Applied Mathematics and Computational Intelligence
Volume 13, No. 3, 2024 [26 – 65]

A Dynamic Selection Method for Touch-Based Continuous Authentication On Mobile Devices

Ahmad Zairi Zaidi[1] and Chun Yong Chong[1*]

[1]School of Information Technology, Monash University Malaysia, Jalan Lagoon Selatan, Bandar Sunway, Subang Jaya, 47500 Selangor, Malaysia

* Corresponding author: chong.chunyong@monash.edu

Received: 10 February 2024
Revised: 30 May 2024
Accepted: 11 July 2024

ABSTRACT

Touch biometric is one of the promising modalities to realise continuous authentication (CA) on mobile devices by distinguishing between touch strokes performed by legitimate and illegitimate users. While the benefit of the scheme is promising, the effectiveness of different classification methods is not thoroughly understood. Particularly, little consideration has been given to dynamic selection of classifiers. In this paper, we proposed a dynamic selection method to deal with the security and usability needs of touch-based CA. Instead of classifying all touch samples using the same classifier, our method dynamically selects the most promising classifiers from a pool based on a competence measure. The classifiers that achieved the highest level of competence will be selected to perform the classification task for a particular test sample. We used four publicly accessible touch biometric datasets (Frank, Serwadda, Antal, and Mahbub) consisting of swipe gesture data collected from various environments and tasks. We conducted a comparative analysis of the proposed method against nine other DS methods, six well-known single classifiers (K-Nearest Neighbour, Support Vector Machine, Decision Tree , Naive Bayes, Logistic Regression and Neural Network), as well as four static ensemble methods. We evaluated the methods using equal error rate (EER) as the primary evaluation metric. The experimental results demonstrated the potential and feasibility of the proposed method, showing that it can improve the authentication performance of touch-based CA with a relatively low EER in many scenarios across multiple datasets, exhibiting relatively high consistency.

Keywords: continuous authentication, dynamic classifier selection, mobile device security, multiple classifier system, touch biometric

1 INTRODUCTION

In recent years, mobile devices have evolved into mainstream devices for most people. Besides phone calls and text messages, mobile devices (particularly smartphones) are essential for personal and professional uses. However, the device is easily lost or stolen due to its portability, which could result in information leakage and financial loss. Therefore, some security mechanism is required to guarantee that the data stored on the devices is secure.

Conventional password-based authentication schemes are still widely used (e.g. PIN and swipe pattern code). These schemes, however, have several disadvantages, including shoulder surfing, [1], easily-guessed password [2], and smudge attack [3]. Biometrics authentication schemes like fingerprint and facial recognition are also used nowadays to overcome the drawbacks of password-based authentication schemes. However, since these schemes need specialised hardware, the implementation costs for these biometrics-based authentication schemes could be higher (e.g. fingerprint scanner and front-facing camera) [4]. On top of that, only initial-login authentication is possible with the authentication schemes mentioned earlier. If an illegitimate user bypassed the initial-login authentication, the device would lose its ability to remain to recognise unauthorised access. As a result, to complement the existing authentication schemes, an additional authentication layer is necessary.

An authentication method known as continuous authentication (CA) allows for continuous device usage monitoring [5]. As one of the potential authentication methods, studies on CA based on touch biometrics are growing. [6]. Touch-based CA makes it possible for the user to use the device while the authentication process runs silently in the background [7]. Furthermore, it is an authentication scheme that is non-intrusive and does not involve installing any specialised hardware [8]. It is, therefore, a promising authentication method to support the initial-login authentication. Furthermore, numerous studies [7–14] have demonstrated the discriminative power of touch biometric in distinguishing legitimate and illegitimate users. The legitimacy of users can be recognised using a classification algorithm, which can distinguish the behavioural traits obtained from touch operations.

1.1 Motivation

Various classification methods have been utilised for user classification in touch-based CA on mobile devices [7–9, 11, 13–16]. While a particular classification method can outperform other methods in various studies, the study on classification methods in touch-based CA is still open for discussion. Generally, existing studies have yet to agree on which classification method is preferable. Few studies in the domain (especially early ones) used more than one dataset [7–9, 11, 13–21]. These studies either employed their private datasets [8, 11, 13, 15] or publicly accessible datasets [7, 9, 14, 16–21]. This finding demonstrates that the classification methods were not benchmarked across multiple datasets. To better understand the classification methods, it is vital to use public datasets, particularly those with various feature sets, as benchmark datasets. Besides, even though certain studies [10, 22, 23] used many datasets (at least two datasets), these studies only tested a few classification algorithms (less than five). On the other hand, one study [9] investigated a wide range of classification techniques (10 methods). However, the study only utilised one dataset.

According to the ”no free lunch” theorem [24], no single classifier can solve all classification problems. Consequently, depending on a single classifier to perform authentication decisions may result in the inconsistent performance of the authentication scheme. Studies comparing different classification methods based on Multiple Classifier Systems in the area of touch-based CA are still lacking. Multiple Classifier Systems (MCS) (also known as ensemble learning technique) have the advantage of smoothing out the weaknesses of single classifiers. Since not all classification problems can be solved by a single classifier and a specific algorithm uses a particular approach to approximate the feature vectors and the respective class labels, several classifiers can complement one another [25]. In touch CA, studies [8, 14] have demonstrated that Random Forest (RF), an ensemble learning technique based on multiple decision trees, yielded promising outcomes. However, RF employs a homogeneous pool of decision trees. It builds a diverse tree to generate the classifiers pool rather than integrating them from pre-defined base classifiers. [26]. The classification of test samples will likewise be performed using the same ensemble model (similar to a single classifier).

It is crucial to improve the classification performance of the CA scheme because a classification method is one of the factors that influence the overall performance of the CA scheme [27]. In this study, we employed Dynamic Selection (DS) technique to enhance the classification performance of touch-based CA. Instead of classifying all test samples with the same classifier, a method in MCS known as DS performs classifiers selection for each test sample. This method involves three steps: (1) generating a pool of classifiers, (2) defining the region of competence for each test sample, and (3) selecting and aggregating the most competent classifiers for the final classification decision..

In the classifiers generation phase (Phase 1), we generated the pool of classifiers using some of the classifiers that have been widely used in the literature, which include K-Nearest Neighbour, Support Vector Machine, Decision Tree , Naive Bayes, Logistic Regression and Neural Network. In the definition of competence region phase (Phase 2), the region of competence was defined based on the K-nearest neighbours of the test sample to be classified. Lastly, in the selection and aggregation phase (Phase 3), the base classifiers in the pool were selected based on the proposed measure of competence in order to keep the most promising classifiers and prune the less promising ones, and aggregating them using simple or weighted majority voting.

A classification framework for touch-based CA utilising the DS method has also been proposed by Zaidi et al. [28]. However, to the best of our knowledge, no studies have specifically examined the measure of competence in DS for touch-based CA. Numerous measures of competence have been presented in the literature in other domains, especially in the application of DS in various domains [29–32]. Since touch-based CA is a specific area of research, it is essential to propose a measure of competence appropriate in the context of the domain. Furthermore, the primary goal of a touch-based CA scheme is to prevent illegitimate users from accessing the device in case of an intrusion while ensuring that the legitimate user can continue using it normally without interruption. Therefore, it is essential to determine a measure of competence to accomplish this goal.

1.2 Objective and Contribution

This paper proposes Dynamic Ensemble Selection for Touch-based CA (DESTOUCH), a DS method for user classification in touch-based CA. To the best of our knowledge, very little study has explored DS in touch-based CA [28]. While other biometrics modalities have explored DS as a classification method [33, 34], we believe it is worth exploring such a method in touch-based CA as well. Therefore, the research objectives (RO) of this study are:

- **RO1: To propose a measure of competence for a DS method in touch-based CA**.

  Developing an effective measure of competence is crucial for the effectiveness of the DS method. This measure will determine how well each classifier can distinguish between the touch strokes performed by legitimate and illegitimate users. By proposing a new measure, this study aims to enhance the security and usability of touch-based CA. This will assist the improvement of mobile device security by ensuring that the authentication system can adapt to varying user behaviours.

- **RO2: To employ various selection and aggregation approaches for the generalisation phase of the DS method.**

  The selection and aggregation of classifiers are key steps in the DS method. By employing and comparing different approaches, this study aims to identify the most effective strategies for selecting and combining classifiers. Achieving optimal selection and aggregation approaches will enhance the overall performance of the CA system, making it more robust.

- **RO3: To evaluate the proposed scheme against single classifiers, static ensemble methods, and other DS methods.**

  Evaluating the proposed DS method against existing methods is essential to demonstrate its effectiveness and potential advantages. By conducting comprehensive evaluations, this study aims to provide empirical evidence that DESTOUCH outperforms traditional single classifiers, static ensembles, and other DS methods.

The measure of competence is designed to estimate the competence level of base classifiers based on the probability of true acceptance of legitimate users and the true rejection of illegitimate users. Our idea is based on the main aim of a touch-based CA, which is to detect unauthorised access by illegitimate users and to detect the usage of the device by a legitimate user. The former prevents illegitimate users from accessing the device, while the latter ensures the legitimate user can use the device normally without interruption. Therefore, a base classifier has to be able to detect the touch strokes of both legitimate and illegitimate users at a high detection rate. In this case, the probability of correct classification of legitimate and illegitimate samples, respectively, represents true acceptance and true rejection.

Furthermore, the outstanding outcomes of DS methods in other domains, such as keystroke dynamics recognition [34], lip-based biometric verification [33], signature verification [35], face recognition [31], hand-digit recognition [36], remote sensing [37], credit scoring [32, 38, 39], and process monitoring [40] served as inspiration for this work. The following is the summary of the main contributions of this paper:

1. A proposal of a measure of competence for dynamic selection of classifiers in touch-based CA..

2. Comparisons of selection approaches based on ranking and threshold as well as aggregation approaches based on simple and weighted majority voting rules.

3. Comparisons of the proposed methods against other static and dynamic selection methods.

It is worth noting that, the proposed DS method has the potential use in various scenarios to enhance mobile device security. For instance, it can be integrated into the existing mobile operating systems to provide an supplementary layer of security, ensuring that illegitimate users are detected and locked out in real-time without requiring additional hardware. This method can be valuable in personal and business scenarios, where sensitive data can be well-protected. Furthermore, it can also be applied in mobile banking and payment systems to prevent unauthorized transactions, to reduce the risk of financial fraud.

The rest of this paper is structured as follows. First, the background on touch-based CA is presented, along with related works, in section 2. Next, we present the proposed classification method in Section 3. Then, the experimental setup utilised in this study is described in section 4. Next, the findings of the experiments are discussed in section 5. Finally, we conclude our findings and explore some potential future works in Section 6.

2 BACKGROUND AND RELATED WORK

This section presents a general review of touch-based continuous authentication on mobile devices, followed by user classification in this domain and related studies. An overview of Multiple Classifier Systems is also provided.

2.1 Touch-based Continuous Authentication on Mobile Devices

A user authentication scheme called continuous authentication (CA) enables continuous monitoring while a user is using the device. After the user completed the initial login session, the CA scheme is initiated. The main goal of the CA scheme is to prevent illegitimate users from accessing the device once it has been recognised that the user is not legitimate. Touch actions on the touch screen of a mobile device are behavioural biometrics, where biometric data can be obtained while a user is using the device. Unlike physiological biometrics like the face and fingerprint, which require the user to pay attention during the data acquisition phase, touch biometric data can be collected silently in the background. Due to the ability for transparent data acquisition, this biometric modality makes it more appropriate for CA.

There are two phases in touch-based CA: enrolment and authentication. Raw touch data acquisition marks the beginning of the enrolment phase. The touchscreen sensor of the mobile device is used to collect raw touch data such as touch coordinates, touch pressure, touch area, and timestamp. After preprocessing, a user profile will be generated from the raw data by extracting features corresponding to a user's behaviour. The user model is then stored in a database once the user's behaviour has been modelled using a classification algorithm. In the authentication stage, the features are extracted with the raw touch data from new touch samples. Then, these features will be compared to the user model using a classifier that has been previously stored to identify whether the captured touch sample comes from legitimate or illegitimate users. The primary area of interest of our study is the classification system of the CA scheme. The following section provides a detailed overview of user classification in touch-based CA.

2.2 User Classification in Touch-based CA

A touch-based CA scheme performs classification tasks to evaluate whether a specific touch stroke belongs to the legitimate user. The CA scheme will lock out the user if it detects the feature vectors belong to an illegitimate user. If not, it will let the user keep using the device. For every touch stroke performed, a vector of raw data is generated. This vector contains the touch position, pressure, area, timestamp, and finger orientation [7]. A touch stroke (touch sample) $x \in X$ of a user $\omega_l, l \in \{L, I\}$ is represented by a vector of N features $F = \{f_1, f_2, ...f_N\}$ that were extracted from the raw data. The features F of the touch sample x_j can be classified as belonging to a legitimate user ω_L or an illegitimate user ω_I using a binary classifier c_i, where L and I are the class labels for the classification problem. For the touch sample x_j, classifier c_i makes the classification decision for touch sample x_j based on a threshold θ. The touch sample x_j is classified as belonging to a legitimate user ω_L if the classification score is higher than the threshold θ, or an illegitimate user ω_I if it is below (as shown in Equation 1).

$$\gamma(x_j) = \begin{cases} \omega_L & \text{if } \lambda(x_j, c_i) \geq \tau \\ \omega_I & \text{otherwise} \end{cases} \tag{1}$$

where $\lambda(x_j, c_i)$ is the classification score for a touch sample x_j using classifier c_i and τ is the threshold.

This domain has made use of numerous classification methods. Work by Frank et al. [7] was among the earliest examples of touch-based CA. The study presented a framework for classifying mobile device users based on their interaction with touchscreens. Thirty behavioural features derived from vertical and horizontal strokes were presented. The authors used Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) classifiers to differentiate between the behavioural features of legitimate and illegitimate users. The classifiers achieved an equal error rate (EER) ranging from 0.00% to 4.00%.

Li et al. [15] proposed a CA scheme using both tap and swipe gestures. Several features were presented for each type of gesture from 75 users of mobile phones. There were no predefined tasks for the users to perform when using the device. The swipe and tap gestures, respectively, yielded 13 and three features. SVM was the only classifier used in the study. For the sliding-up gesture, it achieved a minimum accuracy of 95.78%.

In contrast to the study by Frank et al., [7] and Li et al. [15], Serwadda et al. [9] conducted a benchmark evaluation on a touch dataset employing 28 features with ten classifiers. SVM, Naive Bayes, Random Forest, KNN, Bayesian Network, Neural Network, Decision Tree, Logistic Regression, Scaled Manhattan, and Euclidean Verifier was used by the authors to evaluate data from 190 subjects. They discovered that LR performed the best when they evaluated horizontal strokes in landscape screen orientation, which produced an EER of 10.50%.

Shen et al. [8] investigated touch-based CA by taking into account various touch operations (i.e. up, down, left, and right) across various application tasks (i.e. document reading, picture browsing, web browsing, and free task), as well as on various application scenarios (i.e. short, middle, relative-long, and long periods of authentication). The authors built four classifiers: KNN, SVM, Backward Propagation Neural Network (BPNN), and RF, based on 58 behavioural features. According to their findings, RF achieved an EER of about 1.80% on the left and right touch operations.

Three primary mobile device sensors — the front camera, touchscreen, and location sensors — were used by Mahbub et al. [16] to gather raw data. On the data collected from each sensor, they performed the experiments separately. The authors gathered swipe actions from users for the touchscreen sensor without any predefined task. Each touch stroke yielded 24 features. Seven classifiers were used: KNN, SVM, NB, Linear Regression, Random Tree with Linear Regression, RF, and Gradient Boosting Model. With an EER of 22.10%, the results demonstrate that RF performed better than other classifiers.

By considering various touch operations, Fierrez et al. [10] looked into the effectiveness of touch-based CA by evaluating the performance of three different scenarios. These scenarios include an intra-session, inter-session, and a combination of the two. A session here is when a user begins using the device and lasts until the user stops using it for a predetermined time. When a classifier is trained and tested during the same session, this is referred to as an intra-session scenario. On the other hand, a scenario where a classifier is trained and tested across different sessions is referred to as an inter-session scenario. The authors used 28 features from Serwadda et al. [9] and another five features from Martinez-Diaz et al. [41]. In their studies, SVM, Gaussian Mixture Model (GMM), and the combination of these two classifiers were used. The performance of the fusion method was generally superior to single classifiers. For a single classifier, GMM outperformed SVM with an EER of 3.60% for right swipe touch operations in an intra-session scenario using one of the selected datasets.

Meng et al. [11] studied the performance of a touch-based CA in two scenarios. First, based on users' free device usage and second, web browsing. They conducted a comparative analysis utilising five classifiers — DT, NB, Kstar, Radial Basis Function Network (RBFN), and BPNN — using the 21 touch features that were extracted from 48 participants. Their study also used Particle Swarm Optimisation with RBFN (PSO-RBFN). The authors discovered that compared to free usage, web browsing exhibits less variance in behaviour, with PSO-RBFN doing the best with an EER of 2.38

Syed et al. [14] investigated the effect of user posture, device size, and device configuration on the performance of touch-based CA. The authors used five classifiers: SVM, LR, NB, RF, and Multilayer Perception Neural Network (MLP). They presented 14 features. They discovered that RF yields the best outcomes. Furthermore, when the model was trained and tested using the same posture, which is when the device was held in landscape orientation, the best EER was recorded at 3.80%.

Incel et al. [42] presented DAKOTA, a mobile banking application-based CA that can collect behavioural biometric data through touchscreen and motion sensors. The authors tested the proposed scheme using nine classifiers, including binary SVM, one-class SVM, KNN, MLP, DT, RF, NB, an ensemble of SVM and MLP, and an ensemble of SVM Polynomial kernel and SVM RBF kernel, with 126 features combined from both touchscreen and motion sensors. According to the authors, a binary SVM with an RBF kernel has an EER of 3.50%.

Aaby et al. [43] introduced an omnidirectional approach to touch-based CA. Unlike traditional methods that depend on touch direction, this study focussed on an omnidirectional approach. where the model processed touch data without categorising it by direction. The study evaluated various behavioural feature sets using SVM, KNN, RF, Extra-Trees and Gradient Boosting classifiers. The results demonstrated that Extra-Trees classifier outperformed the traditional methods with an EER of 0.179 when combining five strokes.

Shen et al. [44] introduced IncreAuth, an incremental learning-based CA framework that can perform authentication over long-term device usage. It leveraged a context-aware feature set to characterise touch patterns under complex usage contexts and integrated a Gradient Boosting Decision Tree with a Neural Network (GBDTNN) for efficient online updates. Experimental results demonstrated that the framework achieved a stable authentication performance with low system overheads in a long-term device usage scenario, achieving an EER of 8.77%.

Table 1 summarises the performance of the classifiers used in several studies based on equal error rate (EER), average error rate (AER) or classification accuracy (ACC) (✓ indicates that the classifier has been employed in the study and ☑ indicates the classifier was the best in that study). In general, single classifiers were used to perform user classification in the literature reviewed above. These include:

- ***k*-Nearest Neighbour (*k*NN) [45]**: An instance-based classification method that assumes the new sample of touch stroke from the test set is similar to the data in training set. The algorithm finds the touch strokes in the training samples that are close to the touch strokes from test set based on a Euclidean distance measure.

- **Support Vector Machine (SVM) [46]:** A discriminative classification method that separates the features of a legitimate user and illegitimate users using maximised hyperplane [47]. A new touch sample will be mapped into the separated space and classified to belong as the sample from a legitimate or an illegitimate user.

- **Decision Tree (DT) [48]:** A non-parametric classification method that creates a tree model. The tree is created by choosing features as the decision nodes. A touch sample is classified based on these nodes.

- **Naive Bayes (NB) [49]:** A probabilistic method based on Bayes theorem [50]. It classifies a touch stroke based on the probability that it belongs to a particular class.

- **Logistic Regression (LR) [51]:** A statistical method based on linear regression where the prediction of the legitimate user is transformed using the logistic function. Touch samples from the training data estimate the coefficients of the model using maximum-likelihood estimation.

- **Artificial Neural Network (ANN) [49]:** This method was inspired by the neural network of the human brain. It consists of an input, hidden layers, and an output [52]. The neurons in the input layer receive touch features of each user where the algorithm assigns each neuron with a weight based on a particular function. This information is transferred within the hidden layers. The algorithm produces an output at the output layer after several iterations..

While some classification methods have demonstrated superior performance in certain studies, the discussion over the most effective classification methods for touch-based CA remains unresolved. In general, there is no consensus in existing literature on the most preferable classification method. Notably, only a few studies, especially the earlier ones, utilised more than one dataset [7–9, 11, 13–21]. These studies typically relied on either private datasets [8, 11, 13, 15] or publicly available datasets [7, 9, 14, 16–21], indicating that classification methods were not consistently benchmarked across multiple datasets. For a more comprehensive understanding of classification methods, it is essential to employ public datasets with diverse feature sets as benchmarks. Furthermore, although some studies [10, 22, 23] used many datasets (at least two datasets), these studies only tested a few

Table 1 : Performance of classifiers in the chosen literature

Author	Dataset	Feature	Classifiers	SVM	KNN	DT	NB	LR	NN	RF	Performance (%)
Frank et al. [7]	1	27	2	✓	✓	-	-	-	-	-	EER = 0.00-4.00
Li et al. [15]	1	16	1	✓	-	-		-	-	-	ACC = 95.78
Serwadda et al. [9]	1	28	10	✓	✓	✓	✓	☑	✓	✓	EER = 10.50
Shen et al. [8]	1	58	4	✓	✓	-	-	-	✓	☑	EER = 1.80
Mahbub et al. [16]	1	24	7	✓	✓	-	✓	✓	-	☑	EER = 22.10
Fierrez et al. [10]	4	33	2	✓	-	-	-	-	-	-	EER = 3.60
Meng et al. [11]	1	21	6	-	-	✓	✓	-	☑	-	AER = 2.38
Meng et al. [13]	1	9	5	☑	-	✓	✓	-	✓	-	AER = 4.66
Syed et al. [14]	1	14	5	✓	-	-	✓	✓	✓	☑	EER = 3.80
Incel et al. [42]	1	126	9	☑	✓	✓	✓	-	✓	✓	EER = 3.50
Aaby et al. [43]	1	76	5	✓	✓	-	-	-	-	✓	EER = 17.90
Shen et al. [44]	1	17	5	✓	-	-	-	-	✓	✓	EER = 8.77

classification methods (less than five). On the other hand, one study [9] explored a broad range of classification techniques (10 methods), but only applied them to a single dataset. It is worth noting that there are several datasets have been made publicly available by some studies, which include Frank Dataset [7], Serwadda Dataset [9], Antal Dataset [53], and Mahbub Dataset [16]. The description of these datasets can be found in Appendix A and can also be obtained from Fierrez et al. [10].

The related studies above have explored various classification methods for touch-based CA. Each study shows effectiveness of different classifiers and the challenges associated with touch data. However, a common limitation among these studies is their reliance on static classification methods, which do not adapt to the dynamic and variable nature of touch interactions. Specifically, the performance of a particular classifier differed from a study to another. Variations in the experimental design, feature extraction, and data collection procedure might cause this problem. In addition, the performance of the classifiers may be affected by intra-class variability of touch data. As a result, using the same classifier in various scenarios could cause the CA scheme to operate inconsistently.

Using Multiple Classifier Systems (MCS), also known as the ensemble learning technique, could be one potential approach for overcoming the limitation of single classifiers. Some studies in touch-based CA [8, 9, 14, 16] have attempted to apply RF, an ensemble learning technique based on decision tree. However, research on MCS in the area of touch-based CA is still lacking. The importance of MCS in this area lies in its ability to address the limitations of single classifiers by leveraging the strengths of diverse classifiers. MCS allows the integration of various classifiers to improve overall classification performance. In particular, Dynamic Selection (DS) technique enhances this approach by dynamically selecting the most promising classifiers based on their competence, thereby adapting to the varying characteristics of the touch data. This adaptability is crucial for touch-based CA, where user behaviour can be highly variable. By employing DS, the proposed method in our study aims to achieve better authentication performance, making them significant advancements in the area of touch-based CA. Therefore, this paper employed DS method, which is a method under MCS, to enhance the classification performance in touch-based CA. An overview of MCS and DS is provided in the following section.

2.3 Multiple Classifier Systems (MCS)

In order to overcome the limitations of single classifiers, Multiple Classifier Systems (MCS), also known as ensemble learning technique, produces classification decisions based on the combination of more than one classifiers [54–56]. In the literature, numerous classification methods have been proposed. However, it is well acknowledged that no single classification algorithm can handle all classification problems effectively and efficiently [57]. It is worth noting that satisfactory results of MCS can be achieved if the base classifiers (comprised of multiple single classifiers) are of high classification capability and diverse. The scheme will produce several models, and a decision will be made based either on classifier fusion or classifier selection [58].

In classifier fusion, each classifier in the pool contributes towards the final decision [59]. There are various ways to accomplish the fusion, which include minimum, maximum, average, median, and majority vote strategies. This method will perform less effectively if the pool contains some redundant and inaccurate classifiers [58]. On the other hand, classifier selection makes the final classification decision based on the selection of a single classifier or a subset of classifiers [58]. Classifier selection can perform better than classifier fusion because it selects the single classifiers from the pool that show the most promising ones rather than smoothing out the differences between individual classifiers [40]. Static selection and dynamic selection (DS) are the two main methods for selecting classifiers [57, 60].

Static classifier selection selects the classifier(s) during training phase by using the same classifier(s) to classify all test samples. On the other hand, DS selects the most promising classifier(s) for each test sample during the test phase. Additionally, promising classifiers are selected based on the competence level of a test sample in each competence region. It is worth noting that test samples typically exhibit varying levels of classification difficulty. As a result, using the appropriate classifier for each test sample is advantageous compared to using the same classifiers for all test samples because each classifier has distinct expertise with each classification task [61]. DS typically involves the following components:

1. **Pool Generation:** A pool of multiple classifiers can be generated based on different methods such as homogeneous or heterogeneous classifiers. A pool of homogeneous classifiers can be generated based on algorithm initialisations, parameter settings, algorithm architectures, training sets, and feature sets [60]. On the other hand, pool of heterogeneous classifiers can be generated based on different types of classification algorithms (eg. SVM, kNN and etc.) [60].

2. **Region of Competence Definition:** The region of competence of a base classifier is first defined based on different parts of the feature spaces. The region of competence of a test sample is the data points in the validation set. The feature space of the validation set is divided into different regions where the most competent classifier is determined based on these partitions.

3. **Competence Estimation:** In each region of competence, the most competent classifier will be determined based on a certain measure of competence. During the test phase, the scheme will determine which local region where a test sample belongs to and perform classification based on the most competent classifier for that region.

4. **Classifiers Selection:** In the selection phase, one or more classifiers will be selected to perform the classification task based the competency of the classifiers in the pool. The selection can be performed using different approaches, such as as ranking [36], accuracy [37, 59], accuracy and diversity [62], probabilistic [26, 58, 63, 64], classifier behaviour [65, 66], Oracle [67], and meta-learning [68].

5. **Classifiers Aggregation:** In the aggregation phase, if more than one classifier were selected, a fusion method is used to make the final decision [57]. Combination rules such as majority voting, maximum, minimum, or trainable fusers (e.g., stacking) are usually used. In the case where only one classifier is selected , no aggregation is needed [57].

The advantages of DS methods can be observed in various biometrics studies. For example, in keystroke dynamics recognition, DS methods have shown superior performance in terms of accuracy and reliability compared to single classifier [34]. In lip-based biometric verification, DS methods have demonstrated a better verification rates by dynamically selecting the most competent classifiers [33]. Similarly, in signature-based verification, DS methods have been effective in handling the variability in samples and improving the overall recognition rates [35]. The application of DS in various domains has also been notable such as face recognition [31], hand-digit recognition [36], remote sensing [37], credit scoring [32, 38, 39], and process monitoring [40]. These studies highlight the potential use of DS methods across various biometrics and non-biometrics domains.

Inspired by the outstanding results of DS methods in enhancing classification performance in other domains, it is worth to explore DS methods in touch-based CA as well. In touch-based CA, Zaidi et al. [28] have proposed a classification framework for touch-based CA using DS methods. The author benchmarked existing DS methods and applied them in touch-based CA. It was found that DS methods are more consistent compared to single classifiers. However, to the best of our knowledge, no studies have specifically examined the measure of competence in DS for touch-based CA. Given that touch-based CA is a specialized research area, it is crucial to develop a competence measure tailored to this domain.

3 PROPOSED METHOD

In this section, we present our proposed method, Dynamic Ensemble Selection for Touch-based CA (DESTOUCH). First, we provide an overview of a probabilistic method for measuring the competence level of classifiers. Then, we describe our proposed method consisting of three key components: measure of competence, selection approach, and aggregation approach. Finally, we present the algorithms for the methods based on the proposed measure of competence, selection approach, and aggregation approach.

Our proposed method aims to improve the classification performance of touch-based CA by leveraging the strengths of multiple classifiers. Dynamic selection of classifiers ensures that the most competent classifier(s) are selected for each touch sample, addressing the inherent variability in user behaviour and the limitations of relying on a single classifier. This approach aims enhances both the security and usability of the CA scheme.

3.1 Preliminaries

A DS method selects a subset of classifiers C' from a pool $C = \{c_1, ..., c_M\}$ of M classifiers in order to classify a test sample x_j. The method estimates the competence level δ_{ij} of each base classifier $c_i \in C$ for the test sample x_j [60]. The competence level δ_{ij} is estimated using a measure of competence in the region of competence θ_j of the test sample x_j. During the test phase, the method will define to which local region the test sample x_j belongs and then classify the sample using the most competent classifiers for that region.

First, the region of competence θ_j around test sample x_j is defined. The region of competence θ_j is defined as the K samples in the validation set D_{val} closest to x_j. To obtain the K-nearest neighbour of x_j, the distance between x_j and every data point in D_{val} was calculated. Determining the value of K is a difficult task because the number chosen can affect the efficacy of the DS algorithm [65]. The region of competence can be defined as in Equation 2.

$$\theta_j = \{x_1, x_2, \ldots, x_K\} \tag{2}$$

where K is the number of nearest neighbours of test sample x_j in a validation set D_{val}. Based on the accuracy of correct classification of any test sample x_j in the region of competence θ_j by a classifier c_i, the competence measure can be estimated by Equation 3 [59].

$$ACC(correct_i) = \frac{N_i}{K}, i = 1, ..., M \tag{3}$$

where N_i is the number of samples in local region θ_j that are correctly classified by each classifier $c_i \in C$ of M base classifiers. Since the base classifiers can generate the probability of correct classification (prediction confidence) as the output, we can reformulate Equation 3 as in Equation 4 [63]:

$$P(correct_i) = \frac{1}{K}\sum_{k=1}^{K} P(\omega_l|x_k) \tag{4}$$

where ω_l is the class label of x_k. In our case, the class labels are $l \in \{L, I\}$, where L and I denote the class labels for legitimate user and illegitimate user, respectively. Also, $P(\omega_l|x_k)$ is the probability of correct classification of sample $x_k \in \theta_j$, which is generated by the classifier c_i.

$P(\omega_l|x_k)$ can be utilised to formulate the measure of competence of a base classifier c_j for a test sample x_j in the region of competence θ_j. It represents the degree of confidence of x_k belongs to ω_l. Besides, a probabilistic-based measure of competence can overcome the limitation of local accuracy-based measure of competence that gives equal weight in generating the classification output that solely based on class labels [69]. This advantage is helpful when the region of competence θ_j contains noisy samples. Therefore, this measure can be assigned by weight to each sample $x_k \in \theta_j$. The purpose of assigning the weight is to handle the uncertainty in defining the size of the region of competence θ_j [63]. Equation 5 shows the measure of competence of correct classification in the region of competence θ_j.

$$\delta_{ij} = \frac{\sum_{k=1}^{K} P(\omega_l|x_k) \cdot W_k}{\sum_{k=1}^{K} W_k} \tag{5}$$

where $W_k = \frac{1}{d_k}$, and d_k is the distance from a test sample $x_j \in D_{test}$ to the sample $x_k \in \theta_j$. By introducing this weight, the sample $x_k \in \theta_j$ has an influence on a test sample x_j in the region of competence θ_j, where the closer ones have more influence on the estimation of competence level.

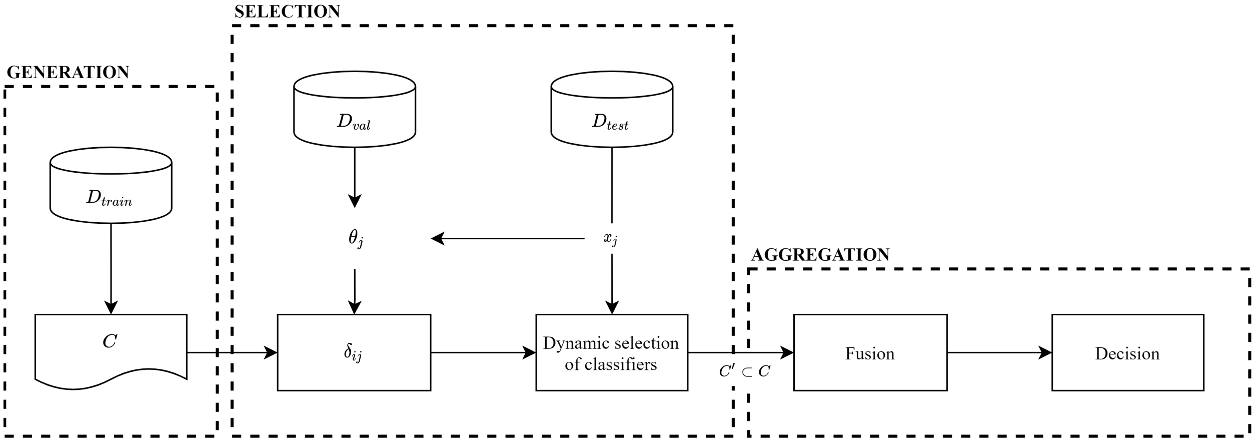


Figure 1 : Framework of DESTOUCH

It is worth noting that our study focuses on the selection criteria based on probabilistic competence measures of classifiers because the recent study in touch-based CA demonstrates promising results [28]. On top of that, Figure 1 shows the framework of our proposed method. During the training phase, the method generates a pool of heterogeneous classifiers $C = \{C_1, ..., C_M\}$. During the test phase, the method defines the region of competence θ_j for a test sample x_j using KNN algorithm. Based on this region, the method will estimate the level of competence δ_{ij} of each classifier $c_i \in C$ based on the proposed measure of competence (will be described in Section 3.2). Then, the subset of the most competent classifiers $C' \subset C$ will be selected and aggregated for the final classification decision (x_j belong to ω_L or ω_I). We also employed various selection and aggregation approaches, as described in Section 3.3 and Section 3.4, respectively.

3.2 Level of Competence Estimation

In each region of competence θ_j, the most competent classifier c_i will be selected based on a measure of competence δ_{ij}. During the test phase, the scheme will define the local region to which a test sample belongs and classify the sample using the most competent classifiers for that region. The validation set D_{val} is utilised to evaluate each base classifier x_j.

In touch-based CA, the main goal is to lock out illegitimate users from accessing the device and allow the legitimate user to continue using the device. The scheme should be able to classify the touch strokes of the legitimate user or illegitimate user with a high probability by utilising a classification algorithm. Besides, the classification problem is viewed as a binary classification, where a test sample (touch stroke) x_j will be classified as belongs to the legitimate user ω_L or illegitimate ω_I. Therefore, to improve the classification performance, we proposed DESTOUCH, a DS method that aims to achieve this goal. In DESTOUCH, for each test sample x_j, it first finds the K nearest neighbours of x_j with the samples in the validation set D_{val} to define the region of competence region θ_j. Based on the samples in θ_j, the proposed method will separate the samples according to its original class labels, which are samples of legitimate user ω_L and the samples of illegitimate users ω_I.

Let θ_j^L and θ_j^I denote the subsets of the samples of the legitimate user and illegitimate users, respectively, in the region of competence θ_j ($\theta_j^L, \theta_j^I \subset \theta_j$). Based on θ_j^L and θ_j^I, we can compute the probability of correct classification of each respective subset. The probability of correct classification in subset θ_j^L (representing the true acceptance) is computed as Equation 6:

$$P_L = P(correct_i | x_k \in \theta_j^L) = \frac{\sum_{k \in \theta_j^L} P(\omega_L | x_k) \cdot W_k}{\sum_{k \in \theta_j^L} W_k} \tag{6}$$

where $W_k = \frac{1}{d_k}$, and d_k is the distance from a test sample $x_j \in D_{test}$ to the sample $x_k \in \theta_j^L$. On the other hand, the probability of correct classification in subset θ_j^I (representing the true rejection) is computed as in Equation 7.

$$P_I = P(correct_i | x_k \in \theta_j^I) = \frac{\sum_{k \in \theta_j^I} P(\omega_I | x_k) \cdot W_k}{\sum_{k \in \theta_j^I} W_k} \tag{7}$$

where $W_k = \frac{1}{d_k}$, and d_k is the distance from a test sample $x_j \in D_{test}$ to the sample $x_k \in \theta_j^I$.

We then combined the the probability of correct classification in subset θ_j^L and θ_j^I to compute the measure of competence δ_{ij} of a classifier c_i for a test sample x_j in the region of competence θ_j as Equation 8.

$$\delta_{ij} = \frac{K^L \cdot P(correct_i | x_k \in \theta_j^L) + K^I \cdot P(correct_i | x_k \in \theta_j^I)}{K} \tag{8}$$

where K^L and K^I are the number of samples in subsets θ_j^L and θ_j^I, respectively, and K is the size of the region of competence θ_j. The role of K^L and K^I here is as the weight of the measure from both subsets. The role of the weight is to ensure the scheme can handle the situation where the samples of only one of the classes exist in the region of competence θ_j. This condition might happen when the region of competence θ_j is located in a safe region, where almost all samples in the region belong to the same class [70].

3.3 Selection Approach

During the selection phase, the level of competence δ_{ij} of each base classifier c_i in the pool of classifiers C is estimated. Once the level of competence δ_{ij} has been estimated, the proposed method will perform the selection of classifiers. To do so, we employed two selection approaches for DESTOUCH, which are based on the ranking of classifiers and based on a selection threshold. Therefore, for the rest of this paper, we refer to our proposed methods as DESTOUCH-R and DESTOUCH-T, respectively:

- **DESTOUCH-R (Ranking-based):** In this selection approach, the base classifiers in the pool of classifiers C are ranked according to its level of competence δ_{ij}. The classifier c_i that has the highest level of competence δ_{ij} will be ranked first, followed by the classifier with the second highest level of competence and so on. Based on a pre-defined selection percentage ρ, the method will select the top N number of classifiers. In this case, the most competent classifiers are those classifiers that have a higher rank amongst the others.

- **DESTOUCH-T (Threshold-based):** In this selection approach, a selection threshold τ is set. The classifier c_i that has level of competence δ_{ij} larger than the threshold τ (i.e. $\delta_{ij} \geq \tau$) is considered as competent and selected to form a subset of selected classifiers $C' \subset C$. Therefore, the most competent classifiers are selected based on the classifiers that reach this minimum requirement, while the less competent classifiers will be pruned from the original pool of classifiers C.

Based on the selected classifiers $C' \subset C$, the proposed method will aggregate the output of the selected classifiers to perform the final classification decision.

3.4 Aggregation Approach

The aggregation phase consists of combining the selected subset of classifiers $C' \subset C$ using a particular fusion approach. We used an aggregation approaches based on majority voting rule to combine the selected classifiers. We chose this type of fusion approach due to its simplicity. Therefore, once the subset of the most competent classifiers $C' \subset C$ for test sample x_j have been selected, the scheme will aggregate the output of each selected classifiers for the final classification decision. We employed two different aggregation approaches in this study. For each selection approach (DESTOUCH-R and DESTOUCH-T), the proposed scheme will perform aggregation based on simple majority voting (simple MV) or weighted majority voting (weighted MV):

- **Simple MV:** In this approach, the scheme will first select the subset of the most competent classifiers C' from the pool of classifiers C for a test sample x_j. Based on selected classifiers C', the scheme aggregates the output of each selected classifier $c_i \in C'$ using simple majority voting rule [71]. The output of the simple majority voting rule is then used for the classification decision for test sample x_j (i.e. which user a touch stroke x_j belongs to).

- **Weighted MV:** In this approach, the scheme will first select the subset of most competent classifiers C' from the pool of classifiers C for a test sample x_i. Based on selected classifiers C, the scheme aggregates the output of each selected classifier $c_j \in C'$ using weighted majority voting rule [72], which weighted by the competence level δ_{ij} of a classifier c_i. Using this approach, the classification decision obtained by the selected classifiers with a higher level of competence δ_{ij} will have a greater influence on the final classification decision. The output of weighted majority voting is then used for the classification decision for the test sample x_j (i.e. which user touch stroke x_j belongs to).

It is worth to note that the main difference between the aggregation approach based on simple MV and weighted MV is that the former simply combines the output of the selected classifiers C', while the latter includes the level of competence δ_{ij} as a weight during the aggregation phase.

Algorithm 1: DESTOUCH-R

```
Input: Training set $D_{train}$, validation set $D_{val}$, test set $D_{test}$, a pool of classifiers $C$,
       neighborhood size $K$.
Output: A subset of the most competent classifiers $C' \subset C$ for each test sample $x_j \in D_{test}$.
Train M base classifiers $C = \{c_1, ..., c_M\}$ using training set $D_{train}$ ;
for each test sample $x_j \in D_{test}$ do
    Find $\theta_j$ as the $K$ nearest neighbuors of $x_j$ in $D_{val}$ ;
    Divide $\theta_j$ into two subsets: Samples of the legitimate user $\theta_j^L$ and illetitimate users $\theta_j^I$ ;
    Compute the classification proability as in Equation 6 and Equation 7, for $\theta_j^L$ and $\theta_j^I$,
      respectively ;
    Compute the level of comptence $\delta_{ij}$ for classfier $c_i \in C$ using Equation 8 ;
    Rank $c_i \in C$ according the the level of competence $\delta_{ij}$ ;
    Select a subset of the top $N$ most competent classifiers $C'$ from the pool of classfiers $C$ ;
    Combine $C'$ using one of the methods in Section 3.4 to classify $x_j$ ;
```

3.5 The Algorithm

In this summarise, the proposed methods are applied to a pool of M heterogeneous classifiers $C = \{c_1, .., c_M\}$. These classifiers are trained on the training set D_{train}. Using the validation set D_{val}, the competence level δ_{ij} of these M classifiers are estimated using the measure of competence proposed in Section 3.2. Then, for each test sample $x_j \in D_{test}$, the subset of most competent classifiers $C' \subset C$ will be selected based on ranking (Algorithm 1) or threshold (Algorithm 2), as described in Section 3.3. Finally, the selected classifiers will be aggregated using simple MV or weighted MV to perform the classification decision (see Section 3.4).

4 EXPERIMENTAL SETUP

This section describes how we set up the experiments to evaluate our proposed method. Several components are involved in the experimental setup, including datasets, DS methods setup, model training and evaluation procedures, and evaluation metrics. At the end of this section, we describe the statistical significance test employed in our study. It is worth noting that the ensemble learning

```
Algorithm 2: DESTOUCH-T
Input: Training set $D_{train}$, validation set $D_{val}$, test set $D_{test}$, a pool of classifiers $C$,
       neighborhood size $K$.
Output: A subset of the most competent classifiers $C' \subset C$ for each test sample $x_j \in D_{test}$.
Train M base classifiers $C = \{c_1, ..., c_M\}$ using training set $D_{train}$ ;
for each test sample $x_j \in D_{test}$ do
    Find $\theta_j$ as the $K$ nearest neighbuors of $x_j$ in $D_{val}$ ;
    Divide $\theta_j$ into two subsets: Samples of the legitimate user $\theta_j^L$ and illetitimate users $\theta_j^I$ ;
    Compute the classification proability as in Equation 6 and Equation 7, for $\theta_j^L$ and $\theta_j^I$
      respectively ;
    Compute the level of comptence $\delta_{ij}$ for classfier $c_i \in C$ using Equation 8 ;
    Select a subset the most competent classifier $C'$ from the pool of classfiers $C$ if $\delta_{ij} > \tau$ ;
    if $C' \neq \emptyset$ then
        Combine $C'$ using one of the methods in Section 3.4 to classify $x_j$ ;
    else
        Combine the original pool of classfiers $C$ using one of the methods in Section 3.4 to
          classify $x_j$ ;
```

library DESlib [73] was used to implement the DS methods, whereas the Scikitlearn [74] was used for other classification methods. The experiments were carried out using Microsoft Windows 10 Enterprise with a 2.4 GHz Intel(R) Xeon(R) CPU and 32 GB of RAM.

4.1 Dataset

Four publicly available touch biometric datasets were used in this study: Frank [7], Serwadda [9], Antal [53], and Mahbub [16]. These datasets, which are summarised in Table 2, consist of swipe gesture data commonly used in touch-based CA [9]. The datasets vary in terms of the number of users (subjects), the number of sessions, the duration of data collection, the setting for data collection, and the number of features. We used the original features that the authors of the dataset had first presented. Moreover, various features have different value ranges. Therefore, we transformed the feature data by scaling the values of all features in the $[0, 1]$ range using Min-Max Scaler to remove any bias from the model-building process. The description of these datasets can be found in Appendix A.

Table 2 : Summary of the selected datasets

Dataset	Subject	Session	Duration	Interval	Environment	Features
Frank [7]	41	7	25 - 50 minutes	Several minutes	Controlled	30
Serwadda [9]	190	2	-	$\geq$ 1 day	Controlled	28
Antal [53]	71	-	4 weeks	-	Controlled	15
Mahbub [16]	48	$\sim$248	1 week	-	Uncontrolled	24

4.2 Setup for DESTOUCH

In order to generate the pool of classifiers, we used some of the classifiers that have been widely used in the literature. There were six base classifiers, which include K-Nearest Neighbour (KNN) [45], Support Vector Machine (SVM) [46], Decision Tree (DT) [48], Naive Bayes (NB) [49], Logistic Regression (LR) [51] and Multi-layer Perceptron Neural Network (NN) [49]. We keep the number of base classifiers small since increasing the pool could make it more likely that an incompetent classifier will be selected [75]. Furthermore, we used a pool of heterogeneous classifiers to ensure that the pool was diverse. In contrast to an MCS with homogeneous classifiers, it needs more advanced training (e.g., various training sets and different feature sets) to produce diversity among the basic classifiers [33]. The hyper-parameters setting for each classification algorithm are described in Appendix B. Lastly, we defined the region of competence θ_j with the K-nearest neighbours algorithm. We chose $K = 7$ because it is a common value in the existing DS studies [38, 60, 76].

4.3 Model Training and Evaluation

We addressed the classification problem in this study as a binary classification task, assuming the possibility of sharing a device (i.e., kids, relatives, or friends could borrow the device) [77]. For each dataset, we chose the first subject as the legitimate user. Then, we chose some other subjects at random to be illegitimate users. We repeated this step for other subjects in the dataset, where one subject was selected as the legitimate user and others as illegitimate users. We aimed to simulate the situation in which illegitimate users have access to the device and seek to retrieve its stored data. In other words, we assumed that the device was left unlocked or that illegitimate users could bypass the initial login process, characterising random attacks as the threat model.

A training set D_{train} was used to train the base classifiers. A legitimate user's data was used to select $N = 40$ samples randomly, and $N/10$ illegitimate users were randomly selected to form the data of illegitimate users. As a result, each illegitimate user provided 10 samples in training data, which were also selected randomly. We used the same amount of training samples for all subjects to obtain the most consistent results and to exclude any bias that might arise if certain users have more or fewer samples [9]. The minimum number of samples per user for a particular experiment has been set at 60. Subjects not meeting this minimum sample in a particular experiment were excluded. Then, a classification algorithm was trained iteratively for all subjects. Additionally, the samples used to create a test set D_{test} as test samples were used as training samples for both legitimate and illegitimate users.

There are two main experimental settings for some datasets: intra-session and inter-session. The same data set was used for the training and evaluation a model in the first case. In the latter case, a model was trained with one session and tested with a later session. Additionally, we used the validation set D_{val} to measure the competence level of a classifier. Due to the small sample size of each user, the whole training set was used as the validation set [78]. A validation set was also used to define the region of competence of a test sample.

4.4 Evaluation Metric

The performance of the classification methods evaluated in this study was assessed based on authentication error rates. False acceptance rate (FAR) is the ratio of touch strokes performed by illegitimate users mistakenly identified as touch strokes performed by legitimate users to the total number of touch strokes performed by illegitimate users. On the other hand, false rejection rate (FRR) is the ratio of the total number of touch strokes performed by a legitimate user that was wrongly identified as touch strokes performed by an illegitimate user to the number of touch strokes performed by a legitimate user. FAR and FRR, respectively, measure the security and usability of a scheme.

A particular classification method is first evaluated using the metrics mentioned above. Then, a threshold can be adjusted to make the scheme more usable or secure [14]. The threshold can be lowered to increase usability with low FRR, but at the cost of high FAR (less restrictive). On the other hand, the threshold can also be increased to be more restrictive with low FAR but high FRR (less usable). In this study, we varied the threshold for each user to obtain an equal error rate (EER). EER measures the trade-off between the security and usability of a scheme. A lower value of EER suggests a better classification method. We experimented with each classification method for each user and reported the results for a particular experiment as the average EER of all users.

We evaluated the performance of classification methods based on the average score of several consecutive strokes rather than a single stroke, similar to other related studies [7–10]. It is worth noting that setting this value is a complex task. A lower number can be set to allow a faster authentication process. However, this can lead to the loss of crucial information, making it difficult for the scheme to learn the characteristics of its users. On the other side, a larger value may include more information. However, this action can allow more time for authentication, where illegitimate users can continue operating with the stolen device longer before it is locked out from the device. We used the mean score of 10 strokes in our study, similar to earlier studies [9, 10].

4.5 Statistical Tests of Significance

The statistical significance of the various classification methods was then assessed. Since each dataset has various scenarios, the classification method with the lowest average EER was ranked first for each scenario (more on this in Section 5). The method with the second-lowest EER will be ranked second, and so on. If there was a tie (i.e. more than one method yielded the same EER), their ranks were averaged. The last step is to analyse all scenarios across all datasets to determine the average rank (AR) for each classification method. Therefore, the method with the highest average rank (lowest value) across all scenarios is the best one.

The Friedman test [79] was used in this study to compare the rankings of several classification methods statistically. Since our comparisons often violate the assumptions behind parametric tests (i.e. normal distribution or homogeneity of variance), we choose this non-parametric test. It evaluates the null hypothesis that the methods under evaluation are equivalent. Various methods were evaluated using a variety of scenarios and datasets. The Friedman statistic was calculated as in Equation 9:

$$\chi_F^2 = \frac{12D}{K(K+1)}\left[\sum_j AR_j^2 - \frac{K(K+1)^2}{4}\right], \tag{9}$$

where K is the number of classification methods and D is the number of scenarios combined from all datasets. AR_j denotes the average rank of the j-th method over all the scenarios $i \in D$ (Equation 10).

$$AR_j = \frac{1}{D} \sum_{i=1}^{D} r_i^j. \tag{10}$$

It is implied that there is a statistically significant difference in the average ranks of EER among the evaluated methods if the null hypothesis of Friedman's test is rejected (at $\alpha = 0.05$). The pairwise comparison of the methods across various scenarios was then examined using the Nemenyi post-hoc test [80]. The test asserts that if the average ranks of two or more methods differ by at least the critical difference (CD), their performances will differ significantly. Equation 11 shows how the CD is calculated:

$$CD = q_{\alpha,\infty,K} = \sqrt{\frac{K(K+1)}{6D}}, \tag{11}$$

where $q_{\alpha,\infty,K}$ is the value based on the Studentized range statistic. We displayed the comparison results using a CD diagram [81, 82] to visualise the ranking of the performance of the evaluated classification methods. The diagram also displays the critical difference between each method to show its significant difference.

We also carried out a pairwise comparison between our proposed DESTOUCH and static classification methods (i.e. single classifiers and static ensemble methods) using Wilcoxon Signed Rank Test [83] and Sign Test [84]. We are more interested in this comparison because most work in touch-based CA concentrates on classification methods using single classifiers and static ensemble methods. The Wilcoxon Signed Rank Test statistic is given as $T = min(R+, R-)$, where $R+$ is the total of the positive rankings and $R-$ is the total of the negative rankings. The null hypothesis is rejected if the p-value is less than the significance level. The significance level for this test was $alpha = 0.05$.

Finally, we used the Sign Test, which compares the total number of wins, ties, and losses. If the total number of wins of a particular method plus half of its total number of ties is higher than or equal to a threshold value nc, it is considered statistically better. According to the null hypothesis, there exist variations between the two methods. The former method performs better than the latter, as the rejected null hypothesis demonstrates. The critical value (at $\alpha = 0.05$) in our analysis is 13 because there are 17 scenarios obtained from the four datasets [81]. The following section summarises our findings from the experiments described in this section.

5 RESULTS AND DISCUSSION

In this section, the experimental results of our proposed method are presented. First, we compare the performance of several DESTOUCH schemes based on their selection and aggregation approaches. Then, we evaluate the performance of the proposed method compared to other DS and static classification methods (i.e. single classifiers and static ensemble methods). We report the experimental results of our study based on four different touch-based datasets. There are various scenarios for each dataset [28]. The scenarios vary in various ways, as described in Appendix A.

5.1 Comparison of Selection and Combination Approaches

In this section, we present the results of the comparison of various DESTOUCH schemes (i.e. DESTOUCH-R and DESTOUCH-T), which differ in terms of selection and aggregation approaches. Table 3 and Table 4 present various schemes for DESTOUCH-R and DESTOUCH-T, respectively. For DESTOUCH-R, three selection percentage ρ were evaluated: 25%, 50% and 75%. The scheme will select N subset of the most competent classifiers C' from the pool of classifiers C based on these percentages. For each selection percentage ρ, DESTOUCH-R will aggregate the selected classifiers C' using either simple MV or weighted MV. On the other hand, for DESTOUCH-T, three selection thresholds τ were evaluated: 0.4, 0.5 and 0.6. The scheme will select the subset of most competent classifiers C' from the pool of classifiers C that achieve this minimum threshold. For each selection threshold τ, DESTOUCH-T will also aggregate the selected classifiers C' using either simple MV or weighted MV,

Table 3 : DESTOUCH schemes based on rank

Scheme	Selection percentage	Aggregation approach
DESTOUCH-R$_{\text{25-S}}$	25	Simple MV
DESTOUCH-R$_{\text{50-S}}$	50	Simple MV
DESTOUCH-R$_{\text{75-S}}$	75	Simple MV
DESTOUCH-R$_{\text{25-W}}$	25	Weighted MV
DESTOUCH-R$_{\text{50-W}}$	50	Weighted MV
DESTOUCH-R$_{\text{75-W}}$	75	Weighted MV

Table 4 : DESTOUCH schemes based on threshold

Scheme	Selection threshold	Aggregation approach
DESTOUCH-T$_{\text{0.4-S}}$	0.4	Simple MV
DESTOUCH-T$_{\text{0.5-S}}$	0.5	Simple MV
DESTOUCH-T$_{\text{0.6-S}}$	0.6	Simple MV
DESTOUCH-T$_{\text{0.4-W}}$	0.4	Weighted MV
DESTOUCH-T$_{\text{0.5-W}}$	0.5	Weighted MV
DESTOUCH-T$_{\text{0.6-W}}$	0.6	Weighted MV

We evaluated each DESTOUCH scheme over 17 scenarios combined from four touch-based biometric datasets (with multiple users per scenario). Friedman rank test [79] was carried out to evaluate the 12 DESTOUCH schemes. We then calculated the Average Rank (AR) across the 17 scenarios. The best scheme is the one that obtains the lowest AR. Table 5 shows the performance of various DESTOUCH schemes. Based on this result, Friedman test shows that the $p = 3.96 \times e^{-11}$. This result indicates that there is a significant difference between the DESTOUCH schemes. We then carried out the Nemenyi post-hoc test [80]. Figure 2 shows the CD diagram, where the higher the average rank (lower in the value), the better the scheme. It is worth noting that the significantly different schemes have a difference in ranking higher than the CD value of $CD = 4.04$. Therefore, the line that connects the two methods indicates that the methods are not significantly different (less than the CD value).

Table 5 : Performance of different DESTOUCH schemes on all datasets according to EER (%)

	Frank						Serwadda								Antal		Mahbub	
	FRK_1	FRK_2	FRK_3	FRK_4	FRK_5	FRK_6	SWD_1	SWD_2	SWD_3	SWD_4	SWD_5	SWD_6	SWD_7	SWD_8	ANT_1	ANT_2	MHB	AR
DESTOUCH-R																		
DESTOUCH-$R_{25\text{-}S}$	1.04	0.91	14.60	3.28	11.92	12.22	2.97	2.10	22.66	14.42	3.25	1.47	20.84	9.70	2.76	5.49	24.25	11.62
DESTOUCH-$R_{50\text{-}S}$	0.97	0.55	13.28	2.76	11.51	11.00	2.57	1.69	21.98	**13.52**	**2.97**	1.20	19.17	8.79	2.22	**4.59**	23.21	5.79
DESTOUCH-$R_{75\text{-}S}$	1.03	**0.51**	**13.04**	2.74	11.29	10.44	**2.54**	1.64	21.34	13.55	3.00	1.17	18.53	8.56	2.50	4.80	22.82	4.91
DESTOUCH-$R_{25\text{-}W}$	1.00	0.80	13.73	2.96	11.24	10.31	2.72	1.71	21.70	14.03	3.04	1.27	19.57	8.64	2.35	4.98	23.71	8.38
DESTOUCH-$R_{50\text{-}W}$	**0.93**	0.65	**13.04**	**2.61**	10.90	**10.01**	2.56	**1.59**	20.57	13.72	3.08	**1.16**	19.19	8.27	**2.17**	**4.59**	22.92	**3.71**
DESTOUCH-$R_{75\text{-}W}$	1.01	0.69	13.28	2.69	**10.85**	10.16	2.64	1.64	20.44	13.69	3.13	1.18	18.86	8.24	2.40	4.80	22.80	4.85
DESTOUCH-T																		
DESTOUCH-$T_{0.4\text{-}S}$	1.00	0.66	13.46	2.94	11.98	10.46	2.76	1.81	19.39	13.72	3.30	1.33	18.75	8.35	3.40	5.17	22.70	8.21
DESTOUCH-$T_{0.5\text{-}S}$	0.98	0.58	13.49	2.92	11.32	10.43	2.67	1.78	19.71	13.69	3.13	1.26	18.83	8.34	3.01	4.98	**22.65**	6.65
DESTOUCH-$T_{0.6\text{-}S}$	0.96	0.55	13.29	2.74	11.05	10.70	**2.54**	1.68	19.83	13.58	3.00	**1.16**	18.92	8.42	2.43	4.68	23.06	4.85
DESTOUCH-$T_{0.4\text{-}W}$	0.99	0.89	13.57	2.92	11.14	10.44	2.85	1.80	**19.08**	13.89	3.41	1.38	18.55	8.00	2.76	5.13	22.90	7.97
DESTOUCH-$T_{0.5\text{-}W}$	0.98	0.81	13.58	2.91	11.10	10.36	2.78	1.73	19.33	13.82	3.31	1.33	**18.48**	**7.98**	2.37	4.98	22.81	6.59
DESTOUCH-$T_{0.6\text{-}W}$	0.96	0.76	13.29	2.78	10.94	10.23	2.65	1.63	19.32	13.68	3.17	1.23	18.76	8.04	2.21	4.71	22.89	4.47

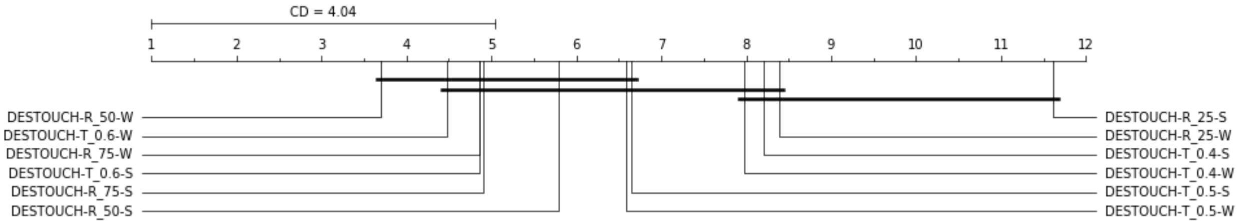


Figure 2 : Average rank of various DESTOUCH schemes. The higher the rank, the better the scheme.

From Figure 2, we can see that for DESTOUCH-R, the scheme with selection percentage $\rho = 50\%$ and weighted MV as the aggregation approach (DESTOUCH-$R_{50\text{-}W}$) performed the best in terms of the average rank. This scheme outperformed the scheme based on simple MV (DESTOUCH-$R_{50\text{-}S}$). In general, the DESTOUCH schemes that use weighted MV as the aggregation approach performed better than those that use simple MV. We can also see that DESTOUCH-$T_{0.6\text{-}W}$ is better than DESTOUCH-$T_{0.6\text{-}S}$, DESTOUCH-$R_{75\text{-}W}$ performs better than DESTOUCH-$R_{75\text{-}S}$, and so forth. Based on this observation, we can conclude that aggregation approaches have an influence on a particular scheme.

Besides, we can observe that for each aggregation approach of DESTOUCH-R, a larger selection percentage ρ generally outperformed the lower selection percentage. We believe a larger number of selected classifiers could produce better results. Except for DESTOUCH-$R_{50\text{-}W}$, this scheme is better than DESTOUCH-$R_{75\text{-}W}$. For each aggregation approach of DESTOUCH-T, we can see that the larger the selection threshold τ, the better the performance of the scheme. Therefore, having a more restricted threshold can produce better performance. Finally, by comparing the best scheme for both DESTOUCH-R and DESTOUCH-T, we can see that DESTOUCH-$R_{50\text{-}W}$ is better than DESTOUCH-$T_{0.6\text{-}W}$.

In the next section, we will compare the performance of our proposed DESTOUCH-R and

DESTOUCH-T with some other DS methods found in the literature. Since DESTOUCH-$\mathrm{R}_{50\text{-W}}$ and DESTOUCH-$\mathrm{T}_{0.6\text{-W}}$, are the best scheme for DESTOUCH-R and DESTOUCH-T, respectively, we will compare the results of these two schemes with other methods in the rest parts of this section. Therefore, we will only refer to these two schemes as simply DESTOUCH-R and DESTOUCH-T, respectively, hereafter.

5.2 Comparison with Other DS Methods

In this section, we compare the performance of our proposed DS methods (DESTOUCH-R and DESTOUCH-T) with some other DS methods presented in the literature (originally designed meant for other domains). This comparison investigates the performance and effectiveness of the proposed DESTOUCH based on the datasets used in our study. Table 6 shows the nine benchmark DS methods, which include DCS-OLA [59], DCS-LCA [59], DCS-Priori [63], DCS-Posteriori [63]), (DES-KNORAE [67], DES-KNORAu [67], DES-RRC [58], DES-P [26], and META-DES [68]). For these benchmark DS methods, we also generated a pool of classifiers that consists of SVM, NB, DT, KNN, LR, and NN. We also set the size of the region of competence θ_j as $K = 7$. We would like to note that DES-RRC does not use the KNN to define the region of competence. It originally defines the region of competence using the whole validation set D_{val} based on a potential function model. However, we set the size of the validation samples to be 7 to reduce the computational time if the whole validation set D_{val} is used.

Table 6 : DS methods considered as the benchmark methods in the experiments. The methods differ terms of the region of competence definition, selection criteria, and selection approach.

Method	Abbreviation	Region of competence definition	Selection criteria	Selection approach
Overall Local Accuracy [59]	DCS-OLA	KNN	Accuracy	DCS
Local class accuracy [59]	DCS-LCA	KNN	Accuracy	DCS
A Priori [63]	DCS-Priori	KNN	Probabilistic	DCS
A Posteriori [63]	DCS-Posteriori	KNN	Probabilistic	DCS
K-Nearest Oracles Eliminate [67]	DES-KNORAE	KNN	Oracle	DES
K-Nearest Oracles Union [67]	DES-KNORAU	KNN	Oracle	DES
DES Randomized Reference Classifier [58]	DES-RRC	Potential function	Probabilistic	DES
DES Performance [26]	DES-P	KNN	Accuracy	DES
META-DES [68]	META-DES	KNN	Meta-Learning	DES

Table 7 shows the performance of each method. Our proposed methods, DESTOUCH-R and DESTOUCH-T, performed the best in 12 out of 17 scenarios combined from the four datasets (where DESTOUCH-R is the best performer in 9 scenarios, and DESTOUCH-T in 3 scenarios). Next, we analysed the overall performance of the DS methods (in terms of ranking) using the Friedman test [79]. Since the p-value $p = 4.433 \times e^{-26}$ is lower than the significance level $\alpha = 0.05$, we can reject the null hypothesis H_0 that all these DS methods are the same. We further analyse the results by carrying out the Nemenyi post-hoc test [80] to analyse the pairwise comparison of the DS methods. Figure 3 shows a CD diagram with the results of the Nemenyi post-hoc test, where $CD = 3.66$. We can see that our proposed methods have the highest average rank of $AR = 2.03$ and $AR = 2.47$ for DESTOUCH-R and DESTOUCH-T, respectively, across 17 scenarios combined from the four datasets. These results show the advantage of our proposed method. We will also compare these methods with other static classification methods in the next section.

Table 7 : Performance of DESTOUCH methods compared to other DS methods on all datasets according to EER (%)

	Frank						Serwadda								Antal		Mahbub	
	FRK_1	FRK_2	FRK_3	FRK_4	FRK_5	FRK_6	SWD_1	SWD_2	SWD_3	SWD_4	SWD_5	SWD_6	SWD_7	SWD_8	ANT_1	ANT_2	MHB	AR
Benchmark DS method																		
DCS-OLA	1.81	2.57	16.16	4.77	12.41	14.18	4.26	3.49	24.85	16.32	4.80	2.32	22.10	11.49	3.43	7.09	28.42	9.53
DCS-LCA	3.58	3.98	16.35	6.13	16.73	13.79	6.00	5.36	23.52	16.88	7.04	3.17	20.82	10.90	5.11	10.16	29.85	10.59
DCS-Priori	1.41	1.68	13.95	4.60	12.18	14.53	4.22	3.25	23.31	16.36	4.24	2.13	20.19	10.60	3.82	6.01	27.13	8.29
DCS-Posteriori	1.82	2.51	14.15	5.50	13.75	15.10	4.35	4.06	22.65	16.57	4.64	2.46	20.39	10.88	6.48	7.07	27.86	9.53
DES-KNORAE	1.16	0.92	13.28	**2.32**	11.55	12.11	2.76	2.05	20.69	14.09	**2.96**	1.35	19.66	9.22	2.50	4.65	25.06	5.03
DES-KNORAU	1.16	0.68	13.66	2.91	11.77	12.19	2.83	2.01	22.76	14.07	3.33	1.39	19.49	9.24	3.24	5.48	23.31	6.47
DES-RRC	0.97	**0.55**	13.48	2.87	11.24	10.53	2.67	1.78	19.78	13.68	3.23	1.24	18.83	8.33	3.01	4.97	**22.63**	3.56
DES-P	0.96	0.58	13.52	2.84	11.48	10.56	2.69	1.78	19.51	**13.65**	3.26	1.25	18.87	8.27	2.98	4.96	22.76	3.62
META-DES	1.03	0.81	13.34	2.72	12.04	11.28	2.74	1.95	20.78	14.07	3.13	1.24	19.22	8.52	2.31	5.02	24.62	4.88
DESTOUCH method																		
DESTOUCH-R	**0.93**	0.65	**13.04**	2.61	**10.90**	**10.01**	**2.56**	**1.59**	20.57	13.72	3.08	**1.16**	19.19	8.27	**2.17**	**4.59**	22.92	**2.03**
DESTOUCH-T	0.96	0.76	13.29	2.78	10.94	10.23	2.65	1.63	**19.32**	13.68	3.17	1.23	**18.76**	**8.04**	2.21	4.71	22.89	2.47

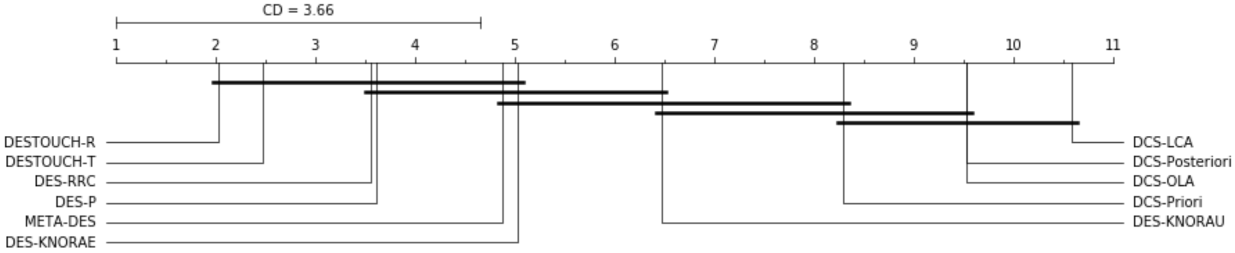


Figure 3 : Average rank of DESTOUCH-R, DESTOUCH-T and other DS methods. The higher the rank, the better the method.

5.3 Comparison with Other Classification Methods

In this section, we compare the performance of our proposed methods (DESTOUCH-R and DESTOUCH-T) with other types of classification methods chosen in this study. The first group of methods are six single classifiers that form the pool of classifiers. The single classifiers include SVM, NB, DT, KNN, LR, and NN. Second, we employed static ensemble methods, which include Random Forest (RF), Majority Voting (MV), Static Selection (SS), and Single Best (SB). Table 8 shows the benchmark methods under the category of static classification methods (i.e. single classifiers and static ensemble methods). RF is a homogeneous ensemble classifier consisting of multiple decision trees (in our case, 100 trees). On the other hand, MV, SS and, SB are heterogeneous ensemble classifiers that consist of the six single classifiers (SVM, NB, DT, KNN, LR and NN). This analysis investigates whether the proposed methods can achieve a better result compared to static classification methods found in the literature. This analysis is crucial since most studies in the domain of touch-based CA are focussing on classification methods using single classifiers and static ensemble methods (e.g.SVM KNN, RF, and etc.). The experimental results are presented in Table 9.

Table 9 shows that our proposed DESTOUCH-R and DESTOUCH-T are the best methods in 7 out of 17 scenarios (where DESTOUCH-R is the best performer in 5 scenarios, and DESTOUCH-T is the best performer in 2 scenarios). We also carried out the Friedman test [79] amongst these methods. The p-value $p = 1.509 \times e^{-28}$ of the test is lower than the significance level $\alpha = 0.05$

Table 8 : Static classification methods (single classifiers and static ensemble methods) considered as the benchmark methods in the experiments

Type	Method	Abbreviation	Ensemble
Single classifiers	Support Vector Machine [46]	SVM	-
	Naive Bayes [49]	NB	-
	Decision Tree [48]	DT	-
	K-Nearest Neighbour [45]	KNN	-
	Logistic Regression [51]	LR	-
	Neural Network [49]	NN	-
Static ensemble	Random Forest [85]	RF	Homogeneous
	Majority Voting [71]	MV	Heterogeneous
	Static Selection [72]	SS	Heterogeneous
	Single Best Classifier [72]	SB	Heterogeneous

and hence, we can reject the null hypothesis H_0 that these methods are the same. Figure 4 shows the CD diagram ($CD = 4.04$) with the results of the Nemenyi post-hoc test. We can observe that our proposed DESTOUCH-R and DESTOUCH-T have the highest average rank (2.21 and 2.65, respectively) across 17 scenarios combined the four datasets. These results show the superiority of our methods compared to not only single classifiers but also static ensemble methods. This better performance compared to single individual classifiers can be explained by the ability of the methods to determine experts in different regions of the feature space (defined by the region of competence). As in DESTOUCH, only the most competent classifiers for a particular test sample are selected based on the proposed measure of competence to determine to whom a particular touch stroke belongs. Therefore, the incompetent classifiers that are not the experts in the local region have no contribution towards the classification decision.

Table 9 : Performance of DESTOUCH methods compared to other classification methods on all datasets according to EER (%)

	Frank						Serwadda								Antal		Mahbub	
	FRK$_1$	FRK$_2$	FRK$_3$	FRK$_4$	FRK$_5$	FRK$_6$	SWD$_1$	SWD$_2$	SWD$_3$	SWD$_4$	SWD$_5$	SWD$_6$	SWD$_7$	SWD$_8$	ANT$_1$	ANT$_2$	MHB	AR
Single classifier																		
SVM	5.75	4.51	18.19	7.08	15.09	14.47	6.82	8.65	22.83	19.97	8.82	4.04	22.66	11.80	5.40	11.95	35.90	9.56
NB	9.86	10.08	22.42	14.56	24.45	22.70	15.86	8.76	31.39	19.06	14.81	6.19	24.53	13.17	10.42	13.96	37.89	11.41
DT	2.90	3.86	17.56	7.91	17.53	16.80	6.52	4.91	30.21	19.07	6.22	3.83	25.77	13.81	5.43	10.02	28.00	9.71
KNN	1.16	0.94	14.83	3.59	**10.08**	14.32	3.54	2.80	24.54	16.01	4.39	1.39	20.57	11.36	2.75	6.75	27.84	5.94
LR	4.75	5.35	18.11	7.43	18.59	14.47	11.77	9.69	24.24	21.70	14.00	5.30	20.91	13.45	11.34	17.67	34.45	10.32
NN	1.51	1.51	14.72	5.34	14.00	13.48	4.72	4.39	25.71	18.45	4.66	2.14	21.24	11.20	3.90	7.78	28.49	7.47
Static ensemble method																		
RF	**0.93**	0.84	13.38	3.47	11.03	**9.58**	2.81	1.82	22.25	14.05	**2.90**	1.77	**18.11**	8.10	**2.16**	4.73	**19.33**	3.15
MV	1.04	0.75	13.79	3.14	10.97	10.86	2.95	1.91	20.90	**13.47**	3.34	1.24	18.13	8.39	3.42	5.39	22.75	3.88
SS	1.14	**0.53**	14.49	**2.57**	12.24	11.69	2.59	1.72	21.18	13.82	2.99	**1.14**	19.18	9.42	2.23	5.05	22.99	3.59
SB	4.69	2.38	19.16	5.76	13.41	14.20	4.63	4.32	28.13	17.81	5.82	2.40	23.57	13.58	3.31	7.66	27.86	8.12
DESTOUCH method																		
DESTOUCH-R	**0.93**	0.65	**13.04**	2.61	10.90	10.01	**2.56**	**1.59**	20.57	13.72	3.08	1.16	19.19	8.27	2.17	**4.59**	22.92	**2.21**
DESTOUCH-T	0.96	0.76	13.29	2.78	10.94	10.23	2.65	1.63	**19.32**	13.68	3.17	1.23	18.76	**8.04**	2.21	4.71	22.89	2.65

Furthermore, we performed a pairwise comparison between each DESTOUCH methods with the static classifiers (single classifiers and static ensemble methods) involved in this section. We carried out two additional statistical analysis: Wilcoxon Signed Rank Test [83] and Sign Test [84]. Table 10 shows the results of Wilcoxon Signed Rank Test. The pairwise comparison tests the null hypothesis

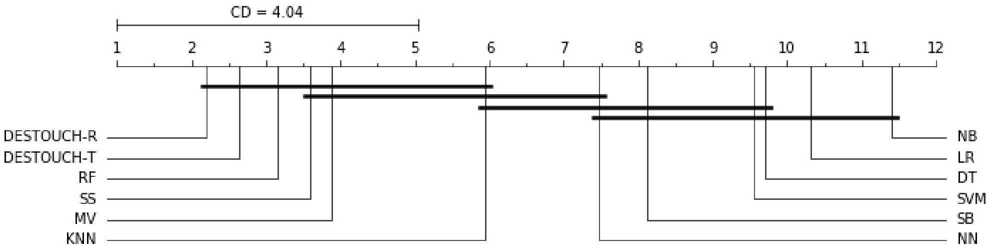


Figure 4 : Average rank of DESTOUCH-R, DESTOUCH-T and other types of classification methods, The higher the rank, the better the method.

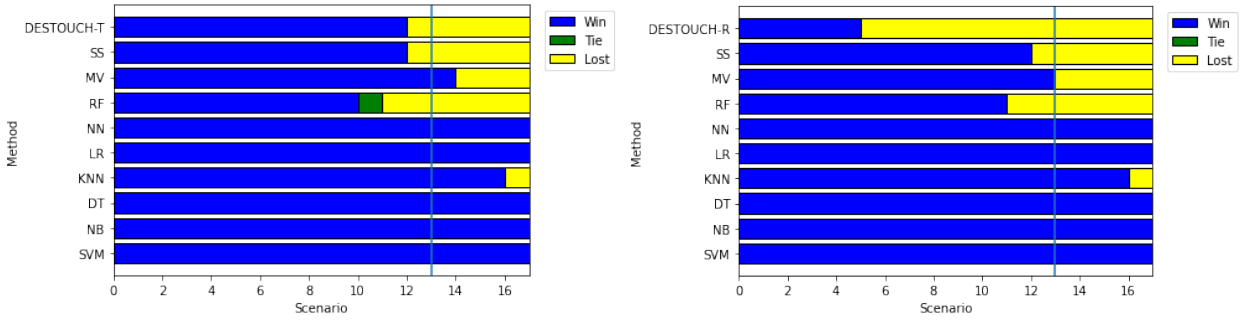
where two classification methods performed equally. The null hypothesis H_0 is rejected if the p-value is less than the significance level $\alpha = 0.05$. Based on the results, we can see DESTOUCH-R and DESTOUCH-T are significantly better than the static classifiers, except RF.

Table 10

Method	p-value	Null Hypothesis
DESTOUCH-R vs SVM	$1.526 \times e^{-05}$	Reject
DESTOUCH-R vs NB	$1.526 \times e^{-05}$	Reject
DESTOUCH-R vs DT	$1.526 \times e^{-05}$	Reject
DESTOUCH-R vs KNN	0.000	Reject
DESTOUCH-R vs LR	$1.526 \times e^{-05}$	Reject
DESTOUCH-R vs NN	$1.526 \times e^{-05}$	Reject
DESTOUCH-R vs RF	0.379	Fail to reject
DESTOUCH-R vs SS	0.009	Reject
DESTOUCH-R vs MV	0.023	Reject
DESTOUCH-R vs SB	$1.526 \times e^{-05}$	Reject
DESTOUCH-R vs DESTOUCH-T	0.329	Fail to reject
DESTOUCH-T vs SVM	$1.526 \times e^{-05}$	Reject
DESTOUCH-T vs NB	$1.526 \times e^{-05}$	Reject
DESTOUCH-T vs DT	$1.526 \times e^{-05}$	Reject
DESTOUCH-T vs KNN	0.000	Reject
DESTOUCH-T vs LR	$1.526 \times e^{-05}$	Reject
DESTOUCH-T vs NN	$1.526 \times e - 05$	Reject
DESTOUCH-T vs RF	0.431	Fail to reject
DESTOUCH-T vs MV	0.017	Reject
DESTOUCH-T vs SS	0.031	Reject
DESTOUCH-T vs SB	$1.526 \times e^{-05}$	Reject

Figure 5a and 5b respectively show the performance of DESTOUCH-R and DESTOUCH-T compared with single classifiers and static ensemble methods in terms of wins, ties and losses (Sign Test) over the 17 scenarios combined from the four datasets. The vertical line each line illustrates the critical values $n_c = 13$ considering significance levels of $\alpha = 0.05$. From Figure 5a and 5b , we can see that

DESTOUCH-R and DESTOUCH-T performed significantly better than all methods, except RF and SS. It is also worth noting that DESTOUCH-R and DESTOUCH-T do not have any significant deference according to Wilcoxon Signed Rank Test and Sign Test.



(a) DESTOUCH-R

(b) DESTOUCH-T

Figure 5 : Performance of (a) DESTOUCH-R and (b) DESTOUCH-T compared with single classifiers and static ensemble methods in terms of wins, ties and losses based on the EER over the 17 scenarios combined from four datasets. The vertical line represents the critical values $n_c = 13$ at significance levels of $\alpha = 0.05$

Table 11 : The best classification method in each scenario according to EER (%)

Dataset	Scenario	Method	Category	EER
Frank	FRK_1	RF	Static ensemble	0.93
	FRK_2	SS	Static ensemble	0.53
	FRK_3	DESTOUCH-R	DS	13.04
	FRK_4	SS	Static ensemble	2.57
	FRK_5	KNN	Single classifier	10.08
	FRK_6	RF	Static ensemble	9.58
Serwadda	SWD_1	DESTOUCH-R	DS	2.56
	SWD_2	DESTOUCH-R	DS	1.59
	SWD_3	DESTOUCH-T	DS	19.32
	SWD_4	MV	Static ensemble	13.47
	SWD_5	RF	Static ensemble	2.90
	SWD_6	SS	Static ensemble	1.14
	SWD_7	RF	Static ensemble	18.11
	SWD_8	DESTOUCH-T	DS	8.04
Antal	ANT_1	RF	Static ensemble	2.16
	ANT_2	DESTOUCH-R	DS	4.59
Mahbub	MHB	RF	Static ensemble	19.33

We further analyse the best method for each scenario on all datasets. Table 11 shows the best method for each scenario. We can see that RF was the best method in 6 out of 17 scenarios. However, it is not the top-ranked method according to the average rank. We believe that this is due to the inconsistency of RF. Figure 6 shows the box plot for DESTOUCH-R, DESTOUCH-T, RF, and SS across all scenarios. From the figure, we can see that our proposed methods are more consistent compared to RF and SS. We argue that a more consistent classification method is preferable for a CA scheme as an inconsistent classification method may greatly affect the security and usability of the scheme. Therefore, the experimental results in this section demonstrated the potential and feasibility of the proposed method, showing that it can improve the authentication performance of touch-based CA with a relatively low EER in many scenarios across multiple datasets, exhibiting relatively high consistency.

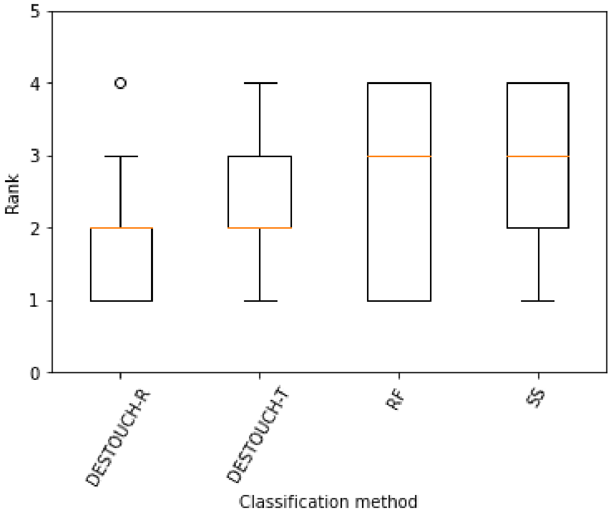


Figure 6 : Box plot of the average rank of DESTOUCH-R, DESTOUCH-T and RF

6 CONCLUSIONS AND FUTURE WORK

In this study, we proposed a Dynamic Selection (DS) method for touch-based continuous authentication (CA). We introduced DESTOUCH, a DS method based on a probabilistic measure of competence. We generated a pool of six heterogeneous classifiers: K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Logistic Regression (LR), and Multi-layer Perceptron Neural Network (NN). Our method estimated the competence level of each classifier by computing the probability of correct classification for legitimate and illegitimate user samples in the test sample's region of competence. The method divided the samples in the region of competence into two subsets: legitimate and illegitimate. It calculates the probabilities representing true acceptance for legitimate users and true rejection for illegitimate users. These probabilities are then integrated to define each classifier's measure of competence. We presented two variants: DESTOUCH-R and DESTOUCH-T. DESTOUCH-R selects classifiers based on competence ranking, while DESTOUCH-T selects them based on a competence threshold. Each variant uses either simple or weighted majority voting (MV) for aggregation.

We conducted experiments using four touch-based biometric datasets (Frank, Serwadda, Antal, and Mahbub). We compared our method with six single classifiers (KNN, SVM, DT, NNB, LR, and NN), four static ensemble methods, and nine other DS methods. Based on the average rank in Equal Error Rate (EER), the proposed method ranked the highest compared to other DS and static classification methods (single classifiers and static ensemble methods). The experimental results also show that the performance of our proposed method is more consistent compared to some of the other top-performing methods. When the authentication error can be decreased even further, the results can provide a better security mechanism for mobile devices by assuring a higher level of security without compromising usability. In other words, a mobile device is able to detect and block illegitimate users more effectively. In addition, mobile device authentication can prevent legitimate users from being locked out during their usual usage session, which is more convenient for the user. Therefore, due to its superior classification capability, we recommend using DESTOUCH for user classification in touch-based CA.

It is worth noting that DESTOUCH uses the traditional KNN algorithm to define the competence region of a test sample. Future work could explore other methods to improve this process. Implementing our methods on mobile devices for real-time authentication is another potential direction. Additionally, using larger datasets to test the long-term validity of our methods is an interesting area for future research.

ACKNOWLEDGEMENT

We thank Monash University Malaysia for proving the Unified Research Computing to carry out the experiments in this study.

REFERENCES

[1] F. Tari, A. A. Ozok, and S. H. S. Holden, “A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords,” in *ACM International Conference Proceeding Series*, vol. 149. New York, New York, USA: ACM Press, 2006, p. 56. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1143120.1143128

[2] N. L. Clarke and S. M. Furnell, “Authentication of users on mobile telephones - A survey of attitudes and practices,” *Computers and Security*, vol. 24, no. 7, pp. 519–527, 2005.

[3] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, “Smudge Attacks on Smartphone Touch Screens,” in *Proceeding in WOOT’10 Proceedings of the 4th USENIX conference on Offensive technologies*. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–7. [Online]. Available: https://www.usenix.org/legacy/event/woot10/tech/full{_}papers/Aviv.pdf

[4] W. Meng, D. D. S. Wong, S. Furnell, and J. Zhou, “Surveying the Development of Biometric User Authentication on Mobile Phones,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1268–1293, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7000543/

[5] V. M. V. V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, “Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges,” *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, jul 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7503170/

[6] A. Z. Zaidi, C. Y. Chong, Z. Jin, R. Parthiban, and A. S. Sadiq, “Touch-based continuous

mobile device authentication: State-of-the-art, challenges and opportunities," *Journal of Network and Computer Applications*, vol. 191, p. 103162, oct 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804521001740

[7] 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.

[8] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion, "Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 1–1, mar 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7335628/

[9] A. Serwadda, V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *IEEE 6th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2013*. IEEE, 2013.

[10] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, "Benchmarking Touchscreen Biometrics for Mobile Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2720–2733, nov 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8353868/

[11] W. Meng, Y. Y. Wang, D. S. D. Wong, S. Wen, and Y. Xiang, "TouchWB: Touch behavioral user authentication based on web browsing on smartphones," *Journal of Network and Computer Applications*, vol. 117, pp. 1–9, sep 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804518301723

[12] Y. Yang, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, "BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics," *Ad Hoc Networks*, vol. 84, pp. 9–18, mar 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1570870518306899

[13] W. Meng, W. Li, and D. S. Wong, "Enhancing touch behavioral authentication via cost-based intelligent mechanism on smartphones," pp. 1–19, dec 2018. [Online]. Available: http://link.springer.com/10.1007/s11042-018-6094-2

[14] Z. Syed, J. Helmick, S. Banerjee, and B. Cukic, "Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability," *Journal of Systems and Software*, vol. 149, pp. 158–173, mar 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0164121218302516

[15] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," *NDSS - Network and Distributed System Security Symposium*, pp. 1–16, 2013. [Online]. Available: https://optimization.asu.edu/papers/XUE-CNF-2013-NDSS.pdfhttp://internetsociety.org/doc/unobservable-re-authentication-smartphones

[16] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, "Active user authentication for

smartphones: A challenge data set and benchmark results," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016, pp. 1–8.

[17] Y. S. Lee, W. Hetchily, J. Shelton, D. Gunn, K. Roy, A. Esterline, and X. Yuan, "Touch based active user authentication using deep belief networks and random forests," in *Proceedings of the 6th International Conference on Information Communication and Management, ICICM 2016*, 2016.

[18] I. Chang, C. Y. Low, S. Choi, and A. Beng Jin Teoh, "Kernel Deep Regression Network for Touch-Stroke Dynamics Authentication," *IEEE Signal Processing Letters*, pp. 1–1, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8378259/

[19] D. J. Gunn, Z. Liu, R. Dave, X. Yuan, and K. Roy, "Touch-Based Active Cloud Authentication Using Traditional Machine Learning and LSTM on a Distributed Tensorflow Framework," *International Journal of Computational Intelligence and Applications*, vol. 18, no. 04, p. 1950022, dec 2019. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S1469026819500226

[20] M. Montgomery, P. Chatterjee, J. Jenkins, and K. Roy, "Touch Analysis: An Empirical Evaluation of Machine Learning Classification Algorithms on Touch Data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Springer Verlag, jul 2019, vol. 11611 LNCS, pp. 147–156. [Online]. Available: http://link.springer.com/10.1007/978-3-030-24907-6{_}12

[21] H. C. Volaka, G. Alptekin, O. E. Basar, M. Isbilen, and O. D. Incel, "Towards Continuous Authentication on Mobile Phones using Deep Learning Models," *Procedia Computer Science*, vol. 155, pp. 177–184, jan 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S187705091930941X

[22] S. Choi, I. Chang, and A. B. J. Teoh, "One-class Random Maxout Probabilistic Network for Mobile Touchstroke Authentication," in *2018 24th International Conference on Pattern Recognition (ICPR).* IEEE, aug 2018, pp. 3359–3364. [Online]. Available: https://ieeexplore.ieee.org/document/8545451/

[23] S. Y. Ooi and A. B.-J. Teoh, "Touch-Stroke Dynamics Authentication Using Temporal Regression Forest," *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 1001–1005, jul 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8713391/

[24] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, apr 1997. [Online]. Available: http://ieeexplore.ieee.org/document/585893/

[25] S. Lessmann, B. Baesens, H. V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, nov 2015. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0377221715004208

[26] T. Woloszynski, P. Podsiadlo, G. Stachowiak, and M. Kurzynski, “A dissimilarity-based multiple classifier system for trabecular bone texture in detection and prediction of progression of knee osteoarthritis,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 226, no. 11, pp. 887–894, nov 2012. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0954411912456650

[27] V. Sharma and R. Enbody, “User authentication and identification from user interface interactions on touch-enabled devices,” in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks - WiSec ’17*, 2017, pp. 1–11. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3098243.3098262

[28] A. Z. Zaidi, C. Y. Chong, R. Parthiban, and A. S. Sadiq, “A framework of dynamic selection method for user classification in touch-based continuous mobile device authentication,” *Journal of Information Security and Applications*, vol. 67, p. 103217, jun 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214212622000928

[29] P. C. P. Smits, “Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 801–813, apr 2002. [Online]. Available: http://ieeexplore.ieee.org/document/1006354/

[30] J. Xiao, L. Xie, C. He, and X. Jiang, “Dynamic classifier ensemble model for customer classification with imbalanced class distribution,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 3668–3675, feb 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417411013686

[31] S. Bashbaghi, E. Granger, R. Sabourin, and G.-A. Bilodeau, “Dynamic ensembles of exemplar-SVMs for still-to-video face recognition,” *Pattern Recognition*, vol. 69, pp. 61–81, 2017. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019480409{&}doi=10.1016{%}2Fj.patcog.2017.04.014{&}partnerID=40{&}md5=b14b38aa7f50c50777380d59f8358167

[32] X. Feng, Z. Xiao, B. Zhong, J. Qiu, and Y. Dong, “Dynamic ensemble classification for credit scoring using soft probability,” *Applied Soft Computing*, vol. 65, pp. 139–151, apr 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494618300279

[33] P. Porwik, R. Doroz, and K. Wrobel, “An ensemble learning approach to lip-based biometric verification, with a dynamic selection of classifiers,” *Expert Systems with Applications*, vol. 115, pp. 673–683, jan 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418305529?via{%}3Dihub

[34] P. Porwik, R. Doroz, and T. E. Wesolowski, “Dynamic keystroke pattern analysis and classifiers with competence for user recognition,” *Applied Soft Computing*, p. 106902, nov 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494620308401

[35] L. Batista, E. Granger, and R. Sabourin, “Dynamic selection of generative-discriminative ensembles for off-line signature verification,” *Pattern Recognition*, vol. 45, no. 4, pp.

1326–1340, apr 2012. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0031320311004353www.elsevier.com/locate/pr

[36] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "Classifier combination for hand-printed digit recognition," in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93).* IEEE Comput. Soc. Press, 1993, pp. 163–166. [Online]. Available: http://ieeexplore.ieee.org/document/395758/

[37] P. C. P. Smits, "Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 801–813, apr 2002. [Online]. Available: http://ieeexplore.ieee.org/document/1006354/

[38] L. Melo Junior, F. Maria Nardini, C. Renso, R. Trani, and J. Antonio Macedo, "A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems," *Expert Systems with Applications*, p. 113351, mar 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417420301767

[39] Y. Xia, J. Zhao, L. He, Y. Li, and M. Niu, "A novel tree-based dynamic heterogeneous ensemble method for credit scoring," *Expert Systems with Applications*, vol. 159, p. 113615, nov 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417420304395

[40] B. Wang and Z. Mao, "Outlier detection based on a dynamic ensemble model: Applied to process monitoring," *Information Fusion*, vol. 51, pp. 244–258, nov 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253518303282?via{%}3Dihub

[41] M. Martinez-Diaz, J. Fierrez, R. P. Krish, and J. Galbally, "Mobile signature verification: Feature robustness and performance comparison," *IET Biometrics*, vol. 3, no. 4, pp. 267 – 277, 2014.

[42] O. D. Incel, S. Gunay, Y. Akan, Y. Barlas, O. E. Basar, G. I. Alptekin, and M. Isbilen, "DAKOTA: Sensor and Touch Screen-Based Continuous Authentication on a Mobile Banking Application," *IEEE Access*, vol. 9, pp. 38 943–38 960, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9367144/

[43] P. Aaby, M. V. Giuffrida, W. J. Buchanan, and Z. Tan, "An omnidirectional approach to touch-based continuous authentication," *Computers & Security*, vol. 128, p. 103146, may 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404823000561

[44] Z. Shen, S. Li, X. Zhao, and J. Zou, "IncreAuth: Incremental-Learning-Based Behavioral Biometric Authentication on Smartphones," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1589–1603, jan 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10164632/

[45] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21 – 27, 1967.

[46] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp.

273–297, 1995.

[47] V. Vapnik, *The nature of statistical learning theory.* New York: Springer science & business media, 2013.

[48] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," *Group*, vol. 37, no. 15, pp. 237–251, 1984.

[49] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification 2nd ed*, 2000.

[50] H. Zhang, "The optimality of Naive Bayes," in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*, 2004.

[51] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques.* Elsevier, nov 2011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/C20090197155

[52] L. Zhou, J. K. Burgoon, D. P. Twitchell, T. Qin, and J. F. Nunamaker, "A comparison of classification methods for predicting deception in computer-mediated communication," *Journal of Management Information Systems*, vol. 20, no. 4, pp. 139–166, 2004.

[53] M. Antal, Z. Bokor, and L. Z. Szabó, "Information revealed from scrolling interactions on mobile devices," *Pattern Recognition Letters*, vol. 56, pp. 7–13, apr 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865515000355?via{%}3Dihubhttps://linkinghub.elsevier.com/retrieve/pii/S0167865515000355

[54] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, no. 1, pp. 3–17, mar 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156625351300047X

[55] M. Ala'raj and M. F. Abbod, "A new hybrid ensemble credit scoring model based on classifiers consensus system approach," *Expert Systems with Applications*, vol. 64, pp. 36–55, dec 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417416303621

[56] J. A. S. Lustosa Filho, A. M. Canuto, and R. H. N. Santiago, "Investigating the impact of selection criteria in dynamic ensemble selection methods," *Expert Systems with Applications*, vol. 106, pp. 141–153, sep 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417418302264

[57] A. S. Britto, R. Sabourin, and L. E. S. Oliveira, "Dynamic selection of classifiers - A comprehensive review," *Pattern Recognition*, vol. 47, no. 11, pp. 3665–3680, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2014.05.003

[58] T. Woloszynski and M. Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2656–2668, oct 2011.

[59] K. Woods, W. Philip Kegelmeyer, and K. Bowyer, "Combination of multiple

classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, apr 1997. [Online]. Available: http://ieeexplore.ieee.org/document/588027/

[60] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195–216, may 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1566253517304074

[61] A. Nabiha and F. Nadir, "New dynamic ensemble of classifiers selection approach based on confusion matrix for Arabic handwritten recognition," in *Proceedings of 2012 International Conference on Multimedia Computing and Systems, ICMCS 2012*. IEEE, may 2012, pp. 308–313. [Online]. Available: http://ieeexplore.ieee.org/document/6320200/

[62] M. C. P. de Souto, R. G. F. Soares, A. Santana, and A. M. P. Canuto, "Empirical comparison of Dynamic Classifier Selection methods based on diversity and accuracy for building ensembles," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. Department of Informatics and Applied Mathematics, Fed. Univ. of Rio Grande do Norte, Natal, Brazil: IEEE, jun 2008, pp. 1480–1487. [Online]. Available: http://ieeexplore.ieee.org/document/4633992/

[63] G. Giacinto and F. Roli, "Methods for dynamic classifier selection," in *Proceedings - International Conference on Image Analysis and Processing, ICIAP 1999*. IEEE Comput. Soc, 1999, pp. 659–664. [Online]. Available: http://ieeexplore.ieee.org/document/797670/

[64] T. Woloszynski and M. Kurzynski, "On a New Measure of Classifier Competence Applied to the Design of Multiclassifier Systems," in *15th International Conference on Image Analysis and Processing - ICIAP 2009, Proceedings*, Chair of Systems and Computer Networks, Wroclaw University of Technology, Wyb. Wyspianskiego 27, Wroclaw 50-370, Poland, 2009, vol. 5716 LNCS, pp. 995–1004. [Online]. Available: http://link.springer.com/10.1007/978-3-642-04146-4{_}106

[65] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognition*, vol. 34, no. 9, pp. 1879–1881, sep 2001. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0031320300001503

[66] P. R. Cavalin, R. Sabourin, and C. Y. Suen, "Dynamic selection approaches for multiple classifier systems," *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 673–688, mar 2013. [Online]. Available: http://link.springer.com/10.1007/s00521-011-0737-9

[67] A. H. Ko, R. Sabourin, and A. S. Britto, "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, no. 5, pp. 1718–1731, 2008.

[68] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, and T. Ing Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognition*, vol. 48, no. 5, pp. 1925–1935, may 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320314004919

[69] A. Qadeer and U. Qamar, "A Dynamic Ensemble Selection Framework Using Dynamic Weighting Approach." Springer, Cham, sep 2020, pp. 330–339. [Online]. Available: http://link.springer.com/10.1007/978-3-030-29516-5{_}25

[70] D. V. R. Oliveira, G. D. C. Cavalcanti, and R. Sabourin, "Online pruning of base classifiers for Dynamic Ensemble Selection," *Pattern Recognition*, vol. 72, pp. 44–58, dec 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317302522?via{%}3Dihub

[71] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, mar 1998. [Online]. Available: http://ieeexplore.ieee.org/document/667881/

[72] L. I. Kuncheva, *Combining Pattern Classifiers.* Wiley, jul 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/0471660264

[73] R. M. O. Cruz, L. G. Hafemann, R. Sabourin, and G. D. C. Cavalcanti, "Deslib: A dynamic ensemble selection library in python," *Journal of Machine Learning Research*, vol. 21, no. 8, pp. 1–5, 2020. [Online]. Available: http://jmlr.org/papers/v21/18-144.html

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825—-2830, 2011.

[75] B. Wang and Z. Mao, "A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule," *Information Fusion*, vol. 63, pp. 30–40, nov 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1566253520302645

[76] T. T. Nguyen, A. V. Luong, M. T. Dang, A. W.-C. Liew, and J. McCall, "Ensemble Selection based on Classifier Prediction Confidence," *Pattern Recognition*, vol. 100, p. 107104, apr 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320319304054

[77] T. Feng, X. Zhao, N. DeSalvo, Z. Gao, X. Wang, and W. Shi, "Security after login: Identity change detection on smartphones using sensor fusion," in *2015 IEEE International Symposium on Technologies for Homeland Security (HST).* IEEE, apr 2015, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7225268/

[78] R. M. Cruz, D. V. Oliveira, G. D. Cavalcanti, and R. Sabourin, "FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection," *Pattern Recognition*, vol. 85, pp. 149–160, jan 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320318302760

[79] M. Friedman, "A Comparison of Alternative Tests of Significance for the Problem of m Rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, mar 1940. [Online]. Available: http://projecteuclid.org/euclid.aoms/1177731944

[80] P. Nemenyi, "Distribution-free multiple comparisons," *Biometrics*, vol. 18, no. 2, 1962.

[81] J. Demšar, “Statistical Comparisons of Classifiers over Multiple Data Sets,” Tech. Rep. 1, 2006. [Online]. Available: http://jmlr.org/papers/v7/demsar06a.html

[82] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, “Orange: Data mining toolbox in python,” *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013.

[83] F. Wilcoxon, *Biometrics Bulletin.*

[84] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman and Hall/CRC, aug 2003. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.1201/9781420036268/handbook-parametric-nonparametric-statistical-procedures-david-sheskin

[85] Leo Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://link.springer.com/10.1023/A:1010933404324

APPENDIX

A. DESCRIPTION OF DATASETS AND SCENARIOS

Four publicly available touch biometric datasets were used in this study. These datasets consist of swipe gesture data. These gestures are commonly used when users operate a mobile device and, therefore, are appropriate for touch-based CA [9]. These datasets differ in terms of the number of users (subjects), the number of sessions, the duration of data collection, the setting for data collection, and the number of features. We used the original features that the authors of the dataset had first presented to evaluate how the proposed method performed with various features. These datasets include:

- **Frank Dataset [7]:** It consists of swipe data from 41 subjects who were asked to read texts and compare images, producing vertical and horizontal strokes. Initially, there were three sessions for reading texts and two for comparing images on the same day. There were several-minute breaks between sessions on that particular day before moving on to the following session. Some subjects took part in a second phase of collecting data for the same tasks after at least a week, but there was only one session for each task. The authors presented 27 touch-based features.

- **Serwadda Dataset [9]:** It is made up of swipe data from 190 subjects for two sessions separated by at least one day. The subjects had to respond to multiple-choice questions during each session. In order to complete the activities, subjects had to swipe back and forth between the horizontal and vertical directions. All short strokes with four or fewer touchpoints were considered outliers and eliminated. The data of each stroke direction were collected from different screen orientations (landscape and portrait). For each swiping action, the authors presented 28 features.

- **Antal Dataset [53]:** It was collected from 71 subjects in four weeks. The subjects were asked to complete tasks that involved document reading and image browsing, which resulted in vertical and horizontal swipes, respectively. Each subject had to complete the tasks throughout several sessions. The authors presented 15 features for each swiping action.

- **Mahbub Dataset [16]:** A multi-modal dataset known as the University of Maryland Active Authentication Dataset 02 (UMDAA-02) was presented by Mahbub et al. [16]. It was originally intended as a multi-modal continuous authentication. However, in this study, we only used the data from the touchscreen sensor since touch-based CA is our primary interest. The data was provided by 48 subjects who had used smartphones for more than two months. Additionally, unlike all the other datasets, there are no pre-defined tasks during data collection. In contrast to the other datasets with many sessions, the Mahbub dataset has no predetermined length of time. Instead, the session started and stopped when the device was unlocked or locked. The authors presented a total of 24 features. In our study, swipes with fewer than five data points were excluded to reduce the occurrence of outliers.

Table 12 illustrates the scenarios for each dataset. The scenarios vary in various ways. The first element is stroke direction. It is either horizontally (scroll to the left or right) or vertically (scroll up or down). Second, there are two screen orientations in the Serwadda dataset: portrait and landscape. We followed the setup similar to the the original paper for each dataset by building a model independently for varied stroke directions and screen orientations.

Table 12 : List of scenarios across all datasets

Dataset	Scenario			Notation
	Stroke Direction	Screen Orientation	Session	
Frank	Vertical	-	Intra-session	FRK$_1$
	Horizontal	-	Intra-session	FRK$_2$
	Vertical	-	Inter-session	FRK$_3$
	Horizontal	-	Inter-session	FRK$_4$
	Vertical	-	Inter-week	FRK$_5$
	Horizontal	-	Inter-week	FRK$_6$
Serwadda	Vertical	Portrait	Intra-session	SWD$_1$
	Horizontal	Portrait	Intra-session	SWD$_2$
	Vertical	Portrait	Inter-session	SWD$_3$
	Horizontal	Portrait	Inter-session	SWD$_4$
	Vertical	Landscape	Intra-session	SWD$_5$
	Horizontal	Landscape	Intra-session	SWD$_6$
	Vertical	Landscape	Inter-session	SWD$_7$
	Horizontal	Landscape	Inter-session	SWD$_8$
Antal	Vertical	-	-	ANT$_1$
	Horizontal	-	-	ANT$_2$
Mahbub	Combined	-	-	MHB

In addition, for the datasets collected throughout multiple sessions, we conducted tests in two different settings: intra-session and inter-session. A session refers to when a user is instructed to begin using the device and when they are instructed to stop using it. The duration is dependent on the specific dataset. For the intra-session scenario, we trained and evaluated a model on the dataset from the same session using the data partitioning technique described in Section 4.3. For inter-session, the model was trained with one session and then evaluated with a another session. In other words, the classifier was trained using the data from the previous session, and the model was evaluated using the data from the subsequent session. This setting is intended to ensure that the test samples were generated after the training samples.

For the Frank dataset, there are multiple sessions on a single day and one session the following week. We conducted the experiments under intra-session, inter-session, and inter-week scenarios. We conducted both intra-session and inter-session experiments for the Serwadda dataset. It is important to note that for each dataset, the duration of each session and the time between sessions varies (see Table 2). Multiple sessions were not used to partition the Antal dataset. Therefore, we were unable to separate it into distinct sessions. For the Mahbub dataset, the period of each data collection was not predetermined. Instead, the author defined a session as unlocking and locking a device. Consequently, a session of one user is not identical to another. We did not do the inter-session experiment for this dataset because there is no defined timeframe for every user.

Therefore, all sessions were merged.

B. HYPERPARAMETERS AND OTHER SETTINGS OF BASE CLASSIFIERS

Hyper-parameters in the classification algorithms mentioned above may affect overall performance. In order to fine-tune the hyper-parameters for each classification algorithm, we used a grid selection method. The hyper-parameters for each classification algorithm are listed in Table 13 along with their corresponding values [28]. The hyper-parameters not listed in this table are set to the default value according to each algorithm implementation.

Table 13 : Summary of parameters and settings associated to each classifier

Model	Hyper-parameter	Value
SVM	Regularization parameter, C	[0.001, 0.01, 0.1, 1, 10, 25, 50, 100, 1000]
	Kernel	RBF
	Kernel coefficient, γ	1 / number of features
	Tolerance for stopping criterion	$1e^{-3}$
NB	-	-
DT	Maximum depth of the tree	[none, 5, 10, 15,20, 25, 30]
	Minimum number of samples to split	[2, 4, 6, 8, 10]
	Minimum number of samples at a leaf	[1, 2, 3, 4, 5]
KNN	Number of neighbours	[1,. . . ,10]
	Algorithm	KD tree
	Distance metric used for finding neighbours	Euclidean
LR	-	-
NN	Number of hidden layers	1
	Number of hidden nodes	50