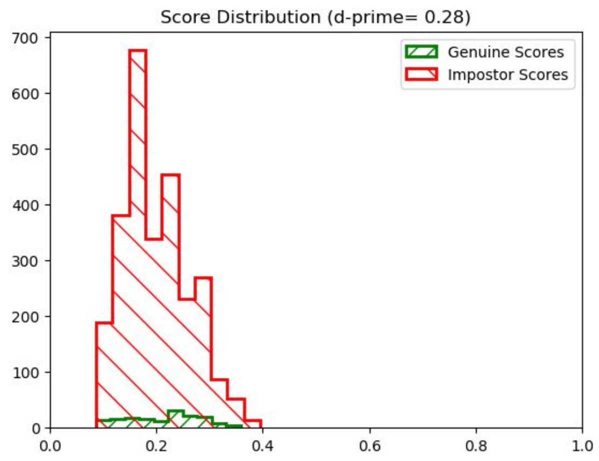

Figure 3. The Score distribution between genuine and imposter scores

There is one last set of results to make note of. How the script performs when using the entirety of the dataset and every feature. As mentioned before, with k=81 we got 5.6% accuracy for KNN, but using all the features with k=81 we got 1.9% to 2%. That shows a significant drop relative to the performance with feature selection. This points to KNN itself being the likely culprit for failure.

VI. Conclusion

Our approach to feature selection yielded fairly high results when used on small samples. Even though our results were, far from ideal when it came to testing on the entire dataset, we determined the major problem existed in the speed of performing all the necessary calculation. Our approach of using k Nearest Neighbor, seemed to be the issue, as our k value was not ideal for our dataset. We were unable to determine a good k value as the runtime of our entire code exceeded thirty minutes, and hence multiple tests on sample datasets were not feasible. We determined some sort of tree approach like the random forest would have been better for future works. This would ensure us to achieve our goal of achieving a system, which would seamlessly implement the touch gesture as a feasible mode of biometric authentication.

VII. Sources

- Teh, Pin Shen, et al. "A survey on touch dynamics authentication in mobile devices." *Computers & Security* 59 (2016): 210-235..
- Niu, Yuan, and Hao Chen. "Gesture authentication with touch input for mobile devices." *International Conference on Security and Privacy in Mobile Information and Communication Systems*. Springer, Berlin, Heidelberg, 2011.
- Jain, Anil K., Arun A. Ross, and Karthik Nandakumar. *Introduction to biometrics*. Springer Science & Business Media, 2011.