



**Manchester  
Metropolitan  
University**

---

Teh, PS and Zhang, N and Tan, SY and Shi, Q and Nawaz, R and Khoh, WH (2019) Sensing Your Touch: Strengthen User Authentication via Touch Dynamic Biometrics. In: ICTC 2019 - 10th International Conference on ICT Convergence: ICT Convergence Leading the Autonomous Future, 16 October 2019 - 18 October 2019, Jeju Island, South Korea.

---

**Downloaded from:** <http://e-space.mmu.ac.uk/626224/>

**Publisher:** IEEE

**DOI:** <https://doi.org/10.1109/ICTC46691.2019.8939685>

Please cite the published version

<https://e-space.mmu.ac.uk>

# Sensing Your Touch: Strengthen User Authentication via Touch Dynamic Biometrics

Pin Shen Teh

Department of Operations, Technology,  
Events and Hospitality Management  
Manchester Metropolitan University  
Manchester, UK  
p.teh@mmu.ac.uk

Ning Zhang

School of Computer Science  
University of Manchester  
Manchester, UK  
ning.zhang-2@manchester.ac.uk

Syh-Yuan Tan

School of Computing  
Newcastle University  
Newcastle upon Tyne, UK  
syh-yuan.tan@newcastle.ac.uk

Qi Shi

Department of Computer Science  
Liverpool John Moores University  
Liverpool, UK  
q.shi@ljmu.ac.uk

Raheel Nawaz

Department of Operations, Technology,  
Events and Hospitality Management  
Manchester Metropolitan University  
Manchester, UK  
r.nawaz@mmu.ac.uk

Wee How Khoh

Faculty of Information Science and  
Technology  
Multimedia University  
Malacca, Malaysia  
whkhoh@mmu.edu.my

**Abstract**— Mobile devices are increasingly used to store private and sensitive data, and this has led to an increased demand for more secure and usable authentication services. Currently, mobile device authentication services mainly use a knowledge-based method, e.g. a PIN-based authentication method, and, in some cases, a fingerprint-based authentication method is also supported. The knowledge-based method is vulnerable to impersonation attacks, while the fingerprint-based method can be unreliable sometimes. To make the authentication service more secure and reliable for mobile device users, this paper describes our efforts in investigating the benefits of integrating a touch dynamics authentication method into a PIN-based authentication method. It describes the design, implementation and evaluation of this method. Experimental results show that this approach can significantly reduce the success rate of impersonation attempts; in the case of a 4-digit PIN, the success rate is reduced from 100% (if only the PIN is used) to 9.9% (if both the PIN and the touch dynamics are used).

**Keywords**—Authentication, biometrics, touch dynamics, machine learning, mobile device security

## I. INTRODUCTION

Mobile devices have become a preferred gadget for users to access information and digital services, and stay connected. The increased usage and dependence on these devices also indicate that they increasingly process and store confidential and sensitive data. As more sensitive data are stored in, or accessible from, mobile devices, the risk and cost of losing these data are becoming higher. Therefore, how to make the authentication service secure and reliable for mobile users is a pressing task. One of the possible measures to strengthen the security and reliability of the authentication service is to integrate a biometrics-based (e.g. touch dynamics) authentication method with a knowledge-based (e.g. PIN) authentication method to form a so-called two-factor authentication method [1].

Touch dynamics refer to the digital signatures generated when a human interacts with a mobile device. A touch dynamics authentication method can be implemented by employing sensors already available in most mobile phones, digital tablets, and other touchscreen devices, making the implementation comparatively cheaper than other biometrics-based authentication methods, such as fingerprint and iris where specialised hardware is required. In addition, the

acquisition of touch dynamics features is less sensitive to external factors such as lighting conditions and background noise levels, making it more usable and reliable in a mobile context. Also, touch dynamics features can be acquired whenever a user uses his/her devices, for example, during their normal (i.e. non-authentication related) input activities, requiring little extra interactions by the user. For these reasons, a touch dynamics authentication method is cheaper, more usable and reliable, and may be more acceptable to the general public than other biometrics-based authentication methods.

To investigate the feasibility and effectiveness of using touch dynamics biometrics as a mobile device authentication solution, we have designed and evaluated a touch dynamics authentication method. This paper reports our work in this regard. More specifically, it describes what types of features and how they are extracted. It then describes the classification of the features to build authentication model, and the use of the model to authenticate a user. The paper also describes the experiments carried out to evaluate the performance of the touch dynamics authentication method and discusses the evaluation results obtained with different parameter value settings.

## II. RELATED WORK

This section critically analyses related work on touch dynamics. As the scope of our work is in touch dynamics using numerical-based input strings, our literature critical analysis here will focus on this input string type. For details on related work in other groups, readers are referred to a recent literature survey of touch dynamics [2]. The work most relevant to ours has largely been focusing on studying the applicability of, or improving the performance in, using touch dynamics as a means of verifying subjects. Here, we discuss the most notable ones.

The work reported in [3] was carried out to test the applicability of verifying subjects based on touch dynamics using numerical-based input strings. In their experiments, some of the touch dynamics features were extracted by using the more sophisticated accelerometer and gyroscope sensors. By using an Euclidean Distance classifier, they obtained an accuracy performance of 20% equal error rate (EER) on a 4-digit PIN. Using more sophisticated sensors to extract some of the features means that this method is energy-consuming [4].

The authors in [5] also investigated the accuracy performance of touch dynamics using a 4-digit PIN. In their investigation, subjects were asked to provide 100 input samples of a predefined PIN (“1593”) each. The authors used the Multi-Layer Perceptron classifier to classify the legitimate subjects, and these subjects can be correctly classified up to 86% of the time. The result is encouraging, but to achieve the reported level of accuracy performance, they have made use of 100 input samples per subject to train the classifier. Acquiring such a large number of samples from the subjects is time-consuming and not always practical during an enrolment phase.

The work reported in [6] was somewhat unique. The authors proposed a method to allow subjects to change their PINs without rebuilding the authentication model. The subjects were asked to input ten different randomly selected 10-digit PINs. Based on the samples collected, they produced a table of all possible feature values for each digit. Using this method, they were able to achieve EER values of 23%, 21%, and 18% on three different PINs with the string lengths of 6, 8, and 10, respectively. However, the majority of the subjects taking part in this experiment were at the age of 17-20, it is not clear whether the experimental findings apply to other age groups.

More recently, Shen et al. [7] investigated the accuracy performance of touch dynamics on 4-digit, 5-digit, and 6-digit PINs. Unlike other related work, they used only raw motion data extracted from accelerometer and gyroscope sensors. To make the raw data useable as features, they computed a set of statistical metrics (min, max, mean, variance, etc.) from the raw data, and used the computed metrics as motion features. A similar method has also been used in other experiments such as [8]–[10]. By far, this method has only been used to extract motion features. It would be interesting to see how well this method works when used to extract other types of features, such as timing and spatial features investigated in our work.

By far, the best accuracy performance was reported by [11]. The authors achieved an EER value of 0.56%. In this work, a two-class classification approach is used in building the authentication model, i.e. to build the model, samples from both legitimate and illegitimate subjects are used. This is also the case for the work reported in papers [8], [11], [12]. However, in real-life, as mobile devices are very much personal devices, illegitimate subject samples may not always be available. Therefore, this approach is less practical.

### III. AUTHENTICATION SYSTEMS DESIGN

#### A. The Architecture and its Functional Units

This section presents an overview of the system architecture for our touch dynamics authentication method. As shown in Fig. 1, the architecture consists of six functional units all run on a user’s mobile device. AIU is an input facility for users to provide their touch dynamics input samples. DSU is a database used to store authentication model. The rest four units provide the core functions of our touch dynamics authentication system and are summarised below.

- RDAU: This unit acquires touch dynamics input samples from users via AIU and extracts raw touch dynamics data from the samples. These raw data are passed to FCU for feature construction.

- FCU: This unit identifies and extracts touch dynamics features from the raw data. The extracted features are passed to MTU for model training.
- MTU: This unit analyses and trains the extracted features to build an authentication model. The built model is stored in DSU and will be used by ADMU for authentication decision-making.
- ADMU: This unit makes an authentication decision by matching a claimant’s touch dynamics features against the model stored in DSU.

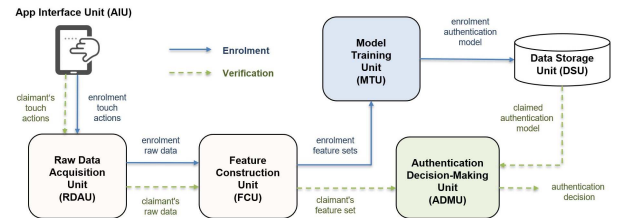


Fig. 1. The touch dynamics authentication system architecture.

The operation of a touch dynamics authentication system can broadly be captured in two phases: enrolment and verification. In the enrolment phase, the touch dynamics input samples of the owner of a mobile device are acquired, processed and transformed into an authentication model that is stored in the DSU. In the verification phase, the touch dynamics input samples of a test subject (i.e. a claimant) is acquired, processed and compared against the model retrieved from DSU to verify if the claimant is indeed whom he/she claims to be (i.e. the owner of the mobile device). The two operational phases along with the units involved are illustrated in Fig. 1. In the next section, we describe the designs of the four core units and discuss the issues involved in more detail.

#### B. Raw Data Acquisition Unit (RDAU)

RDAU is the first core functional unit of the proposed system architecture, responsible for extracting raw touch dynamics data from the subject’s input samples. The design details of this unit and the dataset used for this experiment can be found in [6]. The paper gives detailed discussions with regard to how the raw data acquisition experiment is setup, how the input samples are acquired, how raw touch dynamics data are extracted from the input samples, and how the raw data are processed into a proper format for further analysis. The entire dataset used consists of 3000 samples and 33000 touch actions from 150 subjects. Each subject contributed a total of 20 samples (10 for the 4D string and 10 for the 16D string) from 220 touch actions (50 for the 4D string and 170 for the 16D string). The dataset is available to download at <https://goo.gl/sNACU8>.

#### C. Feature Construction Unit (FCU)

FCU is responsible for extracting a subject’s touch dynamics features from the subject’s raw touch dynamics data. In our design, two categories of features are extracted, first-order features (FOF) and second-order features (SOF). FOF features are a basic set of features extracted directly from the raw touch dynamics data, and SOF features are an extended set of features extracted from FOF features.

##### 1) First-Order Features (FOF)

This section describes the process of extracting FOF features from a subject’s raw touch dynamics data and constructing a cumulative FOF feature vector for the subject.

For each subject, a number of FOF features are captured, one spatial feature and multiple timing features.

The pressure size (PS) is a spatial feature capturing the approximated size of the screen area being touched. A timing feature is an attribute capturing a time interval between two touch actions of one or more keys. Depending on how the intervals are measured, there are three types of timing features, i.e. dwell time (DT), flight time (FT), and input time (IT), and for FT, there are further four variants, i.e. FT1, FT2, FT3, and FT4. The descriptions of these timing features are given in [6].

Once FOF features,  $f_i, i \in \{1, 2, \dots, d\}$ , are extracted from the raw data, they should be organised into the form of an FOF feature vector, i.e.  $v_{FOF}^T = [f_1, f_2, \dots, f_d]$ , where  $T$  indicates a particular type of FOF feature and  $d$  refers to the feature dimension of  $v_{FOF}^T$ . When all the feature vectors are formed for a subject, a cumulative FOF feature vector,  $V_{FOF}$ , can be generated. This is done by concatenating the FOF feature vectors, i.e.  $V_{FOF} = [v_{FOF}^{DT}, v_{FOF}^{FT1}, \dots, v_{FOF}^{PS}]$ .

## 2) Second-Order Features (SOF)

Some classification algorithms (or classifiers) perform better with a larger number of features [13], so increasing the number of features used in training the classifier to generate an authentication model can improve the accuracy performance of the model. For this reason, we extract a new category of features, known as the second-order features, from FOF features, and use both of them in the training of authentication model. As discussed above, FOF features extracted from the raw touch dynamics data are organised into FOF feature vectors. For each of these vectors, a set of SOF features is extracted. The set consists of 19 features, and each feature represents a descriptive statistics metric of the FOF feature vector concerned. Descriptive statistics metrics are used to quantitatively summarise or describe a collection of data in a meaningful way [14]. The list of descriptive statistics metrics used in our experiment (with their corresponding feature identifiers in brackets) are: Minimum (mn), Maximum (mx), Arithmetic Mean (am), Quadratic Mean (qm), Harmonic Mean (hm), Geometric Mean (gm), Median (md), Range (rg), Variance (vr), Standard Deviation (sd), Skewness (sk), Kurtosis (ku), First Quartile (fq), Third Quartile (tq), Interquartile Range (ir), Mean Absolute Deviation (ma), Median Absolute Deviation (mi), Coefficient of Variation (cv), and Standard Error of Mean (se).

Similar to the case for FOF features, SOF features,  $f_i, i \in \{mn, mx, \dots, se\}$ , once extracted from FOF features, should be organised into the form of a SOF feature vector, i.e.  $v_{SOF}^T = [f_{mn}, f_{mx}, \dots, f_{se}]$ , where  $T$  indicates a particular type of FOF feature. When all the SOF feature vectors are formed for a subject, a cumulative SOF feature vector,  $V_{SOF}$ , can be generated. This is done by concatenating the SOF feature vectors, i.e.  $V_{SOF} = [v_{SOF}^{DT}, v_{SOF}^{FT1}, \dots, v_{SOF}^{PS}]$ . Once both FOF and SOF cumulative feature vectors are formed for a subject, they should be combined into a form of a feature set. A feature set consists of the FOF and SOF features extracted from the raw touch dynamics data of a single input string of a subject.

## D. Model Training Unit (MTU)

MTU analyses the touch dynamics feature sets (also referred to as touch dynamics samples or samples) extracted by FCU and trains them to generate an authentication model. The generated model should uniquely represent the corresponding subject's touch dynamics pattern.

Depending on the data used, classifiers can be classified into two groups, one-class classifier (OCC) and two-class classifier (TCC). An OCC classifier only uses data from a single class (e.g. data from legitimate subject). Unlike an OCC classifier, a TCC classifier uses data from two classes (e.g. data from both legitimate and illegitimate subjects). In the mobile device context, obtaining two classes of data with a similar size is not practical. This is due to the fact that a mobile device is rarely shared among multiple users. Also, sharing a passcode with others increases data privacy risks and is not a recommended practice. For these reasons, obtaining illegitimate subject data is not practical, and only the data from the legitimate subject are available for use to train the classifier. If this is the case, a TCC classifier may not perform well, as it requires data from two classes to train a model that separates the two classes apart [15]. By contrast, an OCC classifier only needs data from one class to train a model, so in this case, the model training is not affected by any imbalanced data. Besides, the time taken by a TCC classifier to train a model is longer than that by an OCC classifier, as the former uses more data in training the model. Based on these considerations, we have chosen to use OCC classifier for feature classifications.

In this paper, we have implemented both OCC and TCC classifiers. Specifically, we have implemented two OCC classifiers: (1) one-class k-nearest neighbour (OCKNN) [16], and (2) the support vector data description (SVDD) [17]. For our comparative study, i.e. the study we have carried out to examine the effectiveness of using OCC classifiers versus using TCC classifiers, we have implemented two TCC classifiers as well: (1) k-nearest neighbour (KNN), and (2) support vector machine (SVM) [18]. The classifiers used in our experiments are implemented using the Matlab (version 8.5.0.197613) programming platform and two open source toolboxes. The OCC and TCC classifiers are implemented using the `dd_tools` toolbox [19] and the `PRTTools` toolbox [20], respectively.

## E. Authentication Decision-Making Unit (ADMU)

ADMU makes an authentication decision, i.e. whether a testing sample matches with the authentication model of the owner of the device. The design of ADMU involves two processes, feature matching and feature thresholding. In the feature matching process, the testing sample acquired from an authentication attempt is matched against the stored model in DSU to obtain a classification score. In the thresholding process, the score is compared to a predefined threshold, and if the score is over the threshold, then the sample is classified as legitimate. Otherwise, the sample is classified as illegitimate.

## IV. PERFORMANCE ANALYSIS

### A. Evaluation Method

To perform the accuracy performance evaluation of our touch dynamics authentication methods, we should perform the following four tasks. Firstly, we classify subjects into two sets, one designated as legitimate subjects and the other as illegitimate subjects. Secondly, some of the touch dynamics samples acquired from these subjects are used as training samples, in which these samples are used by MTU to generate authentication models. Thirdly, some of the other samples are used as testing samples, in which these samples and the generated models are used by ADMU to make authentication decisions. Lastly, based on the decisions, the evaluation metrics values are calculated, which indicates the accuracy

performance of the model. The performance evaluation method described above is implemented using the evaluation procedure summarised in Algorithm 1.

---

**Algorithm 1. Evaluation procedure**

---

**Input:** Dataset  $S$  with  $B$  number of subjects,  $\{d_1, d_2, \dots, d_B\}$ , Classifier  $C$ , folds  $K$

**Output:** Accuracy performance of the model  $P$

---

```

for  $b = 1$  to  $B$  do
   $S^+ \leftarrow$  initialise the legitimate subject samples  $\{d_b\}$ 
   $S^- \leftarrow$  initialise the illegitimate subjects samples  $S - \{d_b\}$ 
  Randomly split  $S^+$  and  $S^-$  into  $K$  disjoint folds,  $\{s_1^+, s_2^+, \dots, s_K^+\}$  and  $\{s_1^-, s_2^-, \dots, s_K^-\}$ 
  for  $k = 1$  to  $K$  do
     $S_{tr}^+ \leftarrow$  initialise the legitimate training samples,  $S^+ - \{s_k^+\}$ 
     $S_{tr}^- \leftarrow$  initialise the illegitimate training samples,  $S^- - \{s_k^-\}$ 
     $T_{tr} \leftarrow$  initialise the training set,  $S_{tr}^+ + S_{tr}^-$ 
     $T_{ts} \leftarrow$  initialise the testing set,  $\{s_k^+\} + \{s_k^-\}$ 
    if  $C$  is a OCC classifier then
       $T_{tr} \leftarrow T_{tr} - S_{tr}^-$ 
       $T_{ts} \leftarrow T_{ts} + S_{tr}^-$ 
    end if
    Train  $C$  on  $T_{tr}$  to build model  $M_b$ 
     $P_{fd} \leftarrow P_{fd} +$  (test the accuracy performance of  $M_b$  on  $T_{ts}$ )
  end for
   $P_{sb} \leftarrow P_{sb} + P_{fd}/K$ 
end for
 $P \leftarrow P_{sb}/B$ 

```

---

### B. Evaluation Metrics

To evaluate the accuracy performance of the authentication model, three evaluation metrics are used, the False Rejection Rate (FRR), the False Acceptance Rate (FAR) and the Equal Error Rate (EER). The lower the values of these metrics the better the accuracy performance of the model. FRR and FAR are also used to plot the Detection Error Trade-off (DET) curve [21], which is used to evaluate and compare the accuracy performances of different models in a graphical representation form. To plot the DET curve of a model, a set of FRR and FAR values of the model is needed. The values are obtained by setting the threshold to different values. The curve is formed by plotting the FRR values on the y-axis and the FAR values on the x-axis. The closer the curve to the bottom left corner, the better the accuracy performance of the model.

### C. Results Discussion and Analysis

This section describes the experiments carried out to evaluate the performance of the touch dynamics authentication method and discusses the evaluation results obtained. The experiments were conducted using the evaluation methodology described in Section IV.A. Unless otherwise stated, each experiment was repeated four times, each time using one of the four classifiers (discussed in Section III.D) in turn, and the results reported were the average of the classifiers.

#### 1) Input String Lengths

Using different input string lengths may also affect EER values. To examine the effect, we used two input strings with two different lengths, a 4D and a 16D string. For both input strings, FOF and SOF features were extracted. For each input string, we repeated the experiment four times, and for each time one of the four classifiers were used.

Fig. 2 shows the EER values versus two input strings and four different classifiers. As shown in the figure, for all the classifiers, using the 16D string introduces a lower EER value, indicating that the longer the input string, the more accurate the authentication model. The reasons for this are three-fold. Firstly, the length of the 16D string is four times longer than that of the 4D string, and so is the number of features that are extracted from the 16D string. More features means more information about a subject's touch dynamics pattern can be captured, therefore a more accurate model can be built out of the features. Secondly, when the input string length increases, the number of possible chunk combinations also increases, and so is the ability to better capture a subject's touch dynamics pattern. Finally, when the input string length increases, the number of illegitimate features required to match that of a legitimate model will also increase, which means that the level of difficulty in impersonating a subject successfully also increases.

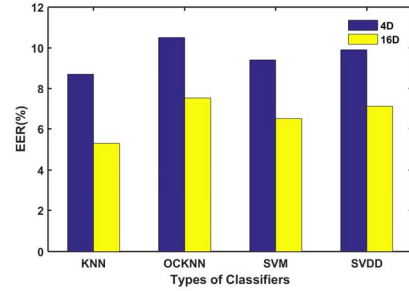


Fig. 2. EER values versus two different input string lengths and four classifiers.

The above results have revealed a correlation between the input string length and security. The shorter the input string length, the lower the level of authentication accuracy, indicating a lower level of security. There is also a correlation between the input string length and usability. The longer the input string, the more the number of touch actions are required to complete the input of the string, thus the harder and slower it is for the users to memorise the string, indicating a lower level of usability. A similar correlation has also been reported in [22]. In summary, the input string length influences the trade-off between security and usability. Therefore, in real-life applications, it should be chosen based on the security and usability requirements of the apps.

#### 2) FOF features

There are four types of FOF (as discussed in Section III.C.1)). Each type of FOF captures a subject's touch dynamics pattern in a different way. To investigate the accuracy performance of the authentication method using different types of FOF, we have extracted all four types of FOF from the 4D string as the test case.

Fig. 3 shows the EER values of the four types of FOF. The EER value of PS is the lowest amongst the four types, which means that the accuracy performance of PS is better than timing features. This result can be explained as follows. The PS values are determined by several factors such as: (1) the physical size of the fingertip used to perform a TAP, (2) the amount of force exerted during a TAP, and (3) the fingertip position or angle during a TAP. The combination of these factors creates a distinctive pattern, which allows PS to better capture each subject's touch dynamics pattern, and, as a result, achieves a higher level of accuracy.

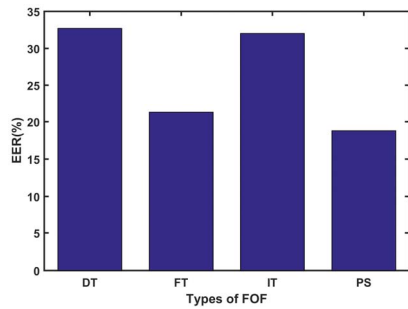


Fig. 3. EER values for different types of FOF.

A better way of understanding the accuracy performances achieved by different types of FOF is to visualise the feature values from different subjects graphically. Fig. 4 shows the feature scatter plots of three types of FOF from three subjects. The subjects are randomly chosen. The x- and y-axis of each figure represents a type of FOF with the feature ID given in brackets. What is striking about the plots shown in the figure is that when PS is used (shown in Fig. 4c), the three subjects can be clearly distinguished or separated. However, this is not the case for FT (shown in Fig. 4b) and DT (shown in Fig. 4a). These observations are consistent with our discussions given above, i.e. PS achieves the best accuracy performance, which is followed by FT and, then, by DT.

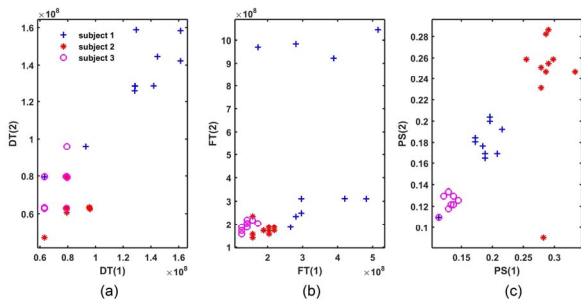


Fig. 4. Feature scatter plots of the three types of FOF: (a) DT, (b) FT, and (c) PS.

### 3) Classifier Performance

The classifiers used in our experiments can be classified into two groups, OCC and TCC. The main difference between the two groups lies in the type of training samples they each use in building authentication models (as discussed in Section III.D). Because of this, the models built by the two groups of classifiers differ in three attributes: (1) the accuracy performance, (2) the training time, and (3) the testing time. To evaluate and compare the two groups of classifiers in terms of these three attributes, we have chosen two classifiers for each group. For the OCC group, we have chosen OCKNN and SVDD, and, for the TCC group, we have chosen KNN and SVM. The input to each of these classifiers is set to be the FOF and SOF features extracted from the 4D string.

Table I shows the EER values, training times, and testing times produced by using the classifiers, and Fig. 5 presents the DET curves of the classifiers. As shown in the table the EER values produced when using OCC are higher than those when using TCC. More specifically, the EER values when using OCKNN and SVDD are 10.5% and 9.9%, respectively, whereas the corresponding values when using KNN and SVM are, respectively, 8.7% and 9.4%. This indicates that the accuracy performances of the models built by OCC are lower than those by TCC. This may be due to the fact that, unlike

OCC, the classifiers in TCC build the models with both legitimate and illegitimate samples, which means that the models can capture more information about the subjects' touch dynamics patterns, leading to more accurate models. However, it should be emphasised that the level of gain in the accuracy performance by using TCC is not significant. As shown in Fig. 5, the DET curves of OCC and TCC are somewhat close to each other.

TABLE I. EER, TRAINING, AND TESTING TIME VALUES OF FOUR CLASSIFIERS

Features	Classifier Group	EER (%)	Training Times (unit)	Testing Times (unit)
OCKNN	OCC	10.5	1	1
KNN	TCC	8.7	1	7
SVDD	OCC	9.9	3	1
SVM	TCC	9.4	22	2

Unlike the case for the accuracy performance, there is no clear correlation between a particular group of classifiers, OCC or TCC, and the model training time, rather the model training time appears to be classifier dependent. Among the four classifiers, SVM is significantly more expensive than the other three classifiers. The second most expensive classifier is SVDD, consuming 3 units of time against 1 by OCKNN and KNN. Two factors influence the model training time: (1) the nature (structure or approach) of a classifier, and (2) the number of samples a classifier uses to train a model. It seems that the first factor plays a dominant role in model training time.

With regards to the model testing time, TCC classifiers are more expensive than OCC classifiers. KNN is the most expensive one among the four classifiers; seven times more expensive than OCKNN and SVDD. SVM is second most expensive, costing twice as much as OCKNN and SVDD. Similar to the case for the model training time, it seems that the nature of a classifier plays a dominant role in model testing time.

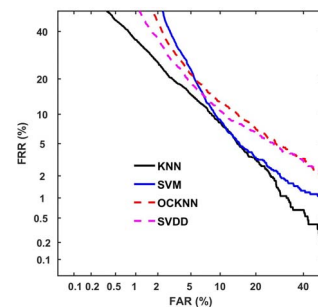


Fig. 5. DET curves of four different classifiers.

Based on the above results and discussions, particularly taking into consideration of the finding that, for a roughly similar level of accuracy performance, OCC classifiers are generally more efficient than TCC classifiers, both in terms of model training and testing times, and that, in a mobile device context, usually only the data from the owner of a device are available for use in training the classifier to build the authentication model, and performance and usability requirements are also important, we recommend the use of OCC classifiers in building an authentication model in this application context.

#### 4) With and Without Touch Dynamics

To evaluate the effectiveness and efficiency of our proposed touch dynamics authentication method, we have compared two authentication systems: one using only a 4-digit PIN (denote as AS1), and the other using both a 4-digit PIN and the touch dynamics authentication method (denote as AS2).

In the evaluation, for both AS1 and AS2, we have used the assumption that the PIN has already been exposed to an impersonator. In this case, with AS1, the probability for the impersonator to successfully gain access to the user's device is 100%. On the contrary, with AS2, this probability is reduced to 9.9%, which is a significant reduction, indicating that the two-factor authentication method can achieve a significantly higher level of security in comparison with the single-factor method. Of course, there is a price to pay for using AS2; there is a non-zero FRR, which impedes usability. With this level of security enhancement offered by AS2, 1 out of 10 legitimate login attempts may be incorrectly rejected. With AS1, the FRR is zero, as none of the login attempts will be falsely rejected as long as the PIN is entered correctly. The above evaluation results also show that, with the use of a touch dynamics based authentication method, there is a trade-off between security and usability. We leave the research question as for how to balance this trade-off to future investigation.

#### V. CONCLUSION

This paper has investigated the feasibility and effectiveness of using touch dynamics biometrics for user authentication on mobile devices. To evaluate the effectiveness of this authentication method, we have acquired a comprehensive touch dynamics dataset. This paper has explained that two types of features can be extracted, a basic set of features, FOF, that can be extracted from the raw data, and an extended set of features, SOF, that can be extracted from FOF features. The paper then describes how the features may be classified using classifiers to build authentication models. Our experimental results show that the use of OCC classifiers is more efficient for roughly the same level of security than TCC classifiers, making the OCC-based classification method more practical in real-world applications. Experimental results showed that the idea of using touch dynamics biometrics to support user authentication in a mobile device or application context is feasible. Our future work is to use deep learning techniques [23]–[26] to automatically extract representative features to build more accurate authentication models.

#### REFERENCES

- [1] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "Recognizing Your Touch: Towards Strengthening Mobile Device Authentication via Touch Dynamics Integration," in *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia*, New York, NY, USA, 2015, pp. 108–116.
- [2] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Comput. Secur.*, vol. 59, pp. 210–235, Jun. 2016.
- [3] I. de Mendizabal-Vazquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Sanchez-Avila, "Supervised classification methods applied to keystroke dynamics through mobile devices," in *2014 International Carnahan Conference on Security Technology (ICCST)*, 2014, pp. 1–6.
- [4] J. Wang, J. Tang, G. Xue, and D. Yang, "Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems," *Comput. Netw.*, vol. 115, pp. 100–109, Mar. 2017.
- [5] S. Sen and K. Muralidharan, "Putting 'pressure' on mobile authentication," in *2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, 2014, pp. 56–61.
- [6] T.-Y. Chang, C.-J. Tsai, W.-J. Tsai, C.-C. Peng, and H.-S. Wu, "A changeable personal identification number-based keystroke dynamics authentication system on smart phones," *Secur. Commun. Netw.*, p. n/a-n/a, May 2015.
- [7] 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, Mar. 2016.
- [8] G. Ho, "TapDynamics: Strengthening User Authentication on Mobile Phones with Keystroke Dynamics," Stanford University, 2013.
- [9] A. Buriro, B. Crispo, F. D. Frari, and K. Wrona, "Touchstroke: Smartphone User Authentication Based on Touch-Typing Biometrics," in *New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops*, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Springer International Publishing, 2015, pp. 27–34.
- [10] N. Zheng, K. Bai, H. Huang, and H. Wang, "You Are How You Touch: User Verification on Smartphones via Tapping Behaviors," in *2014 IEEE 22nd International Conference on Network Protocols (ICNP)*, 2014, pp. 221–232.
- [11] J. Wu and Z. Chen, "An Implicit Identity Authentication System Considering Changes of Gesture Based on Keystroke Behaviors," *Int. J. Distrib. Sens. Netw.*, vol. 2015, p. e470274, Jun. 2015.
- [12] A. Buriro, S. Gupta, and B. Crispo, "Evaluation of Motion-Based Touch-Typing Biometrics for Online Banking," in *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2017, pp. 1–5.
- [13] T. K. Ho, "Nearest neighbors in random subspaces," in *Advances in Pattern Recognition*, A. Amin, D. Dori, P. Pudil, and H. Freeman, Eds. Springer Berlin Heidelberg, 1998, pp. 640–648.
- [14] Prem S. Mann, *Introductory Statistics*, 9th Edition. Wiley, 2016.
- [15] C. Bellinger, S. Sharma, and N. Japkowicz, "One-Class versus Binary Classification: Which and When?," in *2012 11th International Conference on Machine Learning and Applications (ICMLA)*, 2012, vol. 2, pp. 102–106.
- [16] D. M. J. Tax, "One-class classification," Ph.D. thesis, Delft University of Technology, 2001.
- [17] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.
- [18] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans Intell Syst Technol*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.
- [19] D. M. J. Tax, *DDtools 2.1.2, the Data Description Toolbox for Matlab*. 2015.
- [20] R. P. W. Duin and E. Pekalska, *PRTools 5.3.1, A Matlab Toolbox for Pattern Recognition*. 2015.
- [21] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD, 1997.
- [22] J. H. Huh, H. Kim, R. B. Bobba, M. N. Bashir, and K. Beznosov, "On the Memorability of System-generated PINs: Can Chunking Help?," in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, Ottawa, 2015, pp. 197–209.
- [23] N. C. Tay, T. Connie, T. S. Ong, K. O. M. Goh, and P. S. Teh, "A Robust Abnormal Behavior Detection Method Using Convolutional Neural Network," in *Computational Science and Technology*, 2019, pp. 37–47.
- [24] M. Jahangir, H. Afzal, M. Ahmed, K. Khurshid, and R. Nawaz, "An expert system for diabetes prediction using auto tuned multi-layer perceptron," in *2017 Intelligent Systems Conference (IntelliSys)*, 2017, pp. 722–728.
- [25] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep Learning Face Representation by Joint Identification-verification," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA, 2014, pp. 1988–1996.
- [26] R. Yunus et al., "A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques," *IEEE Access*, vol. 7, pp. 2643–2652, 2019.