

A framework of dynamic selection method for user classification in touch-based continuous mobile device authentication

Ahmad Zairi Zaidi^{a,*}, Chun Yong Chong^{a,*}, Rajendran Parthiban^b and Ali Safaa Sadiq^c

^a*School of Information Technology, Monash University Malaysia, 47500 Subang Jaya, Selangor, Malaysia*

^b*School of Engineering, Monash University Malaysia, , 47500 Subang Jaya, Selangor, Malaysia*

^c*School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton, WV1 1LY, United Kingdom*

ARTICLE INFO

Keywords:

Touch biometric
Mobile device security
Continuous authentication
Multiple classifier system
Dynamic classifier selection
Dynamic ensemble selection

ABSTRACT

Continuous authentication can provide a mechanism to continuously monitor mobile devices while a user is actively using it, after passing the initial-login authentication phase. Touch biometric is one of the promising modality to realise continuous authentication on mobile devices by distinguishing between the touch strokes performed by the legitimate and illegitimate users through classification algorithms. While the benefit of the scheme is promising, the effectiveness of different classification methods are not thoroughly understood. Little consideration has been given on the combination of multiple classifiers to perform continuous authentication. In this paper, we propose a novel classification framework for touch-based continuous mobile device authentication (CMDA), utilising dynamic selection of classifiers (DS). Instead of classifying all touch strokes using the same classifier, the proposed framework classifies each touch sample using the most promising classifier(s) from a pool of classifiers. Based on the proposed framework, we evaluated various DS methods in multiple scenarios across four touch datasets. The aim of this evaluation is to assess the feasibility of DS on touch-based CMDA. We then compared these DS methods with well-known single classifiers and static ensemble methods. The experimental results show the potential and feasibility of the DS methods to improve the authentication performance of touch-based CMDA against the benchmark methods. We found that DS methods are capable of producing promising results with relatively low equal error rate (EER) in many scenarios of the datasets, with relatively high consistencies. The obtained results would be valuable for further enhancement of existing user classification methods and the development of new DS methods in touch-based CMDA.

1. Introduction

Mobile devices have become a ubiquitous and primary computing device for most people in recent years. Mobile devices (especially smartphones) have become an inseparable part of our daily lives, not only for phone calling and text messaging, but also other personal and business purposes such as online shopping, banking, and data storage. Due to its portable size, the device can easily be stolen or lost, which will not only cost the users in terms of monetary loss, but also potential information leakage. Besides, as an Internet of Things (IoT) device, it also transmits data to and from other IoT systems such as smart watches [1], vehicle to grid (V2G) system [2], and cloud computing [3]. Therefore, some form of security mechanism is needed to protect the integrity of the device and to ensure the information stored on the device as well as the data transferred from and to the device are protected.

Traditional password-based authentication method, which includes PIN and swipe pattern codes are still commonly used. However, these authentication methods have several drawbacks such as an easily-guessed simple password [4],

shoulder surfing [5], and smudge attack [6]. Biometric authentication methods such as fingerprint and face recognition are common authentication methods nowadays to address the shortcomings of traditional password-based methods [7]. However, such biometric methods require specialised hardware such as front-facing camera and fingerprint scanner, which might increase the implementation cost [7]. On top of that, all the authentication methods mentioned above can only provide one-time initial-login authentication. In the case where an illegitimate user can bypass the initial-login authentication or the legitimate user left the device without setting it to be automatically locked, the device may not be able to detect illegitimate access any longer. In order to complement these authentication methods, several studies are focussing on continuous authentication methods for mobile devices [8]. It is an authentication scheme that can monitor the device usage continuously after the one-time authentication. There is a growing number of studies on continuous authentication schemes based on touch biometrics, one of the behavioural biometric modalities that is based on how a user performs touch interactions on a touchscreen-based mobile device [9].

Touch biometric is one of the promising methods in continuous authentication because it allows the user to keep interacting with the device, while the authentication mechanism transparently performs the authentication task in the background without interfering with users' activity [10]. It is a non-intrusive and passive authentication method that does not require installation of additional hardware or spe-

*Corresponding author

✉ ahmad.zaidi@monash.edu, ahmadzairizaidi@gmail.com (A.Z.

Zaidi); chong.chunyong@monash.edu (C.Y. Chong);

rajendran.parthiban@monash.edu (R. Parthiban); ali.sadiq@wlv.ac.uk (A.S. Sadiq)

ORCID(s): 0000-0002-7418-0177 (A.Z. Zaidi); 0000-0003-1164-0049

(C.Y. Chong); 0000-0003-0983-9796 (R. Parthiban); 0000-0002-5746-0257

(A.S. Sadiq)

cialised equipment [11]. Therefore, it is a suitable authentication method to complement the existing initial-login authentication methods. Several studies have shown that this behavioural biometric modality has discriminative ability to distinguish between the legitimate and illegitimate users [10, 12, 11, 13, 14, 15, 16, 17]. Both the legitimate and illegitimate users can be distinguished using a classification algorithm based on the behavioural features extracted from touch actions such as touch coordinate, pressure, size of a touch area, and duration of a touch.

1.1. Motivation

Many classification methods for user classification have been employed in the area of touch-based continuous mobile device authentication (CMDA). Machine learning-based classification methods such as Support Vector Machine (SVM) [10, 18, 12, 11, 19, 17], *K*-Nearest Neighbour (*KNN*) [10, 12, 11, 19], Decision Tree (DT) [12, 14, 16], Naive Bayes (NB) [12, 19, 14, 16, 17], Logistic Regression (LR) [12, 19, 17], Neural Network (NN) [12, 11, 14, 16, 17], and Random Forest (RF) [12, 11, 19, 17] have been employed to perform the classification task. However, these classifiers are static classification methods, where the same classifier will be used to classify all touch strokes. Authentication error is a multifaceted problem because there is a wide range of factors that can influence its reliability, such as feature extraction methods, feature preprocessing methods, and training samples used. Authentication error is further challenged by the nature of the touch data that has high intra-class variability due to behavioural changes over time affected by several factors such as the hardware and software of the device, psychological and physiological states of the users, and the environment [20, 21, 11, 17]. Besides, according to the "no free lunch" theorem [22], there is no single algorithm that can solve all users' authentication and classification problems. Therefore, using a single classifier for authentication decision may produce inconsistent performance.

Multiple classifier systems (MCS) or ensemble learning method has advantage over the single classification methods, where the former can overcome the weakness of single classifiers by combining the advantage of several classifiers. Since not every classification algorithm can solve all classification problems and a particular algorithm utilises different methods to approximate the feature vectors and its class [23], several classifiers can complement each other [24]. Studies [11, 17] have shown that Random Forest, an ensemble learning method based on multiple Decision Trees, produced promising results. Random Forest, however, uses a homogeneous pool of classifiers based on Decision Tree. This method grows a diverse tree to generate the pool of classifiers instead of combining the classifiers from pre-defined base classifiers [25]. Besides, the same ensemble model will also be used for classifying test samples, which is similar to single classifiers.

Since one of the factors that influence the overall performance of a continuous authentication scheme is the classification methods [26], improving the performance of the

scheme from the perspective of classification performance is crucial in this domain. To the best of our knowledge, no study in the domain of touch-based CMDA has explored classification methods based on dynamic selection method (DS). DS is a method in MCS that performs classifier selection for each test sample, instead of using the same classifier to classify all test samples. This method consists of three phases: (1) pool of classifiers generation, (2) selection of the most competent classifier or a subset of the most competent classifiers, and (3) aggregation of selected classifiers. If only one classifier is selected, the last phase is not necessary.

1.2. Objective and Contribution

In this paper, we present a framework for user classification in touch-based CMDA, utilising DS method. The research objectives (RO) are:

- **RO1:** To investigate and analyse the performance of single classifiers and static ensemble methods across various scenarios in touch-based biometric datasets.
- **RO2:** To propose and develop a novel user classification framework in touch-based CMDA using dynamic selection of classifiers, intending to reduce the authentication error.
- **RO3:** To assess the feasibility of the classification framework by evaluating various DS methods on touch-based biometric datasets across different scenarios.

For DS methods, we used a pool of heterogeneous classifiers (analysed in RO1) because of the fact that different classifiers may have a different view on the same samples to be classified, and their outputs can complement each other [27]. The DS method will dynamically select the most competent classifier or a subset of the most competent classifiers to perform the classification task for each test sample (touch stroke). The selection is based on the competence of a classifier in the local region of the feature space. This work is inspired by the outstanding results of DS methods in other domains such as hand-digit recognition [28], remote sensing [29], credit scoring [30, 31, 32], process monitoring [33], signature verification [34], face recognition [35], and lip-based biometric verification [36]. Finally, the main contributions of this paper can be summarised as follows:

1. The introduction of DS method into touch-based CMDA. We realised this framework by evaluating several DS methods on various publicly available touch-based biometric datasets.
2. An analysis of the performance of DS methods by comparing the methods with state-of-the-art single classifiers and static ensemble methods.
3. A demonstration of the potential of DS method to reduce the authentication error in touch-based CMDA. Also, the outcome of this study help pave the way to explore the usage of DS method in touch-based CMDA

The remainder of this paper is organised as follows. Section 2 presents the background and related works on touch-based CMDA. In Section 3, we present the proposed classification framework. Section 4 describes the experimental setup used in this work. Section 5 discusses the results of the experiments. Finally, in Section 6, we conclude our findings and discuss some potential future works.

2. Background and Related Work

This study focusses on user classification in touch-based CMDA. Therefore, in this section, we first present an overview of touch-based CMDA, classification of users in touch-based CMDA, and related works in the area. We also provide an overview of MCS.

2.1. Touch-based Continuous Mobile Device Authentication (CMDA)

Continuous authentication is a user authentication scheme that allows continuous monitoring while a user is using the device. The continuous authentication scheme starts after the user has passed the initial-login session, which is the static authentication. The primary purpose of the continuous authentication scheme is to lock the illegitimate user out from the system once he or she is detected not to be the legitimate user. If the scheme does not detect any illegitimate access, the current user who is using the device will be able to continue with using the device. Touch operations on a mobile device's touch screen is a behaviour biometric where the biometric data can be collected while the user is using the device. Unlike physiological biometrics such as face and fingerprint, which requires the user to pay attention during the data acquisition phase, touch behaviour allows the data collection to be performed while the user is using the device. This biometric modality makes it more suitable for continuous authentication due to the ability for a transparent data acquisition.

In touch-based CMDA, there are two phases: enrolment and authentication phase. The enrolment phase starts with the collection of raw touch data. Raw touch data such as touch coordinates, touch pressure, touch area, and timestamp are collected from the touchscreen sensor of the mobile device. After preprocessing the raw data, features that represent the behaviour of a user will be extracted from these raw data to create a user profile. A classification algorithm is then used to model the behaviour of the user and store the user model in a database.

During the authentication phase, raw touch data from new touch samples are collected, and the features are also extracted. These features will be compared with the stored user model using a classifier to determine whether the collected touch sample belongs to legitimate or illegitimate users. The main focus of this study is on the classification part of the continuous authentication scheme. In the next section, we explain in detail how user classification is performed in touch-based CMDA.

2.2. User Classification in Touch-based CMDA

A touch-based CMDA scheme performs classification tasks to determine whether a particular touch stroke belongs to the legitimate user or not. Each touch stroke is represented by a feature vector. If the scheme classifies the feature vectors belong to an illegitimate user, the scheme will lockout the user. Otherwise, it will let the current user to continue with operating the device.

During touch operation, each touch stroke produces a vector of raw data that contains touch location, pressure, area, timestamp, and finger orientation [10]. A touch stroke (touch sample) $x \in X$ of a user $\omega_l, l \in \{I, L\}$ is represented by a vector of N features $F = \{f_1, f_2, \dots, f_N\}$ extracted from the raw data. A binary classifier c_i can be used to classify whether the features F of the touch sample x_j belongs to the legitimate user ω_L or illegitimate user ω_I where $\{\omega_L, \omega_I\} \in \Omega$ are class labels for the classification problem. A threshold θ determines the classification decision $\gamma(x_j)$ made by classifier c_i for touch sample x_j . If the classification score is higher than the threshold θ , the touch sample x_j is classified as belongs to the legitimate user ω_L or otherwise it belongs to an illegitimate user ω_I .

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

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

Many classification methods have been employed in this domain. One of the earliest works in touch-based CMDA can be found in work by Frank et al. [10]. The authors proposed a framework for user classification based on the touch interaction of users on the touch screen of mobile devices. They proposed 30 behavioural features from vertical and horizontal strokes. The authors employed two classifiers which are Support Vector Machine (SVM) and K -Nearest Neighbour (KNN) to distinguish between the features of legitimate and illegitimate users. The classifiers achieved an equal error rate (EER) between 0.00% to 4.00%.

Li et al. [18] proposed a continuous authentication scheme based not only on swipe gestures, but also tap gestures. For each type of gesture, several features were extracted. The authors collected the touch data from 75 mobile phone users. The users were allowed to use the device freely without any pre-defined tasks. Thirteen and three features were extracted from swipe and tap gestures, respectively. One classifier, which is SVM, was employed in the study. It achieved a minimum accuracy of 95.78% for sliding up gesture.

Compared to the works by Frank et al. [10] and Li et al. [18], Serwadda et al. [12] carried out a benchmark evaluation with more classifiers (10 classifiers) on a touch dataset using 28 features. The authors collected data from 190 subjects and performed the evaluation using SVM, Naive Bayes, Random Forests, KNN, Bayesian Network, Neural Network, Decision Tree, Logistic Regression, Scaled Manhattan, and Euclidean Verifier. They found that Logistic Regression per-

formed the best when tested on horizontal strokes in landscape screen orientation, which achieved an EER of 10.50%.

Shen et al. [11] studied the performance of touch-based CMDA by considering different types of touch operations (i.e. up, down, left, and right) across different application tasks (i.e. document reading, picture browsing, web surfing, and free task) and on different application scenarios (i.e. short, middle, relative-long, and long periods of authentication). The authors implemented four classifiers which are KNN, SVM, Backward Propagation Neural Network (BPNN), and Random Forest on 58 behavioural features. Based on their findings, Random Forest achieved the lowest EER of around 1.80% for left and right operations. Apart from that, the authors discovered that a combination of different types of touch operations could produce a more stable and discriminative authentication capability. Moreover, the authentication error in a specific task is lower than a free task. Specific task here refers to the task during data collection where subjects were asked to do some pre-defined tasks on the device such as document reading, picture browsing, and web surfing. On the other hand, free task refers to the task where the subjects can use the device freely. Lastly, the authors discovered that small time spans between training and testing are capable of producing reasonably good authentication performance.

Mahbub et al. [19] collected raw data from mobile devices using three main sensors: front camera, touchscreen, and location sensors. They carried out the experiments separately on the data collected from each sensor. For the touchscreen sensor, the authors collected swipes actions from users without any pre-defined task. Twenty-four features were extracted from each touch stroke. Seven classifiers were employed, namely KNN, SVM, Naive Bayes, Linear Regression, Random Tree with Linear Regression, Random Forest, and Gradient Boosting Model (GBM). The results show that Random Forest outperformed other classifiers with an EER of 22.10%.

Fierrez et al. [13] also investigated the performance of touch-based CMDA by considering different types of touch operations. They compared the performance of the scheme in three authentication scenarios. These scenarios are intra-session, inter-session, and the combination of both scenarios. A session is a period where a user starts to use the device and ends when the user stops using the device in a certain period. An intra-session scenario refers to the scenario where the training and testing of a classifier are carried out on the same session. On the other hand, an inter-session scenario refers to the scenario where the training and testing of a classifier are carried out on different sessions. They extracted 28 features based on the work by Serwadda et al. [12] and an additional five features adopted from Martinez-Diaz et al. [37]. Three classification methods were used in their experiments which are SVM, Gaussian Mixture Model (GMM), and the fusion of these two classifiers. In most cases, the performance of the fusion method is better than a single classifier. For single classifier, GMM performs better when compared to SVM with an EER of 3.60% for a right swipe in an intra-

session scenario using one of the chosen datasets. They also found that the use of the data from the latest session for enrolment could mitigate the variance in the data.

Meng et al. [14] studied the performance of touch-based CMDA under two scenarios. First, based on users' free device usage and second, web browsing usage. Based on 21 touch features from 48 subjects, they conducted a comparative study using five classifiers, namely Decision Tree (J48), Naive Bayes, Kstar, Radial Basis Function Network (RBFN), and BPNN. Besides, they also included a combined classifier Particle Swarm Optimisation with RBFN (PSO-RBFN) in their study. They found that the behaviour deviation is smaller during web browsing compared to the scenario of free usage with PSO-RBFN performed the best with EER of 2.38%.

Syed et al. [17] studied the effect of user posture, device size, and device configuration to improve the performance of existing touch-based CMDA. The authors proposed 14 features and employed five classifiers: SVM, Logistic Regression, Naive Bayes, Random Forest, and Multilayer Perception (MLP). They found that Random Forest produces the best results. The best EER is recorded at 3.80% when the model was trained and tested using the same posture, which is when the device was held in landscape orientation. The study shows that user posture, device size, and device configuration have a significant effect on the classifier performance and thus should be considered when developing a touch-based CMDA scheme.

In general, the literature summarised above employed single classifiers to perform user classification (see Table 1). It can be observed from different studies that the performance of a particular classifier varied from one study to another. This issue might be due to the differences in data acquisition protocol, feature extraction methods, and experimental setup that varies from one study to another. Besides, intra-class variability can also affect the performance of the classifiers. Thus, using the same classifier on different scenarios may lead to an unstable performance of the continuous authentication scheme. One possible way to overcome the drawback of single classifiers is by using multiple classifier systems (MCS) or also known as ensemble learning method, by combining the outputs of multiple classifiers for user classification tasks. Some studies [12, 11, 19, 17] used Random Forest, an ensemble learning method for the classification task. However, there is a lack of study on MCS in the domain of touch-based CMDA. The next section presents an overview of MCS.

2.3. Multiple Classifier Systems (MCS)

Multiple classifier systems (MCS) or ensemble learning is a classification method that performs classification decision based on the combination of several classifiers to overcome the drawbacks of single classifiers [38] and to improve the performance of a classification task [39]. In other words, the classification problem does not rely only on a single classifier, but by a joined decision of classifiers [40]. Many classification methods had been proposed in the literature. It is

Table 1
Summary and performance of single classifiers in the chosen literature

Author	Dataset	Feature	Classifiers	SVM	KNN	DT	NB	LR	NN	RF	Performance (%)
Frank et al. [10]	1	27	2	✓	✓	-	-	-	-	-	EER = 0.00-4.00
Li et al. [18]	1	16	1	✓	-	-	-	-	-	-	ACC = 95.78
Serwadda et al. [12]	1	28	10	✓	✓	✓	✓	☒	✓	✓	EER = 10.50
Shen et al. [11]	1	58	4	✓	✓	-	-	-	✓	☒	EER = 1.80
Mahbub et al. [19]	1	24	7	✓	✓	-	✓	✓	-	☒	EER = 22.10
Fierrez et al. [13]	4	33	2	✓	-	-	-	-	-	-	EER = 3.60
Meng et al. [14]	1	21	6	-	-	✓	✓	-	☒	-	AER = 2.38
Meng et al. [16]	1	9	5	☒	-	✓	✓	-	✓	-	AER = 4.66
Syed et al. [17]	1	14	5	✓	-	-	✓	✓	✓	☒	EER = 3.80

Abbreviations have the following meaning: SVM = Support Vector Machine, KNN = K -Nearest Neighbour, DT = Decision Tree, NB = Naive Bayes, LR = Logistic Regression, NN = Neural Network, RF=Random Forest, EER = Equal error rate, AER = Average error rate, ACC = Classification accuracy. ✓ indicates the classifier employed in the study and ☒ indicates the classifier was the best in the study.

generally acknowledged that no one classification algorithm is effectively capable of handling all classification problems [41]. Satisfactory results of MCS can be achieved if the base classifiers in the pool are of high quality and the preparation of the classification model depends on the structure of the ensemble method [36].

A diverse and accurate pool of classifiers has to be generated to achieve a satisfactory result of a MCS [41, 42]. The pool can comprise of homogeneous classifiers or heterogeneous classifiers [41, 42]. A diverse pool of classifiers can be generated based on different initialisation, parameters, architectures, classifiers models, training sets, and feature sets [42]. This diversity will create a pool of classifiers $C = \{c_1, \dots, c_M\}$, with M amount of classifiers. The scheme will generate several models where the decision is made based on the combination of the models either by using classifier fusion or classifier selection [43].

In classifier fusion, each classifier in the pool contributes towards the final decision [44]. The outputs of each model are combined to achieve higher accuracy compared to a single classifier [29]. It assumes that all classifiers are competent in the entire feature space [29]. The fusion can be performed in several methods such as minimum, maximum, average, median, and majority vote. It is worth to note that classifier fusion is a form of decision-level fusion method [29]. If there exist some redundant and inaccurate classifiers in the pool, the performance of this method will be affected [43].

On the other hand, classifier selection performs the final classification decision based on the selection of a single classifier or a subset of classifiers [43]. It can achieve a better performance than classifier fusion since classifier selection chooses the most promising single classifiers from the pool instead of smoothing out the differences of single classifiers [33]. Generally, there are two main classifier selection methods, which are static selection and dynamic selection (DS) [41, 42].

Static classifier selection method performs the selection of classifier(s) during the training phase, where it will use

the same classifier(s) to classify all test samples. On the other hand, DS method selects the most promising classifier(s) for each test sample during the test phase (different classifiers have different expertise in the local region of the feature space). Besides, DS method performs the selection according to the competence of a base classifier in the pool. The selection is based on the measure of competence in each region of competence of a test sample. It is worth to note that test samples, in general, have different classification difficulties. Therefore, using the relevant classifier for each test sample is useful compared to using the same classifiers for all test samples as different classifiers have different expertise in handling a particular classification task [45].

In the domain of touch-based CMDA, Meng et al. [16] have presented a classification selection method based on an intelligent cost mechanism that can select a less costly classifier from a pool of heterogeneous classifiers. The selection was performed periodically. The pool of classifiers consists of Decision Tree, Naive Bayes, RBFN, BPNN, and SVM. The performance of the method is relatively better and more stable compared to a single classifier. However, an extensive study is necessary to explore various methods of classifier selection, especially DS that has the ability to select the most competent classifier(s) for each test sample. This exploration may be helpful in improving the authentication performance as the outstanding results in other domains have been shown in several studies [28, 29, 34, 35, 30, 33, 31, 32, 36]. Our work further explores different classifier selection methods. Our focus will be on DS methods. There are two types of selection approach in DS: dynamic classifier selection (DCS) and dynamic ensemble selection (DES). The former selects a single classifier to perform the classification task, while the latter selects a subset of classifiers and combine them to perform the classification decision. The next section will explain further on these two types of selection approaches. Specifically, Section 3.3.3 explains these two selection approaches.

3. Proposed Framework

In this section, we present a classification framework for user classification in touch-based CMDA. We first introduce the concept of DS method and describe each step in the framework.

3.1. Overview

Various classification methods have been employed in the area of touch-based CMDA. However, these methods are static classification methods, where the same classifier will be used to classify all touch strokes. Authentication error is a crucial problem because there is a wide range of factors that can influence its reliability. Authentication error is also further challenged by the nature of the touch data that has high intra-class variability. On top of that, according to the "no free lunch" theorem [22], there is no single algorithm that can solve all users authentication and classification problems. Therefore, using a single classifier for authentication decision may produce inconsistent performance and may also challenge the reliability of the authentication scheme.

The rationale behind DS method is the selection of the most competent classifier c_i or a subset of the most competent classifiers C' from a pool of M classifiers $C = \{c_1, \dots, c_M\}$ to perform the classification task for a test sample x_j . The method estimates the competence level δ_{ij} of each base classifier $c_i \in C$ [42]. The competence level δ_{ij} is estimated using a measure of competence in the local region θ_j of the feature space where a test sample x_j is located. It is presumed that each classifier is an expert in its local region in the feature space [29]. This idea can be seen in the real world where a particular decision can be made based on experts with different background and expertise.

DS method performs the selection of classifier(s) for each test sample x_j during the test phase, where it selects a single classifier c_i or a subset of classifiers C' [42]. Figure 1 show the proposed framework for user classification in touch-based CMDA based on DS method. There are three main phases in this framework: (1) generation, (2) selection, and (3) aggregation.

In order to realise this framework, we evaluated various DS methods. Table 2 shows the DS methods evaluated in this study. The methods chosen in this study differ in terms of the region of competence definition, selection criteria, and selection approach. It is worth to note that the main challenges in DS are the region of competence definition and the measure of competence of base classifiers [46]. Therefore, in this study, we evaluated the DS methods that have different ways of defining the region of competence (Section 3.3.1) and measure of competence level of base classifiers (Section 3.3.2). Algorithm 1 presents the general phases in DS methods.

3.2. Pool of Classifiers Generation

Multiple classifiers can be generated based on different methods such as based on homogeneous or heterogeneous base classifiers. Pool of homogeneous classifiers can be gen-

erated based on algorithm initialisations, parameter settings, algorithm architecture, training sets, and feature sets [42]. On the other hand, pool of heterogeneous classifiers can be generated based on different types of classification algorithms (eg. SVM, KNN and etc.) [42].

The classifiers in the pool should be accurate and diverse [56]. Accurate classifiers refer to the classifiers that have a classification error rate lower than random guessing, whereas diversity refers to the error made by the classifiers are different from each other. It is worth to note that an MCS will not be able to outperform the base classifiers if the classifiers in the pool are same [57]. Therefore, ensuring the diversity of the pool is crucial. On top of that, the base classifiers in the pool are trained using a training set D_{train} .

In this study, we focus on the generation of pool classifiers $C = \{c_1, \dots, c_M\}$, of $M = 6$ heterogeneous classifiers. Existing studies in the domain have employed various classification algorithms to classify users. Therefore, we used some of the classifiers that have been widely employed in the literature to generate the pool of base classifiers. Below are the six shortlisted base classifiers:

1. **K-Nearest Neighbour (KNN)** [58]: 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.
2. **Support Vector Machine (SVM)** [59]: A discriminative classification method that separates the features of a legitimate user and illegitimate users using maximised hyperplane [60]. 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.
3. **Decision Tree (DT)** [61]: 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.
4. **Naive Bayes (NB)** [62]: A probabilistic method based on Bayes theorem [63]. It classifies a touch stroke based on the probability that it belongs to a particular class.
5. **Logistic Regression (LR)** [64]: 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.
6. **Neural Network (NN)** [62]: This method was inspired by the neural network of the human brain. It consists of an input, hidden layers, and an output [65]. 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

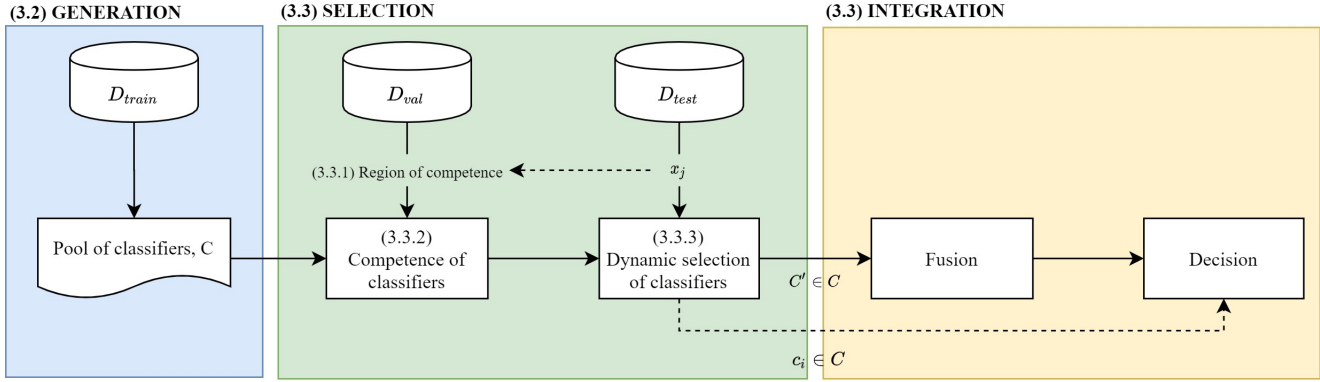


Figure 1: The framework of user classification in touch-based CMDA using DS method. It consists of three phases: (1) generation, (2) selection, and (3) aggregation. During the test phase, a touch stroke x_j will be classified by the most competent classifier $c_i \in C$ or a subset of classifiers $C' \in C$. In the later case, fusion is needed to combine the selected classifiers.

Table 2

DS methods evaluated in this work, summarised in terms of the region of competence definition, selection criteria, and selection approach.

Technique	Acronym	Region of competence definition	Selection criteria	Selection approach
Classifier Rank [28]	DCS-Rank	K NN	Ranking	DCS
Overall Local Accuracy [44]	DCS-OLA	K NN	Accuracy	DCS
Local Class Accuracy [44]	DCS-LCA	K NN	Accuracy	DCS
A Priori [47]	DCS-Priori	K NN	Probabilistic	DCS
A Posteriori [47]	DCS-Posteriori	K NN	Probabilistic	DCS
Multiple Classifier Behavior [48]	DCS-MCB	K NN	Behaviour	DCS
Modified Local Accuracy [29]	DCS-MLA	K NN	Accuracy	DCS
DES-Clustering [49, 50]	DES-Clustering	Clustering	Accuracy & Diversity	DES
DES-KNN [49, 50]	DES-KNN	K NN	Accuracy & Diversity	DES
K-Nearest Oracles Eliminate [51]	DES-KNORAE	K NN	Oracle	DES
K-Nearest Oracles Union [51]	DES-KNORAU	K NN	Oracle	DES
DES-Exponential [52]	DES-Exp	Potential function	Probabilistic	DES
DES-Logarithmic [52]	DES-Log	Potential function	Probabilistic	DES
DES Minimum Difference [53]	DES-MD	Potential function	Probabilistic	DES
DES Randomized Reference Classifier [43]	DES-RRC	Potential function	Probabilistic	DES
DES Performance [25]	DES-P	K NN	Accuracy	DES
DES Kullback-Leibler [25]	DES-KL	Potential function	Probabilistic	DES
K-Nearest Output Profiles [54]	DES-KNOP	K NN	Behaviour	DES
META-DES [55]	META-DES	K NN	Meta-Learning	DES

several iterations. Specifically, multi-layer perceptron (MLP) was used in this study.

We maintain the number of base classifiers to be small since increasing the size of the pool may increase the possibility of choosing a non-competent classifier [66]. Also, we used different types of classifiers to ensure the diversity of the pool. On the other hand, an MCS with homogeneous classifiers requires more advanced training to create diversity amongst the base classifiers (e.g. different training sets and different feature sets) [36].

3.3. Selection

The main difference between DS and static classifier selection methods is the selection of classifier(s) are performed during the test phase. During the test phase, the region of competence of a test sample x_j will be determined. Based

on the selected region, the most competent classifier(s) in the region will be selected to perform the classification task.

3.3.1. Region of Competence Definition

The region of competence of a base classifier is first defined based on different parts of the feature space, where the region is also known as local regions [31]. The region of competence θ_j of a touch stroke x_j is the K data points in the validation set D_{val} such that $\theta_j = \{x_1, \dots, x_K\}$. The feature space is divided into different regions where the most competent classifier is determined based on these partitions.

In this study, we evaluated various methods of defining the region of competence based on the DS methods employed in Table 2. These methods include K -nearest neighbours [28, 44, 47, 48, 29, 49, 50, 51, 54, 55], clustering [49, 50], and potential function [52, 43, 25]. For K NN, the region of competence θ_j of a test sample x_j is defined as the

Algorithm 1: General DS Algorithm

Input: Training set D_{train} , validation set D_{val} , test set D_{test} , a pool of classifiers C , size of competence region K .
Output: The most competent classifier $c_i \in C$ or a subset of the most competent classifiers $C' \subset C$ for each test sample $x_j \in D_{test}$.

Train M base classifiers $C = \{c_1, \dots, c_M\}$ using training set D_{train} (as described in Section 3.2);

for each test sample $x_j \in D_{test}$ **do**

Define the region of competence using one of the methods described Section 3.3.1;

Compute the level of competence δ_{ij} for each classifier $c_i \in C$ using one of the methods described in Section 3.3.2;

if $C' \neq \emptyset$ **then**

Select a subset of the most competent classifiers C' from the pool of classifiers C (i.e. using one of the DES methods as described Section 3.3.3);

Combine C' using the aggregation method as described in Section 3.4 to classify x_j ;

else

Select the most competent classifier c_i from the pool of classifiers C (i.e. using one of the DCS methods as described in Section 3.3.3);

Use the selected classifier c_i to classify x_j ;

end if

end for

K -nearest samples in the validation set D_{val} from x_j . The step will be carried out during the test phase. In order to find the K -nearest neighbour of x_j , the distance from x_j to all data points in D_{val} has to be computed. This process will increase the computational time if the size of D_{val} is large [33]. Also, determining the number of K is a challenging task since the choice of the number may affect the performance of the DS method [48]. In this study, we set the number of $K = 7$ because it is commonly used in existing DS studies [42, 31, 23].

On the other hand, for clustering methods, the region of competence θ_j is defined using a clustering algorithm during the training phase, and the most competent classifiers are determined for each cluster. During the test phase, the scheme will determine which cluster of a test sample x_j belongs to by calculating the distance between sample x_j and the centroid of each cluster. Therefore, the sample x_j will be assigned to a cluster that has the shortest distance with the centroid. As a result, this method requires lower computation cost since only the distance between the test sample and the centroid of each cluster is calculated. However, the definition of region of competence using the clustering method is less precise compared to KNN method [42]. In this study, we set the number of cluster to 2.

In contrast to both KNN and clustering methods that define the region of competence θ_j by considering only some of the samples in the validation set D_{val} , potential function [67] utilises the whole D_{val} to measure the level of competence of a classifier c_i by weighting it with the distance of sample $x_k, k \in D_{val}$ with the test sample $x_j, j \in D_{train}$ using a potential function. Therefore, this method originally does not need to set the number of nearest neighbours or the number of clusters. However, it needs a higher computational cost due to the computation of the distance between the test samples and the samples in the validation set D_{val} [42]. Therefore, in our study, to reduce the computational

cost, we set the size of the region to 7.

3.3.2. Level of Competence Estimation

In each region of competence, the most competent classifier will be determined based on a particular 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. Several criteria have been proposed to estimate the competence level of a classifier. The validation set D_{val} is utilised to evaluate the base classifiers, and therefore the outcome will be used to measure the competence of each base classifier x_j .

The DS methods employed in this study estimate the competence of a classifier based on the criteria such as ranking [28], accuracy [44, 29], accuracy and diversity [49, 50], probabilistic [47, 52, 43, 25], classifier behaviour [48, 54], Oracle [51], and meta-learning [55]. The next section will explain about the DS methods based on the aforementioned selection criteria.

3.3.3. Selection Approach

In the selection phase, one or more classifiers will be selected to perform the classification task. There are two main selection approaches in DS method, namely Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES). DCS will select the classifier that has the highest competence level in a particular region of the feature space to perform the classification task. The selection of a single classifier depends on the generalisability of the classifier [66]. On the other hand, DES will select a subset of the most competent classifiers and aggregate them to make the classification decision [42]. This approach can reduce the risk of over-generalise of a selected single classifier by distributing the risk through selecting a subset of classifiers from the pool [66]. This selection approach can be considered as the com-

bination of classifier selection and classifier fusion, where a subset of classifiers are selected and aggregated using a fusion method.

In this study, seven DCS methods were employed as follows. The classifier that has the highest competence based on each particular method will be selected to perform the classification of the test sample x_j . For all methods, the region of competence $\theta_j = \{x_1, \dots, x_k\}$ is defined using the K -nearest neighbour of test sample x_j .

1. **Classifier Rank (DCS-Rank) [28]:** The competence level δ_{ij} of a base classifier c_i is estimated by first calculating the number of consecutive correct classification of samples in the region of competence θ_j of a test sample x_j . Then, classifier c_i will be given the highest rank if it has the highest number of correct classification for consecutive samples and selected as the most competent classifier.
2. **Overall Local Accuracy (DCS-OLA) [44]:** The competence level δ_{ij} of a base classifier c_i is estimated based on the accuracy of correct classification of samples in the local region θ_j of a test sample x_j . The classifier with the highest competence level based on this measure will be selected to classify the test sample x_j .
3. **Local Class Accuracy (DCS-LCA) [44]:** The competence level δ_{ij} of a base classifier c_i is also estimated based on the accuracy of correct classification of samples in the region of competence θ_j of a test sample x_j , but defined only for the class label assigned for a sample x_j by classifier c_i . The classifier with the highest competence level based on this measure will be selected to classify the test sample x_j .
4. **A Priori (DCS-Priori) [47]:** The competence level δ_{ij} of a base classifier c_i is estimated based on the posterior probability of each class label. The measure then weighted with the distance (based on Euclidean distance) from a test sample x_j to the samples in the region of competence θ_j . The classifier with the competence measure higher than a predefined threshold will be selected to classify the test sample x_j (significantly better than the other base classifiers). If no such a classifier exists, all base classifiers will be aggregated using a majority voting rule.
5. **A Posteriori (DCS-Posteriori) [47]:** The competence level δ_{ij} of a base classifier c_i is also estimated based on the posterior probability but also based on the class predicted by the classifier c_i . The measure is then weighted with the distance (based on Euclidean distance) from a test sample x_j to the samples in the local region θ_j . The classifier with the competence measure higher than a predefined threshold will be selected to classify the test sample x_j (significantly better than the other base classifier). Otherwise, all base classifiers will be aggregated using a majority voting rule.
6. **Multiple Classifier Behavior (DCS-MCB) [48]:** The competence level δ_{ij} of a base classifier c_i is estimated based on the similarities between the output profile of

a test sample x_j and the region of competence using BKS algorithm [68]. The region of competence θ_j is vary depending on this similarity. The samples in the region with lower similarity than the predefined threshold will be removed from the region of competence θ_j . The competence level δ_{ij} of a base classifier c_i is then estimated as the classification accuracy of the final region of competence θ_j . Therefore, the classifier c_i that is higher than a predefined threshold will be selected to classify the test sample x_j . Otherwise, all base classifiers will be aggregated using a majority voting rule.

7. **Modified Local Accuracy (DCS-MLA) [29]:** The region of competence θ_j of a test sample x_j is defined in such a way that the distance between the samples in the region of competence θ_j and the test sample x_j is considered to overcome the issue where a larger number of samples in the region will include samples that are not similar to x_j while a lower number of samples will exclude samples that are similar to x_j . The samples in the region of competence θ_j is weighted by their Euclidean distance to the test sample x_j so that the samples in the region of competence θ_j that are closer to the test sample x_j have a greater influence in estimating the competence level. The competence level δ_{ij} of the classifier will be estimated based on the accuracy in this defined region θ_j . The classifier that has the highest competence level will be selected to classify the test sample x_j .

For DES, 12 methods were employed. These methods differ in the criteria used for estimating the level of competence and the method used for defining a region of competence. The DES methods are described as follows.

1. **DES-Clustering [49, 50]:** The competence level δ_{ij} of a base classifier c_i is estimated by considering the accuracy and diversity (based on Double Fault measure [69]) of the classifier. The ranking is made so that the accuracy is in decreasing order, and the diversity is in increasing order to rank the most accurate and the most diverse classifiers, respectively. The region of competence θ_j is defined using the K-Means clustering algorithm by partitioning the validation set D_{val} during the training phase. The selection of the subset of the most competent classifiers C' is performed during the training phase for each respective cluster. During the test phase, the distance between the centroid of each cluster and a test sample x_j is calculated to find the closest cluster, where each cluster has been determined to have the most competent classifiers C' during the training phase. Finally, the top N accurate and J diverse classifiers will be selected as the most competent classifiers C' to classify the test sample x_j .
2. **DES-KNN [49, 50]:** The competence level δ_{ij} of a base classifier c_j is estimated by considering the accuracy and diversity (based on Double Fault measure [69]) of the classifier, by ranking accuracy in decreasing order, and diversity in increasing order, to rank the

most accurate and the most diverse classifiers, respectively. The region of competence θ_j is defined using K -nearest neighbour of a test sample x_j . The top N accurate and J diverse classifiers will be determined as the most competent classifiers C' by utilising the samples in the region of competence θ_j .

3. **K-Nearest Oracles Eliminate (DES-KNORAE) [51]:** The competence level δ_{ij} of a base classifier c_i is estimated based on the concept of Oracle [70]. The classifiers that can correctly classify all samples in the region of competence θ_j will be selected to form a subset of classifiers C' . If there are no such classifiers, the region of competence θ_j will be reduced by eliminating some samples in the region. The search for the classifiers that can meet this criterion will continue until the criterion is achieved. Once this criterion achieved, those classifiers are selected as the most competent classifiers C' .
4. **K-Nearest Oracles Union (DES-KNORAU) [51]:** The competence level δ_{ij} of a base classifier c_i is also estimated based on the concept of Oracle [70]. It performs the selection of all classifiers that can correctly classify at least one sample in the region of competence θ_j . The classifiers that correctly classifies the sample in the region of competence θ_j will provide the votes, and the votes will be used for the classification decision. The number of votes of a classifier c_i is equal to the number of correctly classified samples in the region of competence θ_j . These votes are collected from all base classifiers to be aggregated for the final classification decision.
5. **DES-Exponential (DES-Exp) [52]:** The competence level δ_{ij} of a base classifier c_i is estimated with respect to random guessing. This method utilises the class support produced by the classifier c_i for class ω_l . It defines the source competence of the classifier c_i at a point x_j in the validation set D_{val} using an exponential function. The competence level δ_{ij} of classifier c_i is generalised to the entire feature space by weighting the source competence with a potential function. The classifier is competent if the competence level is greater than 0 and will be selected and aggregated to perform the classification task.
6. **DES-Logarithmic (DES-Log) [52]:** The competence level δ_{ij} of a base classifier c_i estimated with respect to random guessing. This method utilises the class support produced by the classifier c_i for class ω_l . It defines the source competence of the classifier c_i at a point x_j in the validation set D_{val} using a logarithmic function. The competence level δ_{ij} of the base classifier c_i is then generalised to the entire feature space by weighting the source competence with a potential function. The classifier is competent if the competence level is greater than 0 and will be selected and aggregated to perform the classification task.
7. **DES Minimum Difference (DES-MD) [53] :** The source of competence of a base classifier c_i is first estimated based on the uncertainty of the classifier's decision and its correctness. This criterion is known as Minimal Difference. If the source of competence is higher than 0, the classifier c_i is considered as competent. Otherwise, it is considered as incompetent, where the source of competence te is lower than 0. These values then generalised to the entire feature space using a potential function. The subset of the most competent classifiers $C' \in C$ are selected and aggregated.
8. **Randomized Reference Classifier (DES-RRC) [43]:** The competence level δ_{ij} of a base classifier c_i is estimated based on the concept of a hypothetical randomised reference classifier (RRC), where the probability of correct classification by RRC is utilised. The RRC produces the class supports based on random variables produced by a beta probability distribution, where expected values of the class support produced by the RRC is equal to the class supports produced by a base classifier c_i . The competence level δ_{ij} of a base classifier c_i is generalised to the entire feature using a potential function. The subset of classifiers $C' \subseteq C$ are selected for the classification task if the competence is higher than the probability of RRC.
9. **DES-Performance (DES-P) [25]:** The competence level δ_{ij} of a base classifier c_i is estimated based on the difference between the accuracy of the base classifier c_i in the region competence θ_j (defined by weighted KNN [29]) and the performance of the random classifier c_{RC} that randomly chooses a class with equal probabilities. A subset of classifiers $C' \subseteq C$ with a positive value of competence level δ_{ij} are selected (C' performs better than c_{RC} .) and aggregated to perform the classification task for the test sample x_j .
10. **Kullback–Leibler (DES-KL):** The competence level δ_{ij} of a base classifier c_i is estimated based on information theory perspective, which is calculated as the Kullback–Leibler divergence between the uniform distribution and the vector of class support of the classifier c_i . It measures the closeness between the functions and the probability of random classification. This source of competence then generalised to the entire feature space using a potential function. A subset of classifiers $C' \subseteq C$ with a positive value of competence level δ_{ij} are selected and aggregated to perform the classification task for the test sample x_j .
11. **K-Nearest Output Profiles (DES-KNOP) [54]:** The competence level δ_{ij} of classifier c_i is estimated based on the similarity between the test sample x_j and the samples in the region of competence θ_j , in the form of output profiles. The region of competence θ_j is defined based on the output profiles, which is the decision of the base classifier c_i . This method performs the selection of classifiers that correctly classified at least one sample in the region of competence θ_j . Each classifier provides the vote, and the votes from all classifiers are aggregated for the final classification decision.

sion.

12. **META-DES [55]:** The competence level δ_{ij} of a base classifier c_j is estimated based on multiple criteria, which include neighbours' hard classification, posterior probability, overall local accuracy, output profiles classification, and classifier's confidence. This method views the criteria as a meta-problem, where it extracts meta-features of each criterion and trains a meta-classifier (using Naive Bayes) to determine whether the base classifier c_j is competent or not (meta-classes) to classify the test sample x_j . The base classifiers C' that achieved a certain level of competence are selected and then aggregated for the final classification decision.

3.4. Aggregation

In the aggregation phase, if more than one classifier were selected (i.e. DES methods), a fusion method is used to make the final decision [41]. Combination rules such as majority voting (MV), maximum, minimum, or trainable fusers (e.g., stacking) are usually used. In the case where only one classifier is selected (i.e. DCS), no aggregation is needed [41]. In this study, all DES methods use majority voting as the aggregation method. The reason of this choice is the simplicity of MV in combining multiple classifiers.

4. Experimental Setup

In this section, we present the experimental setup to evaluate our proposed framework. The experimental setup consists of several components, namely, datasets, benchmark methods, model training and testing, and evaluation metrics. We also describe the statistical test of significance that we used for our analysis.

All experiments in this study were carried out using the Python machine learning package, Scikit-learn [71] and an ensemble learning library, DESlib [72] for the DS methods. The experiments were performed on a PC with 2.4 GHz, Intel(R) Xeon(R) CPU and 32 GB RAM, using the Microsoft Windows 10 Enterprise operating system.

4.1. Dataset

We performed the experiments in this study using four publicly available touch biometric datasets. All datasets consist of swipe gestures as the touch operation. Swipe gestures are the most frequently performed strokes during normal device usage where users perform activities such as browsing web pages, reading texts, and viewing images. Therefore, this type of gesture is suitable for touch-based CMDA [12].

Table 3 shows the details of the datasets. The datasets differ in terms of the number of subjects, the number of sessions, the duration taken to acquire the data, data collection environment, and the number of features. We also used the original features presented by the authors of the datasets to investigate the performance of the proposed method with different kinds of features.

4.1.1. Frank Dataset

Frank dataset [10] consists of swipes data from 41 subjects on Android smartphones. The subjects were required to read texts and compare images, which produced vertical and horizontal strokes, respectively, in several sessions. Initially, there were three sessions for text reading and two sessions for image comparison on the same day. Between each session, there were several minutes of intervals before continuing to the next session on that day. After at least a week, some subjects participated in another session of data collection for the same tasks, but only one session for each text reading and image comparison. The authors presented 27 touch-based features.

4.1.2. Serwadda Dataset

Serwadda dataset [12] consists of swipe data from 190 subjects using an Android device, in two separate sessions with an interval of at least one day. In each session, subjects were required to answer a series of multiple-choice questions. The tasks produced horizontal and vertical swipes when subjects swiped back and forth to answer the questions. Any short strokes that have four or fewer touchpoints were considered as outliers and were discarded. For this dataset, the portrait and landscape orientation of the screen were processed separately. Compared to Frank dataset, the orientation of the screen is included as part of the features. In Serwadda dataset, for each direction of stroke, there are two screen orientations involved. The authors presented 28 features for each swipe action.

4.1.3. Antal Dataset

Antal et al. [73] collected touch data from 71 subjects using eight different Android devices in four weeks. The subjects were required to perform tasks related to the reading of documents and browsing of an image gallery, which produce vertical and horizontal swipes, respectively. Each subject was required to perform the tasks in multiple sessions and might have used different devices in different sessions. The authors presented 15 features for each swipe action.

4.1.4. Mahbub Dataset

Mahbub et al. [19] presented a multi-modal dataset, called as the University of Maryland Active Authentication Dataset 02 (UMDAA-02) dataset. It is originally a multi-modal dataset for continuous authentication. The data were acquired from multiple sensors, including the front-facing camera, touchscreen, gyroscope, accelerometer, magnetometer, light sensor, GPS, Bluetooth, WiFi, proximity sensor, temperature sensor, and pressure sensor. The data were collected from 48 subjects using smartphones for a period of over two months. We chose this dataset for our experiments because it includes touch data from swipes actions. We did not use data from other sensors other than those from the touchscreen in this study, as our main focus is on touch-based CMDA. Besides, there are no pre-defined tasks to perform during data collection (or in other words, free tasks), which is quite different from all the other datasets. Compared with other datasets that have multiple sessions, the sessions of Mahbub dataset

Table 3
Summary of the selected datasets

Dataset	Subject	Session	Duration	Interval	Environment	Features
Frank [10]	41	7	25 - 50 minutes	Several minutes	Controlled	30
Serwadda [12]	190	2	-	≥ 1 day	Controlled	28
Antal [73]	71	-	4 weeks	-	Controlled	15
Mahbub [19]	48	~ 248	1 week	-	Uncontrolled	24

are not based on a specific duration, but instead, each session starts and ends when the device is unlocked and locked, respectively. There are 24 features presented by the authors. To prevent outliers, swipes with less than five data points were discarded in our study.

For all datasets, there are different types of touch-based features. The range of values varies from one feature to another. To avoid any bias in model building, we transformed the feature data by scaling the values of all features in the range of $[0, 1]$ using Min-Max Scaler. Besides, these datasets can also be obtained from [13].

4.2. Benchmark Methods and Hyper-parameter setting

To evaluate the performance of the DS methods, we compared their performance with some benchmark methods. The first group of benchmark methods are the single classifiers that form the pool of classifiers. The single classifiers include SVM, NB, DT, KNN, LR and NN (MLP), as described in Section 3.2. Second, we employed static ensemble methods, which include:

- **Random Forest (RF) [74]**. It generates multiple decision trees and combines each output of the trees using majority voting to obtain the final classification decision.
- **Majority Voting (MV) [75]**: The outputs of each base classifier in the pool, as discussed in Section 3.2, are combined using the majority voting rule to make the classification decision.
- **Static Selection (SS) [76]**: It selects the best-performing classifiers in the pool and combines them.
- **Single Best (SB) [76]**: It selects the base classifier in the pool that achieve the highest accuracy based on the validation set D_{val} .

It worth to note that RF is a homogeneous ensemble classifier that consists of multiple decision trees (the same type of classification algorithm), whereas MV, SS and SB are heterogeneous ensemble classifiers that consist of different types of classifiers as discussed in Section 3.2. RF and MV aggregate all base classifiers in the original pool of classifiers C . On the other hand, SB and SS perform classifier(s) selection first before performing the final classification decision. Also, we would like to note that SB only selects a

single classifier $c_i \in C$ while SS selects a subset of classifiers $C' \subset C$. Finally, if an MCS involves a selection of classifier(s), (i.e. SB and SS), the same selected classifier(s) will be used to classify all test samples. This type of MCS is known as a static selection of classifier(s), which different from DS [41, 42], where the selection of classifier(s) is performed for each test sample.

Also, it is recommended that for a DS method to surpass MV, SS, and SB of the same pool as the minimum requirement [41]. The classification methods mentioned above also have hyper-parameters that may impact the performance. Therefore, we employed a grid search algorithm to tune the hyper-parameters for each classification algorithm. Table 4 shows the hyper-parameters for each classification algorithm and its respective values. The hyper-parameters that are not mentioned in this table used the default setting based on their respective algorithm's implementation.

4.3. Model Training and Evaluation

We treated the classification problem as a binary classification task, where device-sharing is possible (i.e. children, family members, or acquaintances might borrow the device) [77]. Therefore, we employed binary classifiers to model a user profile considering this situation. For each subject, we first choose one of the subjects as the legitimate user. We then randomly choose some other subjects as illegitimate users. We repeated this step for all users in the dataset. We aim to simulate the situation where the illegitimate users have gained access to the device and are trying to access the information stored on the device. In other words, we assume that the device was unlocked, which allows the illegitimate users to access it, or the illegitimate users could by-pass the initial login session. This simulation can be considered as the approximation to the real situation of unknown attacker [20]. In this case, our threat model is considered as *random attacks*. In other words, illegitimate users do not have the information about the legitimate user to imitate the legitimate user's touch gestures.

The base classifiers were trained using a training set D_{train} . N randomly chosen samples from a legitimate user's data and also N samples combined from the other $N/10$ randomly chosen subjects as the illegitimate user's data. Thus, each illegitimate user contributes 10 training samples. In this study, we used $N = 40$ training samples to train the classifiers, and the other four users were randomly chosen as the data of illegitimate users. As discussed in the work by Serwadda et al. [12], we also set the same number of training

Table 4
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
RF	Number of trees	100

samples across different subjects to obtain the most uniform results and to avoid any bias that may arise in the case where some users have more or fewer samples. In order to ensure this requirement is met, we set the minimum number of samples for each user to 60 for a particular scenario of the experiments. Subjects who did not meet this minimum number of samples in a particular scenario were excluded from the evaluation. Furthermore, the samples that were not chosen as training samples for both legitimate and illegitimate users were used as the test samples to form the test set D_{test} . It is worth noting that some datasets have two main scenarios in the experiments, which are intra-session and inter-session. In the former scenario, we trained and tested a model on the dataset of the same session. While in the latter scenario, we trained a model with one session and tested it with a later session to ensure that the test samples have been generated later than the training samples.

Besides, in order to estimate the competence of classifiers, we used a validation set D_{val} . Similar to [78], due to the low sample size from each user, the whole training set was used as the validation set. The region of competence of a test sample was defined using this validation set. We estimated the competence of the base classifiers in the pool to classify a particular test sample by considering the samples that belong to this region.

4.4. Evaluation Metric

Evaluation metric plays a vital role in evaluating the performance of a classification method. Since this study focusses on the security mechanism of mobile devices, it is crucial to choose the right evaluation metric that suits the purpose. In this study, we evaluated the performance of the classification methods based on the authentication error rates. The error rates can be measured in terms of false acceptance rate (FAR) and false rejection rate (FRR).

FAR is the ratio between the number of touch strokes of illegitimate users that were wrongly classified as the touch strokes of the legitimate user, and the total number of test samples of the illegitimate user, as shown in Equation 1:

$$FAR = \frac{\sum X(I, L)}{\sum X(I)} \times 100\% \quad (1)$$

where $X(I, L)$ is the number of touch samples of illegitimate users that are classified as touch samples from the legitimate user, and $X(I)$ is the total number of touch samples of illegitimate users.

On the other hand, FRR is the ratio between the number of test samples of the legitimate user that were wrongly classified as the touch strokes of illegitimate users, and the total number of touch strokes of the legitimate user. Equation 2 shows the calculation of FRR.

$$FRR = \frac{\sum X(L, I)}{\sum X(L)} \times 100\% \quad (2)$$

where $X(L, I)$ is the number of touch samples of legitimate users classified as touch samples from the illegitimate user and $X(L)$ is the total number of touch samples of legitimate users.

FAR and FRR, respectively measure the security and usability of a particular scheme. A threshold can be adjusted to make the scheme more usable or more secure [17]. The threshold can be lowered to increase the 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

security as well as usability and the overall performance of a scheme. Equation 3 shows the calculation of EER.

$$EER = \frac{FAR + FRR}{2} \quad (3)$$

where the difference between FAR and FRR should have the smallest value based on the variation of thresholds [79].

Therefore, we used EER to compare the overall performance of a classification method. A lower value of EER suggests a better classification method. We experimented each classification method for each user and reported the results for a particular experiment as the average of all users.

To compute EER, we employed the classification score (prediction probability) generated from a particular classification algorithm. Each test sample (a touch stroke) was evaluated using the classification algorithms in this study. The score represents the algorithms' certainty in predicting a test sample belonging to either a legitimate or illegitimate user. The score is in the range of 0 and 1. The higher the value, the more confident the classifier is on predicting the class. For example, if the score for predicting the legitimate user is larger than the threshold, the test sample is assigned as belongs to the legitimate user, or otherwise, the illegitimate user. Similar to other related studies [10, 12, 11, 13], we also used an average score of multiple consecutive strokes for the authentication decision instead of the score of a single stroke. In this study, we used the average score of 10 strokes.

Even though this metric can show the performance of each classification method, it may not be sufficient to show its superiority by using the metric alone. Statistical tests are needed to show that a particular method is significantly different from others.

4.5. Statistical Tests of Significance

We further analysed the results of the experiments by examining the statistical significance of the classification methods. There are several scenarios in each dataset. For example, in Antal datasets, there are two scenarios: vertical and horizontal strokes. For each scenario of a dataset (see more in Section 5), the classification method that achieved the lowest average EER across all users in a particular scenario was ranked at 1, followed by the second lowest EER as rank 2, and so forth. In case of a tie (more than one method achieved the same EER), their ranks were averaged. Finally, the average rank for each classification method is then obtained by considering all scenarios across all datasets. Thus, the best method is the one that has the highest average rank (lowest in value) throughout different scenarios.

We employed Friedman test [80], to statistically compare the ranks of various classification methods in this study. We chose this non-parametric test because the assumptions of parametric tests tend to be violated in our comparisons (i.e. normal distribution or homogeneity of variance). It tests the null hypothesis that there is no difference between the evaluated methods. The test was conducted for multiple methods

on multiple scenarios across different datasets. The Friedman statistic was computed as follows:

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

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 $i \in D$ that is:

$$AR_j = \frac{1}{D} \sum_{i=1}^D r_i^j. \quad (5)$$

If the null hypothesis of Friedman's test is rejected ($\alpha = 0.05$), it indicates a significant difference among the evaluated methods in terms of the average ranks of EER. Then, we used the Nemenyi post-hoc test [81] to test the comparison of the pairwise comparison of the methods on multiple scenarios. The test states that the performances of two or more methods are significantly different if their average ranks differ by at least the critical difference (CD). The CD is computed as:

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

where $q_{\alpha, \infty, K}$ is the value based on the Studentized range statistic. We displayed the results of the comparison using a CD diagram [82, 83] 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.

The next section presents our results and finding based on the experiment setup in this section.

5. Results and Discussion

In this section, we present the experimental results of our study. The DS methods presented in Section 3 were evaluated across four touch-based biometric datasets on different scenarios. The scenarios for each dataset are shown in Table 5. The scenarios differ in several aspects. The first aspect is the direction of strokes. It is either vertically (scroll up or down) or horizontally (scroll to the left or right). Second, in Serwadda dataset, there are two types of screen orientations - either portrait or landscape. We followed the setup based on the original paper of each dataset by building the models separately on different directions of strokes and different screen orientations.

On top of that, for datasets that were collected in several sessions, we carried out experiments in two setups: intra-session and inter-session. A session here refers to one period where a user was instructed to start using the device and end when he or she was asked to stop using the device. The duration depends on a particular dataset. Based on the definition,

Table 5
List of scenarios across all datasets

Dataset	Scenario	Notation
Frank	Vertical strokes in intra-session	FRK ₁
	Horizontal strokes in intra-session	FRK ₂
	Vertical strokes in inter-session	FRK ₃
	Horizontal strokes in inter-session	FRK ₄
	Vertical strokes in inter-week	FRK ₅
	Horizontal strokes in inter-week	FRK ₆
Serwadda	Vertical strokes on portrait screen orientation in intra-session	SWD ₁
	Horizontal strokes on portrait screen orientation in intra-session	SWD ₂
	Vertical strokes on portrait screen orientation in inter-session	SWD ₃
	Horizontal strokes on portrait screen orientation in inter-session	SWD ₄
	Vertical strokes on landscape screen orientation in intra-session	SWD ₅
	Horizontal strokes on landscape screen orientation in intra-session	SWD ₆
	Vertical strokes on landscape screen orientation in inter-session	SWD ₇
	Horizontal strokes on landscape screen orientation in inter-session	SWD ₈
Antal	Vertical strokes	ANT ₁
	Horizontal strokes	ANT ₂
Mahbub	Combined strokes across all sessions	MHB

we carried out experiments on both intra-session and inter-session scenarios. The purpose of carrying out this kind of experiment is to investigate the performance of a particular classification method across different times. Since touch gesture is a behavioural biometric, it is worth to investigate the performance of a classifier across a different period.

For the intra-session scenario, we trained and tested a model on the dataset of the same session according to the data splitting method elaborated in Section 4.3. For inter-session, we trained the model with one session and tested it with a later session. For Frank dataset, it has several sessions in one day and another session in another week. We carried out the experiments in intra-session, inter-session, and inter-week scenarios. For Serwadda dataset, we carried out the experiments based on intra-session and inter-session as well. It is worth to note that the duration of each session and the interval from one session to another is different, for each dataset (see Table 3). For Antal dataset, it was not divided based on multiple sessions. Therefore, we were not able to separate it according to several sessions. For Mahbub dataset, the data were collected without any fixed duration for each session. Instead, the author defined one session as when a user unlocked and locked the device. Therefore, a session for a particular user is not the same as another user. Since this dataset does not have any fixed period for every user, we did not carry out the inter-session experiment for this dataset. Instead, all sessions were combined.

We compared the performance of DS methods with the single classifiers in the pool and static ensemble methods based on equal error rate (EER). The results in terms of EER are an average across different users in a particular scenario.

5.1. Comparison of Single Classifiers

In this section, we present the performance of single classifiers in the pool of classifiers. Six classifiers in the pool

of classifiers, namely, SVM, NB, DT, KNN, LR, and NN (MLP) were evaluated on the four datasets across different scenarios. These classifiers were commonly used in the domain (see Section 2.1). Therefore, before generating a pool of classifiers for the DS methods, we first evaluated the performance of these classifiers. This evaluation was carried out to achieve our first objective (RO1). Table 6 shows the average EER of each single classifier on different scenarios. The best classifier for each scenario is highlighted in bold.

For Frank dataset, the experiments were carried out on intra-session, inter-session, and inter-week scenarios. For each scenario, we modelled the user's touch strokes based on vertical and horizontal strokes separately. In the intra-session scenario, the best single classifier on vertical strokes (FRK₁) was KNN with an EER of 1.16%. For horizontal strokes (FRK₂), the best classifier was also KNN with EER of 0.94%. In the inter-session scenario, the performance of each classifier degraded for both types of strokes. This result is common in touch dataset due to the behavioural changes after some period [10, 11, 13]. The best classifier for vertical (FRK₃) and horizontal (FRK₄) strokes in inter-session scenario was NN (EER = 14.72%) and KNN (EER=3.59%), respectively. Lastly, in the inter-week scenario, the overall performance further degraded compared to the inter-session scenario. This result is expected since the model was built and evaluated with the data of two different weeks. The best classifier were KNN and NN for vertical (FRK₅) and horizontal (FRK₆) strokes, respectively, with EER of 10.08% and 13.48%.

For Serwadda dataset, there were two main scenarios: intra-session and inter-session scenarios. Compared to Frank dataset, Serwadda dataset has landscape and portrait screen orientations as a separate model for each screen orientation and directions of strokes. Therefore, there were eight scenar-

Table 6
Performance of single classifiers 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
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
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
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
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
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
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

ios altogether for this dataset. For intra-session scenario on portrait screen orientation, KNN was the best single classifier with EER of 3.54% and 2.80% for vertical (SWD₁) and horizontal (SWD₂) strokes, respectively. For intra-session scenario on landscape screen orientation, KNN was also the best classifier with EER of 4.39% and 1.39% for vertical (SWD₅) and horizontal (SWD₆) strokes, respectively. It can be seen that the performance of the classifiers on Serwadda dataset is worst than the performance of the classifiers on Frank dataset in the intra-session scenario. This result can be explained by the fact that one session in Frank dataset took around several minutes only. For inter-session scenario on portrait screen orientation, the best classifiers were SVM (EER = 22.83%) and KNN (16.01%) for vertical (SWD₃) and horizontal (SWD₄) strokes respectively. For landscape screen orientation, the best classifier was KNN for both vertical (SWD₇) and horizontal (SWD₈) strokes, with EER of 1.39% and 20.57%, respectively. In general, it can be seen that the performance of all classifiers on the inter-session scenarios of Serwadda dataset was worst than the Frank dataset. This observation could be due to the intervals between each session, where on Serwadda dataset, the interval was at least one day while in Frank dataset, just a few minutes apart. We also believe that the longer duration of a session in Serwadda dataset contributed towards the aforementioned observation.

For Antal dataset, even though it was collected in 4 weeks, it was not separated by sessions. Therefore, we can only model the user's behaviour based on all strokes during the data collection on vertical and horizontal strokes. The best classifier for both directions of strokes was KNN with EER of 2.75% and 6.75%, for vertical (ANT₁) and horizontal (ANT₂) strokes respectively. Since the dataset was not separated into several sessions, we believe that this has caused the EER to be slightly higher compared to the other datasets.

For Mahbub dataset, even though it was separated in sessions, it was not collected in such a way that all sessions performed the task in the same sessions. Instead, the sessions started and ended when a user unlocked and locked the device, respectively. Moreover, the data was collected in an unconstrained environment and without a specified direction of strokes. Therefore, we were not able to run an experiment with session-based and stroke's direction-based scenarios. The best performing classifier in this dataset (MHB) was KNN with 27.84%. It can be observed that the performance of the classifiers on this dataset is worse compared to other datasets. This result might be due to the duration of the

data collection was around one week, and there was a pre-defined task that causes the variations of users' behaviour.

In general, the performance of each classifier is not consistent across different datasets and scenarios. However, KNN outperformed the other classifiers in most cases. On the other hand, NB is the worst classifier in most cases. Besides, the performance of the classifiers on the intra-session scenario were always better than inter-session scenarios. For the datasets that have inter-session, the performance of each classifier is degraded. This result can be explained by the fact that a user's behaviour could changes over time and thus affect the performance of a classifier. We can see that the performance of all classifiers on the dataset that was collected in a longer period has worse performance. We can also see that the performance of classifiers on the dataset that was collected without any pre-defined task performed poorly. Therefore, we can conclude that the performance of a classifier in a pre-defined task performs better than those in the free task. The experiments show that the time and task have an impact on the performance of a particular classifier. In the next sections, we aim to investigate the performance of MCS on these datasets.

5.2. Comparison of Static Ensemble Methods

In this section, we aimed to analyse the performance of MCS-based static ensemble methods. This analysis was also carried out to achieve our first objective (RO1). We evaluated static ensemble methods, which include Random Forest (RF), Majority Voting (MV), Static Selection (SS) and Single Best (SB). RF is a widely used ensemble learning method in touch-based CMDA [12, 11, 19, 17]. It is a homogeneous MCS based on diversified decision trees. On the other hand, MV, SS and SB are heterogeneous MCS based on the base classifiers mentioned in Section 3.2. The purpose of the experiments in this section is to investigate whether MCS methods can improve authentication performance.

Table 7 shows the experimental results. We can see that RF outperforms the other methods in this section in most cases (9 scenarios). It is followed by SS (5 scenarios) and MV (3 scenarios). SB did not show any superiority in any scenarios. By comparing the performance of static ensemble methods with single classifiers, (Table 6), we can observe that in most scenarios, the authentication performance of the best single classifier can further be improved by at least one of the methods belonging to static ensemble methods. For example, in Mahbub dataset (MHB), the best single classifier was KNN with an EER of 27.84% reduced to 19.33%

Table 7
Performance of static ensemble 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
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
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
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
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

by RF. On the other hand, except for vertical strokes on the inter-week scenario of Frank dataset (FRK₅), the MV cannot outperform the best single classifier, which was KNN.

The results in this section show the advantages of MCS compared to single classifiers. Therefore, we can conclude that utilising MCS for touch-based CMDA can improve the authentication performance. However, the MCS methods employed in this section used the same model to perform the classification. In the next section, we investigate the performance of a different type of MCS, which is DS, where for each test sample, the method selects the most promising model to perform the classification task.

5.3. Comparison of DS Method

In this section, we first analysed the performance of DS method and compared it with other classification methods in the previous sections. Table 8 shows the results of DS methods across different datasets. In order to analyse the overall performance of DS methods, we employed the Friedman test [80] to analyse the ranking of DS methods (see Section 4.5). We chose to analyse the results based the ranking of each DS method across different scenarios since we treat each scenario independent of each other. Therefore, comparing EER across different scenarios may not be relevant. As the p-value is lower than $\alpha = 0.05$, we can reject the null hypothesis that all methods are the same. We further analysed the results by carrying out the Nemenyi post-hoc test [81] to check whether the difference in the average rank of DS methods was higher than the critical difference (CD). The performance of the two DS methods is statistically different when their difference in average rank was higher than the CD. Figure 2 shows CD diagram with the results of the Nemenyi post-hoc test. It is worth to note that the DS methods where the difference in the average ranks is lower than the CD are connected by a black bar. That bar shows the performance is statistically equivalent.

There are several interesting finding from this experiment. First, we can see that DES methods are superior when compared to DCS methods. This result can be explained by the ability of DES to aggregate the selected subset of the most competent classifiers instead of a selection of just one most competent classifier as in DCS. Selecting only one classifier may also cause the selection of a classifier with low confidence in its classification decision. This result shows that DES can benefit from the classification capability of several competent classifiers instead of one classifier. Second, the top six DS methods were DES-RRC, DES-Log, DES-MD, DES-P, DES-Exp and DES-KL. It is worth to note that these top six methods (except DES-P) were DS methods based on a probabilistic measure of competence. As such, we can conclude that probabilistic DS methods are superior in our study.

DES-MD, DES-P, DES-Exp and DES-KL. It is worth to note that these top six methods (except DES-P) were DS methods based on a probabilistic measure of competence. As such, we can conclude that probabilistic DS methods are superior in our study.

We can also see that for DCS, the probabilistic DCS methods (i.e. DCS-Priori and DCS-Posteriori) are among the top DCS methods, which rank higher than that of accuracy-based DCS methods (i.e. DCS-OLA, DCS-LCA and DCS-MLA). The superiority of probabilistic-based DS methods may be due to its ability to measure the competence level of base classifiers based on predicted class probabilities ("soft classification") instead of merely using the predicted class label, which is a "hard classification" method. We believe that this type of method can handle a high degree of overlap between the classes due to intra-user variability in touch biometric data.

Next, we also compared the performance of DS methods with other classification methods chosen in this study. To compare DS methods with other types of classification methods, we chose the top 10 DS methods in the previous experiment and compare them with other classification methods. These DS methods include DES-RRC, DES-Log, DES-MD, DES-P, DES-Exp, DES-KL, META-DES, DES-KNORAE, DES-KNORAU and DES-KNN. The rationale behind this decision is to have a better visualisation with a smaller number of methods to highlight the top performer. We carried out another rank test with these 20 methods, as shown in Figure 3. Based on the Friedman test, the p-value is lower than the significance level $\alpha = 0.05$, so we can reject the null hypothesis that these 20 methods are the same. Based on the average rank, we can observe that the top six methods were DES-RRC, DES-Log, DES-MD, DES-P, RF, and DES-Exp. Amongst these top six methods, only RF is not a DS method. However, it is still an MCS (a static ensemble method). From the figure, we can also observe that single classifiers were inferior compared to DS methods. We can further conclude that employing DS methods can provide better performance for the continuous authentication scheme..

We also analysed the best method for each scenario on all datasets. Table 9 show the best method for each scenario. It is interesting to find that, even though DES-RRC was ranked first, it was not amongst the best method in any scenario. However, RF and DES-Log seem to perform the best in many scenarios. We believe this happened due to the inconsistency of the methods. We further analysed it by plotting the ranks of top six methods across all scenarios and

Table 8
Performance of 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
DCS-Rank	1.71	2.57	16.21	4.77	12.56	14.35	4.25	3.48	24.85	16.29	4.78	2.31	22.08	11.51	3.43	7.09	28.23
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
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
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
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
DCS-MCB	1.33	1.74	15.01	3.90	12.20	15.64	3.85	2.93	22.98	16.27	3.92	1.87	19.91	10.91	3.82	6.16	27.41
DCS-MLA	3.58	3.98	16.35	6.13	16.73	13.79	6.00	5.36	23.52	16.88	7.04	3.17	20.83	10.90	5.11	10.16	29.85
DES-Clustering	1.29	1.12	15.20	3.61	11.69	11.35	3.17	2.11	22.34	14.20	3.28	1.51	21.15	8.69	3.50	5.78	24.84
DES-KNN	1.07	1.02	13.72	3.10	11.33	12.64	2.95	1.92	21.75	14.01	3.34	1.32	20.32	10.04	3.23	5.41	24.60
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
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
DES-Exp	0.96	0.57	13.47	2.89	11.46	10.54	2.68	1.80	19.57	13.71	3.24	1.28	18.83	8.35	3.05	5.04	22.63
DES-Log	1.06	0.54	13.13	2.81	10.77	11.09	2.55	1.67	20.01	13.44	2.93	1.29	19.32	8.57	3.04	4.56	23.42
DES-MD	0.98	0.55	13.45	2.90	11.22	10.47	2.68	1.78	19.78	13.69	3.23	1.26	18.83	8.34	3.01	5.04	22.65
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
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
DES-KL	0.98	0.64	13.35	2.88	11.39	10.62	2.76	1.80	19.62	13.76	3.34	1.30	18.80	8.35	3.41	5.19	22.72
DES-KNOP	1.25	0.88	13.65	2.98	12.60	12.52	2.90	2.06	23.19	14.27	3.31	1.35	19.71	9.34	3.21	5.47	23.99
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

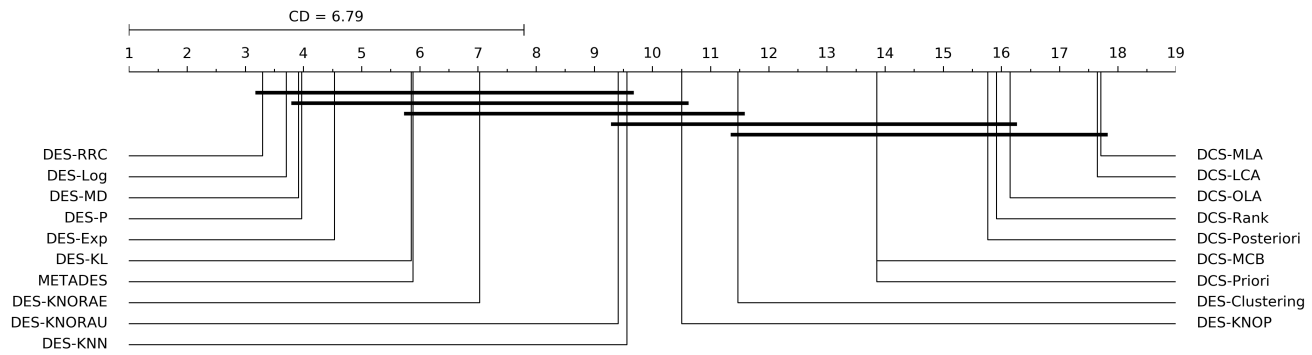


Figure 2: Average rank of DS methods. The higher the rank, the better the method.

datasets in a box plot, as shown in Figure 4. From the figure, we can see that even though RF and DES-Log performed the best in many scenarios, these two methods were less consistent compared to the other methods. Our experiment results show that DES-RRC is more consistent than RF and DES-Log. We argue that for the case of continuous mobile de-

vice authentication, a more consistent classification method is preferable as the inconsistent perform may reduce the security and usability of the scheme.

The analysis in section was carried out to achieve our second (RO2) and third objectives (RO3). The results presented in our study indicate that 1) probabilistic-based DS

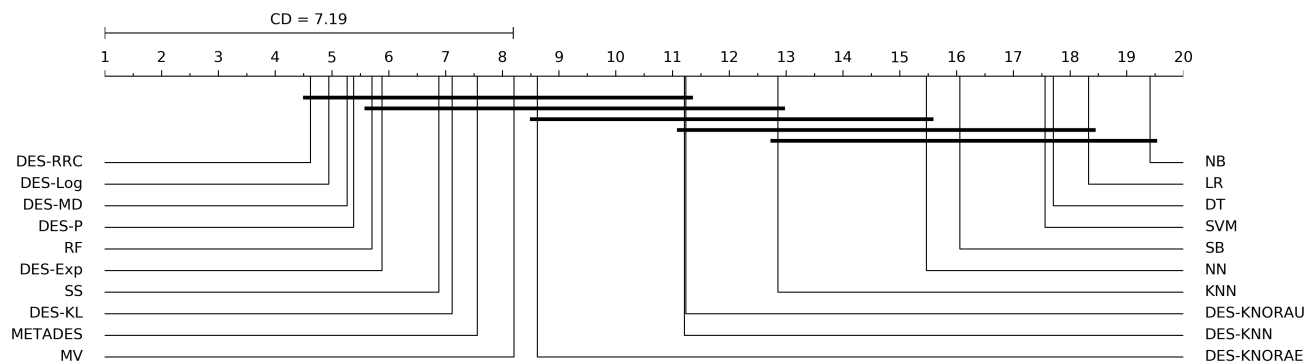


Figure 3: Average rank of DS methods for single classifiers and static ensemble methods. The higher the rank, the better the method.

Table 9
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	DES-Log	DS	13.13
	FRK ₄	DES-KNORAE	DS	2.32
	FRK ₅	KNN	Single classifier	10.08
	FRK ₆	RF	Static ensemble	9.58
Serwadda	SWD ₁	DES-Log	DS	2.55
	SWD ₂	DES-Log	DS	1.67
	SWD ₃	DES-P	DS	19.51
	SWD ₄	DES-Log	DS	13.44
	SWD ₅	RF	Static ensemble	2.90
	SWD ₆	SS	Static ensemble	1.14
	SWD ₇	RF	Static ensemble	18.11
	SWD ₈	RF	Static ensemble	8.10
Antal	ANT ₁	RF	Static ensemble	2.16
	ANT ₂	DES-Log	DS	4.56
Mahbub	MHB	RF	Static ensemble	19.33

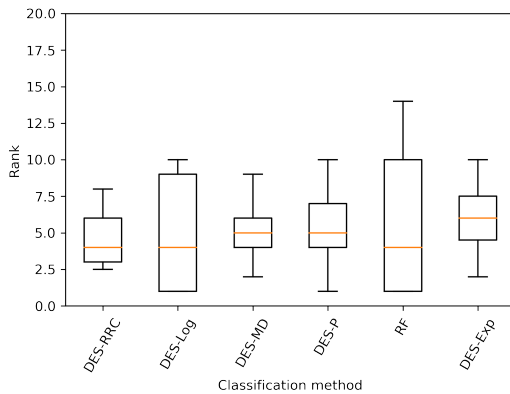


Figure 4: Box plot of the average rank of top six methods

methods are superior amongst DS methods, 2) DS methods are better than single classification, and 3) DES-RRC is more consistent compared to other top six classification methods. To further understand the efficiency of DS methods, we perform follow-up experiments by analysing the computational time of all classification methods discussed in the next section.

5.4. Comparison of Computational Time

We carried out an additional analysis to compare the computational time of all classification methods used in this study. The computational time here refers to the training and testing of each method. Computational time is a crucial consideration in continuous mobile device authentication. An ideal continuous authentication is expected to be able to lockout an illegitimate user quickly with high detection accuracy.

Table 10 shows the computational time of each methods in seconds. It is worth to note that, for MCS methods,

the computational time includes the training of each base classifiers in the pool. Besides, Figure 5 shows the average ranks of computational time across different scenarios on all datasets. The higher the rank, the lower the computational time.

From the figure, we discovered several findings. First, the computational time of single classifiers has a higher rank than MCS methods. Except for RF, which is ranked higher than DT, other MCS methods have lower rank for computational time. This observation can be explained by the tuning of various hyperparameters in DT using a grid search method, which may lead to a higher computational time. Since RF is the only homogeneous MCS, we believe that this is why it has a higher rank, where the base classifiers of RF did not perform parameter tuning. On the hand, other MCS methods are based on a pool of heterogeneous classifiers, which include the training of different types of classifiers in the pool.

Besides, amongst the MCS methods (static ensemble methods and DS), DS methods in general have a lower rank in computational time. This result can be explained by more time needed to compute the region of competence in DS methods. The distance between each test sample and all samples in the validation set has to be computed. Even though DS methods and most MCS methods require more computational time than single classifiers, they have a better performance in term of EER.

The complexity of a DS method generally includes three major operations: (1) pool of classifiers generation, (2) selecting the most competent classifier or a subset of the most competent classifiers, and (3) aggregating the subset of the selected classifiers (if applicable). Suppose there are m classifiers in the pool. Each base classifier is trained separately using n training samples. The first operation generates a pool

Table 10
Computational time of single classifiers, static ensemble methods and DS methods on all datasets (seconds)

	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
SVM	0.1710	0.1910	0.1680	0.1729	0.1826	0.1674	0.1828	0.1722	0.1705	0.1477	0.1449	0.1416	0.1426	0.1425	0.1450	0.1455	0.2873
NB	0.0018	0.0017	0.0018	0.0017	0.0017	0.0015	0.0018	0.0017	0.0023	0.0017	0.0021	0.0016	0.0020	0.0015	0.0021	0.0016	0.0222
DT	2.1296	2.3573	2.1022	2.1568	2.1381	1.9709	2.1106	2.1604	2.0465	1.8374	1.7327	1.7542	1.7116	1.7654	1.7189	1.7447	2.3827
KNN	0.1490	0.1579	0.1532	0.1549	0.1396	0.1345	0.1443	0.1546	0.1475	0.1275	0.1278	0.1236	0.1249	0.1260	0.1230	0.1216	0.2361
LR	0.0073	0.0082	0.0064	0.0073	0.0069	0.0068	0.0064	0.0067	0.0077	0.0055	0.0079	0.0054	0.0070	0.0054	0.0066	0.0053	0.0098
NN	0.0886	0.1783	0.0917	0.1319	0.1438	0.1837	0.1105	0.1615	0.1747	0.2223	0.1738	0.2520	0.1510	0.1513	0.1592	0.1386	0.6496
RF	0.2344	0.2511	0.2296	0.2333	0.2295	0.2139	0.2242	0.2460	0.2241	0.1918	0.1897	0.1842	0.1881	0.1855	0.1913	0.1829	0.4064
MV	2.2676	2.4630	2.3052	2.4151	2.4035	2.5080	2.1703	2.2101	2.4113	2.7671	2.3365	2.7208	2.4298	2.3502	2.4020	2.3353	3.2056
SS	2.5532	2.9018	2.5286	2.6322	2.6182	2.4713	2.5620	2.6609	2.5503	2.3474	2.1914	2.2830	2.1420	2.1983	2.1579	2.1632	3.4830
SB	2.5486	2.8987	2.5251	2.6281	2.6135	2.4669	2.5569	2.6565	2.5442	2.3426	2.1834	2.2785	2.1360	2.1930	2.1509	2.1583	3.3974
DCS-Rank	2.6753	3.0238	2.6505	2.7548	2.7380	2.5892	2.6851	2.7830	2.6789	2.4703	2.3229	2.4087	2.2714	2.3196	2.2861	2.2864	3.7811
DCS-OLA	2.6735	3.0230	2.6501	2.7520	2.7374	2.5896	2.6799	2.7842	2.6784	2.4697	2.3206	2.4073	2.2698	2.3199	2.2855	2.2855	3.7715
DCS-LCA	2.6753	3.0241	2.6515	2.7536	2.7410	2.5911	2.6803	2.7866	2.6804	2.4709	2.3219	2.4087	2.2725	2.3207	2.2880	2.2861	3.8095
DCS-Priori	2.6846	3.0324	2.6645	2.7655	2.7473	2.6034	2.6938	2.7977	2.7015	2.4825	2.3579	2.4221	2.2955	2.3299	2.3137	2.2982	4.2931
DCS-Posteriori	2.6849	3.0333	2.6647	2.7670	2.7458	2.6038	2.6874	2.8011	2.7045	2.4839	2.3614	2.4255	2.2970	2.3320	2.3190	2.2985	4.3890
DCS-MCB	2.6938	3.0407	2.6783	2.7792	2.7649	2.6125	2.6967	2.8174	2.7357	2.4953	2.4022	2.4442	2.3246	2.3368	2.3569	2.3066	5.2208
DCS-MLA	2.6757	3.0247	2.6530	2.7548	2.7355	2.5921	2.6830	2.7859	2.6805	2.4715	2.3251	2.4091	2.2729	2.3211	2.2896	2.2865	3.8619
DES-Clustering	2.7073	3.0589	2.6851	2.7866	2.7668	2.6182	2.7120	2.8149	2.7123	2.5099	2.3561	2.4474	2.3078	2.3562	2.3197	2.3311	3.8199
DES-KNN	2.7235	3.0732	2.7172	2.8087	2.7919	2.6545	2.7208	2.8775	2.8259	2.5340	2.5110	2.4916	2.4037	2.3665	2.4470	2.3496	8.3063
DES-KNORAE	2.6747	3.0238	2.6515	2.7542	2.7420	2.5902	2.6804	2.7853	2.6814	2.4723	2.3276	2.4107	2.2748	2.3231	2.2932	2.2882	3.8004
DES-KNORAU	2.6747	3.0266	2.6502	2.7533	2.7370	2.5900	2.6816	2.7824	2.6798	2.4716	2.3254	2.4065	2.2717	2.3196	2.2904	2.2863	3.8020
DES-Exp	2.6776	3.0274	2.6554	2.7584	2.7425	2.5944	2.6837	2.7869	2.6835	2.4761	2.3273	2.4106	2.2747	2.3248	2.2911	2.2898	3.8075
DES-Log	2.6795	3.0281	2.6563	2.7600	2.7409	2.5936	2.6845	2.7869	2.6846	2.4751	2.3295	2.4118	2.2764	2.3247	2.2929	2.2910	3.8063
DES-MD	2.6796	3.0270	2.6568	2.7570	2.7417	2.5944	2.6826	2.7863	2.6843	2.4748	2.3277	2.4129	2.2765	2.3260	2.2934	2.2900	3.8140
DES-RRC	2.7586	3.1083	2.7316	2.8362	2.8324	2.6742	2.7653	2.8613	2.7700	2.5512	2.4191	2.4950	2.3605	2.4049	2.3880	2.3702	3.9019
DES-P	2.6765	3.0244	2.6518	2.7532	2.7390	2.5919	2.6807	2.7840	2.6802	2.4714	2.3252	2.4090	2.2748	2.3206	2.2907	2.2870	3.8032
DES-KL	2.6793	3.0291	2.6561	2.7552	2.7398	2.5952	2.6866	2.7859	2.6841	2.4761	2.3302	2.4137	2.2765	2.3247	2.2959	2.2902	3.8121
DES-KNOP	2.7930	3.1434	2.7704	2.8717	2.8497	2.7059	2.8000	2.9028	2.7982	2.5881	2.4478	2.5248	2.3908	2.4384	2.4076	2.4020	3.9291
META-DES	3.0298	3.3825	3.0047	3.1111	3.0893	2.9389	3.0369	3.1387	3.0355	2.8283	2.6842	2.7624	2.6282	2.6770	2.6469	2.6418	4.3253

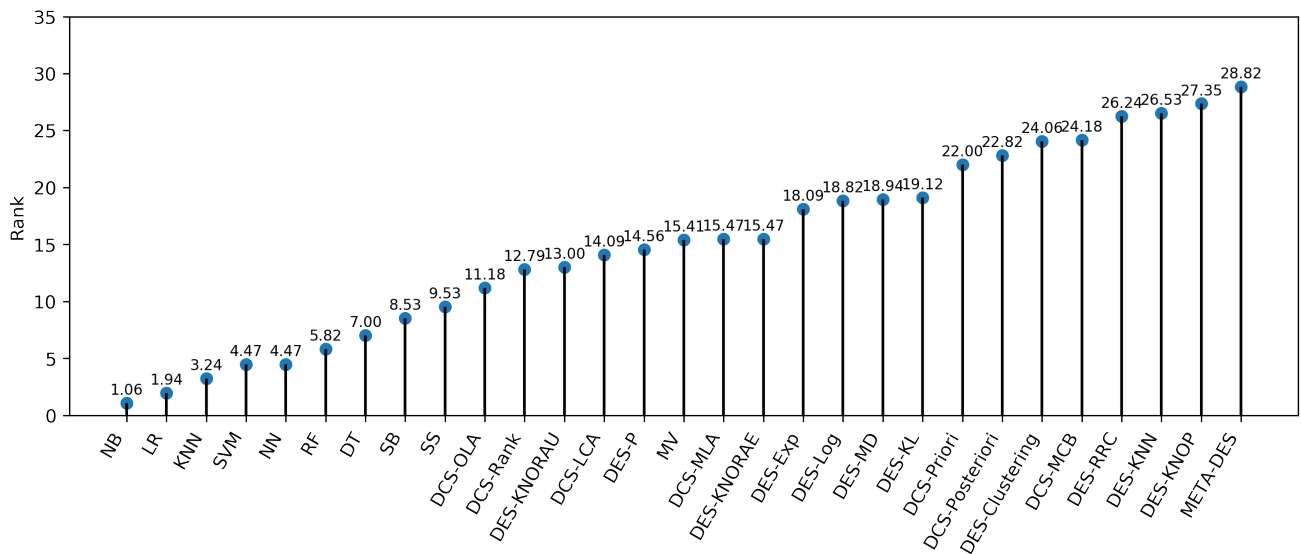


Figure 5: Average rank of the computational time of DS methods, single classifiers, and static ensemble methods. The higher the rank, the lower the computational time of the method.

of base classifiers C . The time complexity can be represented by $O(nC)$, where C is a function with respect to the training of m classifiers using a training set of size n . Since each base classifier can be trained independently, this operation can be processed in parallel. The time complexity of a DS method is generally more focused on the second and the third operations. The operations are affected by defining the region of competence, estimating the competence level of each base classifier, and the dynamic selection and clas-

sification process of each test sample. The estimation of the competence level of each base classifier involves the evaluation of the validation set and defining the region of competence with time complexity $O(R)$, where R is a function with respect to the region of competence definition using the validation set of size n . Thus, the total time complexity for the whole operation of a DS method is approximately represented by $O(nC + R)$. Therefore, it can be seen that the time complexity of a DS method is approximately linear with re-

spect to the number of samples.

To summarise, we found that the computational cost of DS methods is generally higher than single classifiers and static ensemble methods. On the other hand, in terms of EER and consistency, the performance of DS methods is better than single classifiers and static ensemble methods. While the computational cost is higher for DS methods, a slight improvement of the EER of the continuous authentication scheme would significantly reduce the possibility of allowing illegitimate users to access the device. We believe this limitation can be overcome by the improvement of the computational capabilities of mobile devices. In addition, one possible way to deal with the limited computational resources of mobile devices is by training the models on cloud servers and only using the mobile device for authentication task.

5.5. Threats to Validity

Since this work is an empirical study, it is crucial for us to be aware of any potential threats to the validity of the results obtained and the conclusions derived from this study. Therefore, in this section, we discuss the threats to the internal and external validity in our study. It is worth to note that the validity discussed in this section are primarily in the context of experimental design.

Threats to experimental bias can be a potential risk, in terms of datasets collection and classification methods. The mobile device users involved in the public datasets used in our study might not represent the whole population of mobile device users. To overcome this threat, we chose to use open datasets published in the literature. Several studies other than the original authors of the datasets have employed at least one of datasets used in our study as well [13, 84, 85, 86]. We believe the choice of datasets is appropriate as it involves the swipe gesture, which is commonly used touch gestures on touchscreen-based mobile devices.

Furthermore, the choice of classification methods is another possible source of bias. We are aware that there are various classification algorithms, which we did not consider in our study. Our consideration is motivated by the aim of finding an empirical comparison between DS methods and other types of classification methods (single classifiers and static ensemble methods). Some of the classification methods other than DS methods can be found in the domain of touch-based CMDA (as shown in Section 2) and other machine learning literature. We believe that the representativeness of each types of classification methods are included.

It is worth to note that user classification is one of the phases in the whole authentication process. Other related phases such as data preprocessing, feature extraction, and feature selection may improve the classification results. There are various methods for this phase. Therefore, a thorough evaluation of several related methods may incur additional computational effort in the whole study since we considered a large number of classification methods. Since our focus is only on the classification methods, we view these additional steps as not necessary in the study.

The threat to external validity relates to the generalisability of our results. The sampling procedure might also produce bias to the results. We considered all users in each scenario that met our requirements for the overall results. We also considered a randomly chosen touch sample for each user, meant to be used for the training of each classification method. Therefore, the sampling method can ensure a particular classification method will produce results based on experiments done on multiple users with randomly chosen training samples.

6. Conclusions and Future Work

In this study, we presented a framework for user classification in touch-based continuous mobile device authentication (CMDA) based on dynamic selection method (DS). The method is based on the multiple classifier systems (MCS), where for each test sample, the most competent classifier or a subset of the most competent classifiers will be selected to perform the classification task. The framework consists of three stages, which include pool of classifiers generation, classifiers selection, and classifiers aggregation. The proposed framework performs the selection of the most competent classifier or a subset of the most competent classifiers from a pool of classifiers to perform the classification task for a test sample. The method estimates the competence level of each base classifier in the pool. The competence level is estimated using a measure of competence in the local region in the feature space where the test sample is located. Based on this region, the classifier or a subset of classifiers that achieved the highest competence level will be selected and aggregated (if a subset of classifiers was selected) to perform the classification task for a particular test sample. Several DS methods were evaluated on four touch-based biometric datasets.

We first evaluated the classification performance of single classifiers and static ensemble methods, to be compared with DS methods. Experimental results show that DS methods are capable of producing promising results with relatively low EER in many scenarios of the datasets, with relatively high consistency. Notably, a DS method, namely DES-RRC, performed more consistently compared to other top performing methods in term of the rankings of EER across different scenarios. Our results show that there are potential improvements of authentication performance when employing DS methods, compared to single classifiers and static ensemble methods. For example, the performance of DES-RRC, achieved an EER of 0.55% compared to KNN (single classifier) with an EER of 0.94% and RF (static ensemble method) with an EER of 0.84% in of the experimental scenarios. When the authentication error can be further reduced, the results can provide a better security mechanism for mobile devices by ensuring a higher security while maintaining the usability at an acceptable level. In other words, a mobile device can detect illegitimate users better and block them from using the device. Moreover, the authentication mechanism of the mobile device can avoid locking out legit-

imate users during their normal usage session, which is more convenient for the user. Therefore, we suggest to employ DS methods for the classification of users in touch-based CMDA due to its better classification capability.

Besides, another MCS method, which is RF, has also shown to be the best performer in many scenarios. However, based on the ranking, it was less consistent. Therefore, our findings suggest that DS method is better in general, with more consistent performance. However, realising the classification ability of RF, including RF in the pool of classifiers, and applying DS method can also be one of our future works. Besides, other future works include formulating a measure of competence that can handle the high intra-class variability of touch-based biometric datasets and a better way of defining the region of competence of a particular test sample to improve the authentication results. On top that, we would also like to analyse the proposed framework using other performance metrics to see the performance from different perspectives.

CRedit authorship contribution statement

Ahmad Zairi Zaidi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Visualization. **Chun Yong Chong:** Writing - Review & Editing, Supervision. **Rajendran Parthiban:** Supervision. **Ali Safaa Sadiq:** Writing - Review & Editing

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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

References

- [1] W. Xu, Y. Shen, C. Luo, J. Li, W. Li, A. Y. Zomaya, Gait-Watch: A Gait-based context-aware authentication system for smart watch via sparse coding, *Ad Hoc Networks* 107 (2020) 102218. doi:10.1016/j.adhoc.2020.102218. URL <https://linkinghub.elsevier.com/retrieve/pii/S1570870520300287>
- [2] A. A. Khan, V. Kumar, M. Ahmad, B. B. Gupta, M. Ahmad, A. A. Abd El-Latif, A secure and efficient key agreement framework for critical energy infrastructure using mobile device, *Telecommunication Systems* 78 (4) (2021) 539–557. doi:10.1007/s11235-021-00826-6. URL <https://link.springer.com/article/10.1007/s11235-021-00826-6>
- [3] W.-Z. Zhang, I. A. Elgendy, M. Hammad, A. M. Ilyyasu, X. Du, M. Guizani, A. A. El-Latif, Secure and Optimized Load Balancing for Multitier IoT and Edge-Cloud Computing Systems, *IEEE Internet of Things Journal* 8 (10) (2021) 8119–8132. doi:10.1109/JIOT.2020.3042433. URL <https://ieeexplore.ieee.org/document/9279239/>
- [4] N. L. Clarke, S. M. Furnell, Authentication of users on mobile telephones - A survey of attitudes and practices, *Computers and Security* 24 (7) (2005) 519–527. doi:10.1016/j.cose.2005.08.003.
- [5] F. Tari, A. A. Ozok, 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, ACM Press, New York, New York, USA, 2006, p. 56. doi:10.1145/1143120.1143128. URL <http://portal.acm.org/citation.cfm?doid=1143120.1143128>
- [6] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, J. M. Smith, Smudge Attacks on Smartphone Touch Screens, in: *Proceeding in WOOT'10 Proceedings of the 4th USENIX conference on Offensive technologies.*, USENIX Association, Berkeley, CA, USA, 2010, pp. 1–7. URL https://www.usenix.org/legacy/event/woot10/tech/full_papers/Aviv.pdf
- [7] W. Meng, D. D. S. Wong, S. Furnell, J. Zhou, Surveying the Development of Biometric User Authentication on Mobile Phones, *IEEE Communications Surveys & Tutorials* 17 (3) (2015) 1268–1293. doi:10.1109/COMST.2014.2386915. URL <http://ieeexplore.ieee.org/document/7000543/>
- [8] V. M. V. M. Patel, R. Chellappa, D. Chandra, B. Barbello, Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges, *IEEE Signal Processing Magazine* 33 (4) (2016) 49–61. doi:10.1109/MSP.2016.2555335. URL <http://ieeexplore.ieee.org/document/7503170/>
- [9] A. Z. Zaidi, C. Y. Chong, Z. Jin, R. Parthiban, A. S. Sadiq, Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities, *Journal of Network and Computer Applications* 191 (2021) 103162. doi:10.1016/j.jnca.2021.103162. URL <https://linkinghub.elsevier.com/retrieve/pii/S1084804521001740>
- [10] M. Frank, R. Biedert, E. Ma, I. Martinovic, D. Song, Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication, *IEEE Transactions on Information Forensics and Security* 8 (1) (2013) 136–148. arXiv:1207.6231, doi:10.1109/TIFS.2012.2225048. URL <http://ieeexplore.ieee.org/document/6331527/>
- [11] C. Shen, Y. Zhang, X. Guan, R. A. Maxion, Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication, *IEEE Transactions on Information Forensics and Security* 11 (3) (2015) 1–1. doi:10.1109/TIFS.2015.2503258. URL <http://ieeexplore.ieee.org/document/7335628/>
- [12] A. Serwadda, V. Phoha, 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. doi:10.1109/BTAS.2013.6712758.
- [13] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, A. Morales, Benchmarking Touchscreen Biometrics for Mobile Authentication, *IEEE Transactions on Information Forensics and Security* 13 (11) (2018) 2720–2733. doi:10.1109/TIFS.2018.2833042. URL <https://ieeexplore.ieee.org/document/8353868/>
- [14] W. Meng, Y. Y. Wang, D. S. D. Wong, S. Wen, Y. Xiang, TouchWB: Touch behavioral user authentication based on web browsing on smartphones, *Journal of Network and Computer Applications* 117 (2018) 1–9. doi:10.1016/j.jnca.2018.05.010. URL <https://linkinghub.elsevier.com/retrieve/pii/S1084804518301723>
- [15] Y. Yang, B. Guo, Z. Wang, M. Li, Z. Yu, X. Zhou, BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics, *Ad Hoc Networks* 84 (2018) 9–18. doi:10.1016/j.adhoc.2018.09.015. URL <https://www.sciencedirect.com/science/article/abs/pii/S1570870518306899>
- [16] W. Meng, W. Li, D. S. Wong, Enhancing touch behavioral authentication via cost-based intelligent mechanism on smartphones (dec 2018). doi:10.1007/s11042-018-6094-2. URL <http://link.springer.com/10.1007/s11042-018-6094-2>

- [17] Z. Syed, J. Helmick, S. Banerjee, 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* 149 (2019) 158–173. doi:10.1016/j.jss.2018.11.017. URL <https://linkinghub.elsevier.com/retrieve/pii/S0164121218302516>
- [18] L. Li, X. Zhao, G. Xue, Unobservable re-authentication for smartphones, *NDSS - Network and Distributed System Security Symposium* (2013) 1–16. URL <https://optimization.asu.edu/papers/XUE-CNF-2013-NDSS.pdf><http://internet.society.org/doc/unobservable-re-authentication-smartphones>
- [19] U. Mahbub, S. Sarkar, V. M. Patel, 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, IEEE, 2016*, pp. 1–8. arXiv:1610.07930, doi:10.1109/BTAS.2016.7791155. URL <http://ieeexplore.ieee.org/document/7791155/>
- [20] H. Xu, Y. Zhou, M. R. Lyu, Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones, in: *Symposium on Usable Privacy and Security (SOUPS 2014)*, ACM Press, New York, New York, USA, 2014, pp. 187–198. doi:10.1145/3098243.3098244. URL <http://dl.acm.org/citation.cfm?doid=3098243.3098244><https://www.usenix.org/system/files/conference/soups2014/soups14-paper-xu.pdf>
- [21] T. Feng, J. Yang, Z. Yan, E. M. Tapia, W. Shi, TIPS: context-aware implicit user identification using touch screen in uncontrolled environments, in: *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications - HotMobile '14*, ACM Press, New York, New York, USA, 2014, pp. 1–6. doi:10.1145/2565585.2565592. URL <http://dl.acm.org/citation.cfm?doid=2565585.2565592>
- [22] D. Wolpert, W. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation* 1 (1) (1997) 67–82. doi:10.1109/4235.585893. URL <http://ieeexplore.ieee.org/document/585893/>
- [23] T. T. Nguyen, A. V. Luong, M. T. Dang, A. W.-C. Liew, J. McCall, Ensemble Selection based on Classifier Prediction Confidence, *Pattern Recognition* 100 (2020) 107104. doi:10.1016/j.patcog.2019.107104. URL <https://linkinghub.elsevier.com/retrieve/pii/S0031320319304054>
- [24] S. Lessmann, B. Baesens, H. V. Seow, L. C. Thomas, Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, *European Journal of Operational Research* 247 (1) (2015) 124–136. doi:10.1016/j.ejor.2015.05.030. URL <https://www.sciencedirect.com/science/article/abs/pii/S0377221715004208>
- [25] T. Woloszynski, P. Podsiadlo, G. Stachowiak, 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* 226 (11) (2012) 887–894. doi:10.1177/0954411912456650. URL <http://journals.sagepub.com/doi/10.1177/0954411912456650>
- [26] V. Sharma, 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. doi:10.1145/3098243.3098262. URL <http://dl.acm.org/citation.cfm?doid=3098243.3098262>
- [27] Y. Xia, C. Liu, B. Da, F. Xie, A novel heterogeneous ensemble credit scoring model based on bstacking approach, *Expert Systems with Applications* 93 (2018) 182–199. doi:10.1016/j.eswa.2017.10.022. URL <https://linkinghub.elsevier.com/retrieve/pii/S0957417417306966>
- [28] M. Sabourin, A. Mitiche, D. Thomas, G. Nagy, Classifier combination for hand-printed digit recognition, in: *Proceedings of 2nd International Conference on Document Analysis and Recognition (IC-DAR '93)*, IEEE Comput. Soc. Press, 1993, pp. 163–166. doi:10.1109/ICDAR.1993.395758. URL <http://ieeexplore.ieee.org/document/395758/>
- [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* 40 (4) (2002) 801–813. doi:10.1109/TGRS.2002.1006354. URL <http://ieeexplore.ieee.org/document/1006354/>
- [30] X. Feng, Z. Xiao, B. Zhong, J. Qiu, Y. Dong, Dynamic ensemble classification for credit scoring using soft probability, *Applied Soft Computing* 65 (2018) 139–151. doi:10.1016/j.asoc.2018.01.021. URL <https://linkinghub.elsevier.com/retrieve/pii/S1568494618300279>
- [31] L. Melo Junior, F. Maria Nardini, C. Renso, R. Trani, J. Antonio Macedo, A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems, *Expert Systems with Applications* (2020) 113351. doi:10.1016/j.eswa.2020.113351. URL <https://linkinghub.elsevier.com/retrieve/pii/S0957417420301767>
- [32] Y. Xia, J. Zhao, L. He, Y. Li, M. Niu, A novel tree-based dynamic heterogeneous ensemble method for credit scoring, *Expert Systems with Applications* 159 (2020) 113615. doi:10.1016/j.eswa.2020.113615. URL <https://linkinghub.elsevier.com/retrieve/pii/S0957417420304395>
- [33] B. Wang, Z. Mao, Outlier detection based on a dynamic ensemble model: Applied to process monitoring, *Information Fusion* 51 (2019) 244–258. doi:10.1016/j.inffus.2019.02.006. URL <https://www.sciencedirect.com/science/article/pii/S1566253518303282?via=ihj>
- [34] L. Batista, E. Granger, R. Sabourin, Dynamic selection of generative-discriminative ensembles for off-line signature verification, *Pattern Recognition* 45 (4) (2012) 1326–1340. doi:10.1016/j.patcog.2011.10.011. URL <https://www.sciencedirect.com/science/article/abs/pii/S0031320311004353>www.elsevier.com/locate/pr
- [35] S. Bashbaghi, E. Granger, R. Sabourin, G.-A. Bilodeau, Dynamic ensembles of exemplar-SVMs for still-to-video face recognition, *Pattern Recognition* 69 (2017) 61–81. doi:10.1016/j.patcog.2017.04.014. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019480409&doi=10.1016%2Fj.patcog.2017.04.014&partnerID=40&md5=b14b38aa7f50c50777380d59f8358167>
- [36] P. Porwik, R. Doroz, K. Wrobel, An ensemble learning approach to lip-based biometric verification, with a dynamic selection of classifiers, *Expert Systems with Applications* 115 (2019) 673–683. doi:10.1016/j.eswa.2018.08.037. URL <https://www.sciencedirect.com/science/article/pii/S0957417418305529?via=ihj>
- [37] M. Martinez-Diaz, J. Fierrez, R. P. Krish, J. Galbally, Mobile signature verification: Feature robustness and performance comparison, *IET Biometrics* 3 (4) (2014) 267–277. doi:10.1049/iet-bmt.2013.0081.
- [38] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, *Information Fusion* 16 (1) (2014) 3–17. doi:10.1016/j.inffus.2013.04.006. URL <https://www.sciencedirect.com/science/article/pii/S156625351300047X>
- [39] J. A. S. Lustosa Filho, A. M. Canuto, R. H. N. Santiago, Investigating the impact of selection criteria in dynamic ensemble selection methods, *Expert Systems with Applications* 106 (2018) 141–153. doi:10.1016/j.eswa.2018.04.002. URL <https://linkinghub.elsevier.com/retrieve/pii/S0957417418302264>
- [40] M. Ala'raj, M. F. Abbod, A new hybrid ensemble credit scoring model based on classifiers consensus system approach, *Expert Systems with Applications* 64 (2016) 36–55. doi:10.1016/j.eswa.2016.07.017. URL <https://www.sciencedirect.com/science/article/pii/S0957417416303621>

- [41] A. S. Britto, R. Sabourin, L. E. S. Oliveira, Dynamic selection of classifiers - A comprehensive review, *Pattern Recognition* 47 (11) (2014) 3665–3680. doi:10.1016/j.patcog.2014.05.003.
URL <http://dx.doi.org/10.1016/j.patcog.2014.05.003>
- [42] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, Dynamic classifier selection: Recent advances and perspectives, *Information Fusion* 41 (2018) 195–216. doi:10.1016/j.inffus.2017.09.010.
URL <https://linkinghub.elsevier.com/retrieve/pii/S1566253517304074>
- [43] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, *Pattern Recognition* 44 (10-11) (2011) 2656–2668. doi:10.1016/j.patcog.2011.03.020.
- [44] K. Woods, W. Philip Kegelmeyer, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (4) (1997) 405–410. doi:10.1109/34.588027.
URL <http://ieeexplore.ieee.org/document/588027/>
- [45] A. Nabih, 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, 2012*, pp. 308–313. doi:10.1109/ICMCS.2012.6320200.
URL <http://ieeexplore.ieee.org/document/6320200/>
- [46] M. C. Groccia, R. Guido, D. Conforti, Multi-Classifer Approaches for Supporting Clinical Decision Making, *Symmetry* 12 (5) (2020) 699. doi:10.3390/sym12050699.
URL <https://www.mdpi.com/2073-8994/12/5/699>
- [47] G. Giacinto, 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. doi:10.1109/ICIAP.1999.797670.
URL <http://ieeexplore.ieee.org/document/797670/>
- [48] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behaviour, *Pattern Recognition* 34 (9) (2001) 1879–1881. doi:10.1016/S0031-3203(00)00150-3.
URL <https://www.sciencedirect.com/science/article/abs/pii/S0031320300001503>
- [49] R. Soares, A. Santana, A. Canuto, M. de Souto, Using Accuracy and Diversity to Select Classifiers to Build Ensembles, in: *The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006*, pp. 1310–1316. doi:10.1109/IJCNN.2006.246844.
URL <http://ieeexplore.ieee.org/document/1716255/>
- [50] M. C. P. de Souto, R. G. F. Soares, A. Santana, 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)*, IEEE, Department of Informatics and Applied Mathematics, Fed. Univ. of Rio Grande do Norte, Natal, Brazil, 2008, pp. 1480–1487. doi:10.1109/IJCNN.2008.4633992.
URL <http://ieeexplore.ieee.org/document/4633992/>
- [51] A. H. Ko, R. Sabourin, A. S. Britto, From dynamic classifier selection to dynamic ensemble selection, *Pattern Recognition* 41 (5) (2008) 1718–1731. doi:10.1016/j.patcog.2007.10.015.
URL http://link.springer.com/10.1007/978-3-642-04146-4_106
- [52] T. Woloszynski, 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, Vol. 5716 LNCS, Chair of Systems and Computer Networks, Wroclaw University of Technology, Wyb. Wyspińskiego 27, Wroclaw 50-370, Poland, 2009*, pp. 995–1004. doi:10.1007/978-3-642-04146-4_106.
URL http://link.springer.com/10.1007/978-3-642-04146-4_106
- [53] B. Antosik, M. Kurzynski, New Measures of Classifier Competence - Heuristics and Application to the Design of Multiple Classifier Systems, in: *Advances in Intelligent and Soft Computing, Vol. 95, Springer Verlag, 2011*, pp. 197–206. doi:10.1007/978-3-642-20320-6_21.
URL http://link.springer.com/10.1007/978-3-642-20320-6_21
- [54] P. R. Cavalin, R. Sabourin, C. Y. Suen, Dynamic selection approaches for multiple classifier systems, *Neural Computing and Applications* 22 (3-4) (2013) 673–688. doi:10.1007/s00521-011-0737-9.
URL <http://link.springer.com/10.1007/s00521-011-0737-9>
- [55] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, T. Ing Ren, META-DES: A dynamic ensemble selection framework using meta-learning, *Pattern Recognition* 48 (5) (2015) 1925–1935. doi:10.1016/j.patcog.2014.12.003.
URL <https://www.sciencedirect.com/science/article/pii/S0031320314004919>
- [56] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine Learning* doi:10.1023/A:1022859003006.
- [57] R. Lysiak, M. Kurzynski, T. Woloszynski, Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers, *Neurocomputing* 126 (2014) 29–35. doi:10.1016/j.neucom.2013.01.052.
URL <https://linkinghub.elsevier.com/retrieve/pii/S092523121300698X>
- [58] T. M. Cover, P. E. Hart, Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory* 13 (1) (1967) 21 – 27. doi:10.1109/TIT.1967.1053964.
- [59] C. Cortes, V. Vapnik, Support-Vector Networks, *Machine Learning* 20 (3) (1995) 273–297. doi:10.1023/A:1022627411411.
- [60] V. Vapnik, *The nature of statistical learning theory*, Springer science & business media, New York, 2013.
- [61] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees, *Group* 37 (15) (1984) 237–251.
- [62] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification 2nd ed*, 2000.
- [63] H. Zhang, The optimality of Naive Bayes, in: *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004, 2004*.
- [64] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, Elsevier, 2011. doi:10.1016/C2009-0-19715-5.
URL <https://linkinghub.elsevier.com/retrieve/pii/C20090197155>
- [65] L. Zhou, J. K. Burgoon, D. P. Twitchell, T. Qin, J. F. Nunamaker, A comparison of classification methods for predicting deception in computer-mediated communication, *Journal of Management Information Systems* 20 (4) (2004) 139–166.
- [66] B. Wang, Z. Mao, A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule, *Information Fusion* 63 (2020) 30–40. doi:10.1016/j.inffus.2020.05.001.
URL <https://linkinghub.elsevier.com/retrieve/pii/S1566253520302645>
- [67] L. A. Rastrigin, R. H. Erenstein, *Method of collective recognition*, Energoizdat, Moscow 595.
- [68] Y. S. Huang, C. Y. Suen, A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals, *IEEE Transactions on Pattern Analysis and Machine Intelligence* doi:10.1109/34.368145.
- [69] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, *Image and Vision Computing* doi:10.1016/S0262-8856(01)00045-2.
- [70] L. I. Kuncheva, A theoretical study on six classifier fusion strategies, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2) (2002) 281–286. doi:10.1109/34.982906.
- [71] 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, É. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (Oct) (2011) 2825–2830. arXiv:1201.0490.
- [72] R. M. O. Cruz, L. G. Hafemann, R. Sabourin, G. D. C. Cavalcanti, DESlib: A Dynamic ensemble selection library in Python (feb 2018). arXiv:1802.04967.
URL <http://arxiv.org/abs/1802.04967>
- [73] M. Antal, Z. Bokor, L. Z. Szabó, Information revealed from scrolling interactions on mobile devices, *Pattern Recognition Letters* 56 (2015)

- 7–13. doi:10.1016/j.patrec.2015.01.011.
 URL <https://www.sciencedirect.com/science/article/pii/S0167865515000355?via=ihubhttps://linkinghub.elsevier.com/retrieve/pii/S0167865515000355>
- [74] Leo Breiman, Random Forests, *Machine Learning* 45 (1) (2001) 5–32. arXiv:dx.doi.org/10.1023/A:1010933404324, doi:10.1023/A:1010933404324.
 URL <http://link.springer.com/10.1023/A:1010933404324>
- [75] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (3) (1998) 226–239. doi:10.1109/34.667881.
 URL <http://ieeexplore.ieee.org/document/667881/>
- [76] L. I. Kuncheva, *Combining Pattern Classifiers*, Wiley, 2004. doi:10.1002/0471660264.
 URL <https://onlinelibrary.wiley.com/doi/book/10.1002/0471660264>
- [77] T. Feng, X. Zhao, N. DeSalvo, Z. Gao, X. Wang, 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, 2015, pp. 1–6. doi:10.1109/THS.2015.7225268.
 URL <http://ieeexplore.ieee.org/document/7225268/>
- [78] R. M. Cruz, D. V. Oliveira, G. D. Cavalcanti, R. Sabourin, FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection, *Pattern Recognition* 85 (2019) 149–160. doi:10.1016/j.patcog.2018.07.037.
 URL <https://linkinghub.elsevier.com/retrieve/pii/S0031320318302760>
- [79] M. Smith-Creasey, M. Rajarajan, Adaptive threshold scheme for touchscreen gesture continuous authentication using sensor trust, in: *Proceedings - 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 11th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Conference on Embedded Software and Systems*, IEEE, 2017, pp. 554–561. doi:10.1109/Trustcom/BigDataSE/ICCESS.2017.284.
 URL <http://ieeexplore.ieee.org/document/8029487/>
- [80] M. Friedman, A Comparison of Alternative Tests of Significance for the Problem of S_m Rankings, *The Annals of Mathematical Statistics* 11 (1) (1940) 86–92. doi:10.1214/aoms/1177731944.
 URL <http://projecteuclid.org/euclid.aoms/1177731944>
- [81] P. Nemenyi, Distribution-free multiple comparisons, *Biometrics* 18 (2).
- [82] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, *Tech. Rep. 1* (2006).
 URL <http://jmlr.org/papers/v7/demsar06a.html>
- [83] 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, B. Zupan, Orange: Data mining toolbox in python, *Journal of Machine Learning Research* 14 (2013) 2349–2353. doi:10.5555/2567709.
- [84] I. Chang, C. Y. Low, S. Choi, A. Beng Jin Teoh, Kernel Deep Regression Network for Touch-Stroke Dynamics Authentication, *IEEE Signal Processing Letters* (2018) 1–1doi:10.1109/LSP.2018.2846050.
 URL <https://ieeexplore.ieee.org/document/8378259/>
- [85] S. Choi, I. Chang, A. B. J. Teoh, One-class Random Maxout Probabilistic Network for Mobile Touchstroke Authentication, in: *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 3359–3364. doi:10.1109/ICPR.2018.8545451.
 URL <https://ieeexplore.ieee.org/document/8545451/>
- [86] S. Y. Ooi, A. B.-J. Teoh, Touch-Stroke Dynamics Authentication Using Temporal Regression Forest, *IEEE Signal Processing Letters* 26 (7) (2019) 1001–1005. doi:10.1109/LSP.2019.2916420.
 URL <https://ieeexplore.ieee.org/document/8713391/>