Towards Continuous User Authentication Using Personalised Touch-Based Behaviour

Peter Aaby
Edinburgh Napier University
School of Computing
Edinburgh, United Kingdom

Mario Valerio Giuffrida
Edinburgh Napier University
School of Computing
Edinburgh, United Kingdom

William J Buchanan
Edinburgh Napier University
School of Computing
Edinburgh, United Kingdom

Zhiyuan Tan
Edinburgh Napier University
School of Computing
Edinburgh, United Kingdom

***Abstract*—In this paper, we present an empirical evaluation of 30 features used in touch-based continuous authentication. It is essential to identify the most significant features for each user, as behaviour is different amongst humans. Thus, a fixed feature set cannot be applied to all models. We highlight this importance by selecting features accordingly using our approach, seeking to individually select and empirically test the discriminative power of a range of features as well as feature interaction in the context of individual users. We test five different feature selection techniques: Mutual Information, Sequential Forward Selection, Sequential Floating Forward Selection, Sequential Backwards Selection, and Sequential Floating Backwards Selection. Our results show that a unique set of features can be selected for each user, while increasing or maintaining performance, i.e. up to 27 out of 30 features were removed for one user without affecting performance. We also show that distinctive features should be evaluated on a user basis, as particular features may be significant for some, while redundant for others. Moreover, for each user, the same features are selected for horizontal and vertical strokes while performance persists when using a horizontal model to predict vertical behaviour and vice versa.**

***Index Terms*—Behavioural Biometrics, Continuous Authentication, Feature-selection**

I. Introduction

Smartphones typically provide a range of knowledge-based and physiological biometric authentication methods to secure access through lock-screens. The former includes PIN codes, passwords, and drawing patterns, whereas the latter integrates specific hardware, such as biometric fingerprint scanners or facial recognition. However, such solutions are typically used for one-off authentication, where the users authenticate once before starting a new session, with secrets being keyed in or by providing irrevocable fingerprint/facial images for more user-friendly authentication. Knowledge-based authentication methods are inherently vulnerable since secrets can be lost, shared, or even stolen, whereas biometrics are vulnerable to replay attacks [1], [2]. In these contexts, users must also actively engage with the authenticator, where up to 9% of the time is spent unlocking devices, taking away valuable time and requiring conscious attention by the user [3].

Instead, *Continuous Authentication* (CA) aims to ease the burden on users by binding their behaviour closer to a digital profile, through passively collecting sensory input and measuring signals against known behaviour. CA then compares if an incoming stream of signals is within an acceptable confidence level of an owner's behaviour. Thus, CA attempts to address the shortcoming of traditional authentication by removing the demand for active user input, while also following the user's dynamic behavioural pattern making it more difficult to capture and replay. The popularity of smartphones and their inherent mobility also present an increased risk of theft and a consequent loss of property compared to computers. Smartphones may also carry an increasing amount of Personal Private Information (PPI) data and allow financial transactions where users have adopted mobile payment methods.

Consequently, by utilising high-quality models of observed behaviour, CA could enable a paradigm shift from traditional one-off authenticators toward continuous seamless and unobtrusive user authentication over time. However, the challenge of uniquely creating a high-quality model remains, since users behave differently, and therefore one solution cannot be applied across all users. Different smartphone sensors support behavioural detection, such as accelerometer and gyroscopic data that may be combined to detect hand movement, orientation, and grasp [4], [5]. However, in this paper, we focus purely on touch-based CA using information that can be gathered exclusively from the touchscreen on smartphones. Touch data includes (x, y) coordinates of finger touch-down movement and when the finger is lifted together with auxiliary information including timestamps, device orientation, pressure, the area covered by a finger, and application IDs. Through the collection of raw touch data, researchers have focused on advancing CA by the engineering of features, selecting appropriate classifiers, and tuning hyperparameters while training models using varying sample sizes. We extend the body of work by exploring and empirically evaluating features for individual users. We also highlight that, within CA, a behaviour is expressed through features. Thus, the inclusion or exclusion of specific features should either improve or decrease model performance depending on how well a feature aligns with a user's unique behaviour.

A. Challenges and Motivation

CA is still in its infancy with a range of challenges [6]. We highlight the major *two* challenges motivating this paper: (i) human behaviour is unpredictable and subject to change over time as users adapt to various environments; (ii) different users may expose individual behaviour through distinct feature-sets. Therefore, feature selection should be done at a user level.

B. Contributions

In this paper, we present a rigorous analysis of user-level feature selection for CA applications. A One-vs-Rest (OvR) approach is introduced to create a training set for each user of interest, allowing for the analysis of feature-importance in the context of unique and individual user-behaviour. OvR has not been thoroughly explored in related work. Different types of behaviour are expressed through 30 features, and since humans may behave differently, the selection of the most discriminative features is essential. Selecting minimal but highly discriminative features could reduce noise in behavioural models and potentially improve performance. In this work, features are empirically tested using KNN and SVM classifiers while applying five different feature-selection algorithms for each classifier. We evaluate our method using a subset of the *TouchAlytics* dataset [7]. The experimental results show that our approach improves the state-of-the-art by identifying *Sequential Forward Selection* as the optimal feature selection technique in combination with an SVM classifier for the selected users.

The rest of the paper is structured as follows: Section II reviews the related work. Section III describes the proposed method. Section IV presents the feature selection techniques and analysis, with SectionV extending through results and discussion, before concluding the paper in Section VI.

II. Related Work

TouchAlytics: The dataset presented in [7] includes touch-based behavioural data as a viable sensory input for use in Continuous Authentication (CA). They acquired data by developing an Android application that presents a user with Wikipedia articles to read or a "find five differences on a picture" game. Reading articles was designed to collect vertical strokes, while the game caused users to slide horizontally between pictures. While using the app, touch data was recorded and allowed the extraction of 32 features. The Pearson correlation and Mutual Information (MI) were used to rank such features. Three features were removed using expert knowledge obtained by evaluating the two ranking methods. K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) classifiers were applied, producing results to support touch data as a viable sensory input for CA with Equal Error Rates (EERs) around 2-9% when combining 10-13 strokes.

Which classifiers work?: Similar to [7], Serwadda et al. [8] collected data from 190 subjects focusing on which versifiers work while separating behaviour into four templates such that horizontal and vertical behaviour is modelled individually for portrait and landscape modes. They trained models using 80 samples from a target user while drawing 80 randomly chosen strokes from imposters (i.e., the OvR approach). Each model uses the same 28-dimensional feature set. When testing, ten strokes were averaged using a sliding window to allow for more stable authentication. Individual classifiers achieved a mean EER between 10.5% and 42% using Logistic Regression (LR) and Decision Tree (DT), best and worst, respectively. Interestingly, horizontal models generally outperformed the vertical ones in portrait mode, while there was limited change in horizontal or vertical scores in the landscape mode. Furthermore, SVM seemed to be the most stable classifier when considering both mean EER and its variance across all models while KNN scored second worst.

Horizontal vs Vertical behaviour: Fierres et al. [9] supported the evidence found by [8], whereby horizontal strokes are more discriminative. Their system of classification works by training a model using T randomly chosen samples from the legitimate user, and $T/10$ samples from an imposter population. Two classifiers were trained using two different feature sets. The first model applied SVM with a 28-dimensional feature set proposed by [8]. The second model implemented a Gaussian Mixture Model (GMM) using another signature feature set consisting of the five best features selected through Sequential Floating Feature-Selection from a 61-dimensional feature set [10]. They tested behavioural models by averaging ten strokes using a sliding window and found that horizontal strokes were faster than vertical strokes with EER around 10%, independent of device orientation. Additionally, strokes performed in the portrait mode were more stable than the landscape mode.

Dot-to-dot CA: In 2016 a hybrid authenticator called TMGuard was introduced by Manulis et al. [11] which combines Android's dot-to-dot unlock-patterns with touch-based CA. As such, this method is not fully transparent since users must draw patterns to unlock. The research surveyed 75 participants and demonstrated that individuals might expose stable but unique behaviour when interacting with the dot-to-dot unlock-pattern. Similar to others, TMGuard evaluates strokes separately by grouping up, down, left, and right. Contrary to earlier work, this work defines unique behaviour only using two features: the Speed of Touch Movement (STM) and the Angle of Touch Movement (ATM). Behaviour is then evaluated using a statistic-based profile matching approach over several strokes which distance the work from those applying machine learning methods. Regardless, the work finds similarities by concluding that users may expose consistent behaviour when performing the same strokes, although this varies across users.

Users and their device: Zahid et al. [12] investigated the effect of user-posture, the difference in screen size across different smartphones and tablets, and provided insights to inter-session variation. They extracted 18 features from their raw data and discarded four features using MI similar to [7]. Their result shows that the EER exponentially improves when increasing training sample size from 10, 20 - 30% with a flat performance at 40% and gradually decays using further training data. After training, user-authentication is performed by combining five strokes providing a mean EER between 3.8-8.8%, min and max rates, respectively. Models from tablets perform better than smartphones with smaller screen sizes and transferring user profiles between devices appears to degrade authentication performance.

One-class classifier approach: In [13], the authors present an evaluation of 45 participants using WeChat over two weeks. This work differs by approaching CA using one-class SVM classification and by categorising behaviour into four significant groups including vertical, horizontal, oblique, and clicks. Up to 16 features were extracted from each category and selected using fisher scores [14]. Models were also trained with varying sample sizes and hyperparameters with the best performance found by combining nine strokes and using 80 samples for training. Results are presented using F-scores with oblique strokes outperforming others while clicks are inferior.

Summary: CA has greatly improved, due to the engineering of behavioural features which has been tested against several classification approaches. KNN and SVM are commonly used and provides a good foundation for comparability amongst papers using EER as a performance metric. At the same time, other classifiers may also prove suitable such as GMM, LR, DT, and Neural Networks [8]. In this paper, we limit our investigation to KNN and SVM classifiers as the focus remains of identifying the distinctive features in the context of individual users. While feature selection was mentioned in the related works, the application is limited and not rigorously explored, especially in the context of modelling individual users. In work applying feature selection, statistical ranking techniques such as MI are often used to estimate significant features before manual removal using expert knowledge; thus, the correlation between features and applied classifiers remain unknown. Furthermore, applying feature selection in combination with OvR distances this work by uncovering features that may be important to most as well as those only important to some and potentially improving EER score.

For most of the related works, EER is reported to describe model performance, which defines the decision threshold where False Acceptance and Rejection Rates are equal. For comparability, our results are presented using the average EER score for all feature selection techniques. However, several related works [7], [8], [9], [15] consider multiple strokes for authentication, which prohibits exact performance comparison between papers. Our work will take a pragmatic stance by reporting EER and authenticating users using singular strokes. Consequently, all EER scores may be improved by considering multiple strokes but is currently beyond the scope of identifying the most significant features for individual users.

III. Proposed Methods

In this section, we present the methods used to select users of interest, clean the selected data and ensure class balancing for model fairness, as well as the methods used for model selection and hyper-parameter tuning.

A. Data Set and Users of Interest

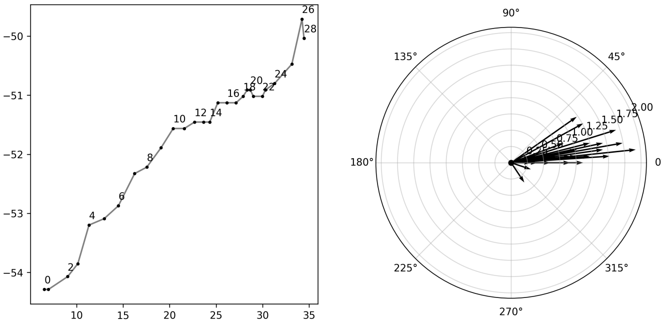
The data used for this research is extracted from a public set collected by Frank et al. [7], containing touch inputs from 41 subjects interacting with seven different documents over two weeks. However, not all users participated in the entire experiment. Thus, we only select users that had provided data for the whole duration of the experiment (2 weeks), because of the interest towards assessing model stability over time. Consequently, the data set is reduced to 14 users and separated into inter-session (week one) and inter-week (week two). Amongst the 14 users, a further two users were removed (namely, user ID 5 and 35), as they exhibited inconsistent behaviour. All users carried out two general tasks involving reading Wikipedia articles and playing an image comparison game. The activities are referred to as document IDs. Documents 1, 2, 3, and 6 are Wikipedia articles, whereas 4, 5, and 7 are Gaming (comparing pictures).

B. Data Cleaning and Filtering

Filtering taps and long or idle strokes: Fig.1 presents the raw data collected from a single stroke performed by a user playing the picture comparison game described in [7]. A single stroke consists of points of (x, y) coordinates, which collectively compose a trajectory. Since this work focus on sliding strokes, the number of points included in a stroke must exceed that of a tap. Fig.2 presents the distribution of points generated in strokes across the selected subset of data. In this work, each stroke must contain a minimum of five points, while lengthy strokes are defined as the top 0.1% (roughly 550 points) highest points. We also remove strokes with inter-stroke time exceeding 1000ms. These filters allow the removal of brief and lengthy strokes such as tap, sticky fingers and strokes far between each other.

Missing values: Features with no value may arise in strokes with few points, such as when calculating *20% pairwise velocity* or *median acceleration over the first five points* over strokes with less than 10 points. Missing values also occur for the last stroke performed by users, as intra-stroke times are unavailable due to being the last interaction. Strokes with incomplete values are discarded in its entirety as they provide no value and constitute an insignificant number of strokes.

Finger orientation: In contrast to the original feature set proposed by [7], we remove the *change of finger orientation* feature, as the variable is consistent across all samples and therefore provide no distinctive behavioural information. However, all other features are kept for the feature selection



(a) Raw stroke with 29 points

(b) Directional pairwise vectors

Fig. 1: Example stroke from 1 vertical touch-screen interaction.

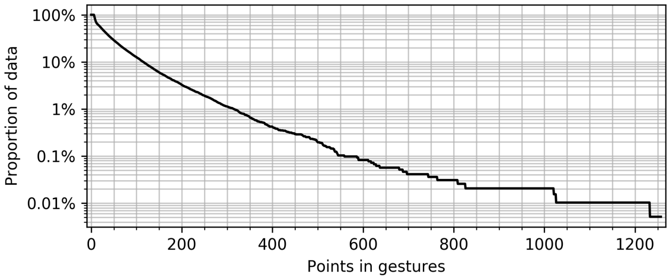


Fig. 2: Distribution of points within strokes across all users.

TABLE I: Features included in feature selection step.

#	Description	#	Description
1	inter-stroke_time	16	80 perc. pairwise acceleration
2	stroke_duration	17	median velocity at last 3pts
3	start x	18	largest dev. end-to-end line
4	start y	19	20 perc. dev. end-to-end line
5	stop x	20	50 perc. dev. end-to-end line
6	stop y	21	80 perc. dev. end-to-end line
7	direct end-to-end distance	22	average direction
8	mean resultant length	23	length of stroke
9	up/down/left/right flag	24	ratio F7:F23
10	direction of end-to-end line	25	average velocity
11	20 perc. pairwise velocity	26	median acceleration first 5 pts.
12	50 perc. pairwise velocity	27	mid-stroke pressure
13	80 perc. pairwise velocity	28	mid-stroke area covered
14	20 perc. pairwise acceleration	29	mid-stroke finger orientation
15	50 perc. pairwise acceleration	30	phone orientation

technique to analyse, which is contrary to the original work by Frank et al. who removed *average velocity*, *length of trajectory*, and *orientation of end-to-end line*. Section V further highlights why these features should be included since they may present important biometric properties for some users.

Stroke direction: Similar to [7], each stroke is categorised as up, down, left, or right by evaluating directional data. An example is shown in Fig. 1b, highlighting the spread over pairwise vectors from a horizontal stroke. Each pairwise vector reveals minutiae behavioural detail within a stroke. As such, it is essential to extract the right features based on raw data, as well as selecting the most discriminative features identifying an individual user. Overall, we evaluate 30 features, as shown in Table I, of which a subset of them is selected for each user individually (details described in Section IV).

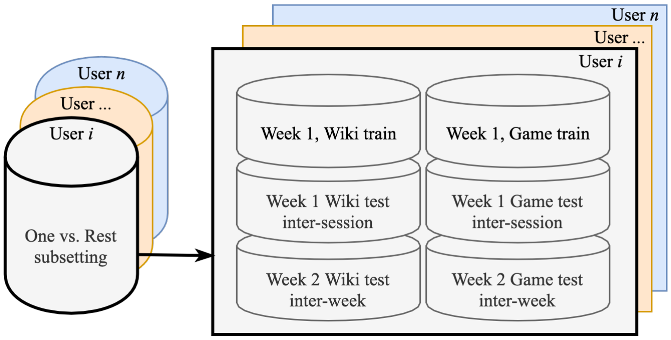
C. Class Balancing and Model Fairness

Since the classification task remains to tell the device owner apart from a non-owner, the multi-class challenge can be transformed into a two-class classification consisting of n subsets with binary class labels. Binarizing multi-class with this approach is also known as One-vs-Rest (OvR), signifying a single user as the positive class while grouping remaining users into another negative *rest* class. However, transforming a multi-class problem into OvR causes class imbalance, as the negative samples are more than positive ones, which may cause classification bias. We overcome class imbalance by relabelling to OvR and down-sampling the majority class, as shown in Fig. 3. However, each user contributes a different number of samples and balancing should be fair amongst models to ensure the approach is stable and comparable between users. As such, the user contributing the lowest maximum strokes will define an upper limit of allowed strokes in the models per class. Thus, for each training set, the positive class is limited to include only the 30 first strokes from a target class and roughly three samples from each remaining user in the negative class. Remaining strokes are discarded, allowing model fairness and comparability between users despite some contributing more strokes than others. Furthermore, the feature selection technique is quicker to evaluate when applying smaller sample sizes. At the same time, related work indicates adequate performance with small sample sizes [12], [15].

Each user interacted with the document IDs in a different order. Therefore, we construct each training set by including document ID in the order of target user and their interaction with the Android data collection app. We supply the first two of three IDs when training Wikipedia models while less data is available for games models. Only a single document ID out of two IDs are applied to Game models. For example, user #2 interacted with document IDs in the order of [1, 3, 2, 4, 5, 6, 7], including ID 1 and 3 in the behavioural Wikipedia model and using ID 2 as inter-session test and 7 as inter-week testing data using the sampling strategy illustrated in Fig. 3.

D. Model Selection and Parameter Tuning

As part of the original experiment developed by Frank et al. [7], each user is tasked to read Wikipedia articles and play picture comparison games. Each task was designed to provoke specific interactions such as vertical strokes with Wikipedia and horizontal strokes with games. As such, it is possible to model each interaction separately, and we define reading as *Wiki* and comparing pictures as *Game*. Since we are interested in model stability over time, it is necessary to test the selected features and trained model on data over time. For the first week, the last document ID from week one is held out from training and used to calculate *inter-session* performance. For the second week, the trained model from week one is tested using document IDs from week two, constituting *inter-week* results. Thus, results can be evaluated over time by comparing the performance of *inter-session* versus *inter-week* scores.



User *n*
User ...
User *i*
One vs. Rest subsetting
User *n*
User ...
User *i*
Week 1, Wiki train
Week 1, Game train
Week 1 Wiki test inter-session
Week 1 Game test inter-session
Week 2 Wiki test inter-week
Week 2 Game test inter-week

Fig. 3: One-vs-Rest sub-sampling approach for each of the 12 users selected.

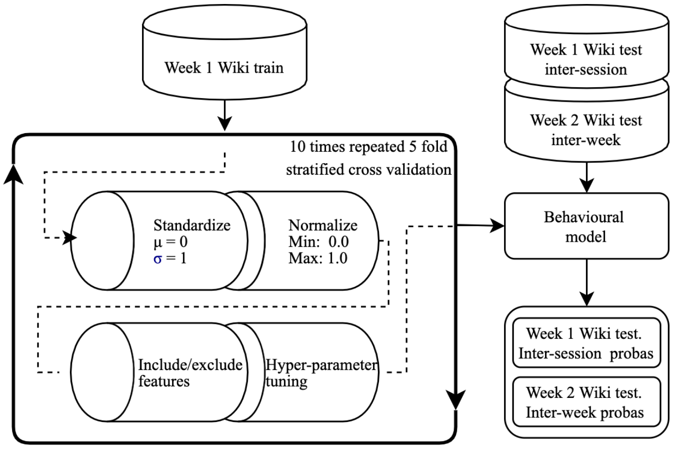

Week 1 Wiki train
10 times repeated 5 fold stratified cross validation
Standardize μ = 0 σ = 1
Normalize Min: 0.0 Max: 1.0
Include/exclude features
Hyper-parameter tuning
Week 1 Wiki test inter-session
Week 2 Wiki test inter-week
Behavioural model
Week 1 Wiki test. Inter-session probas
Week 2 Wiki test. Inter-week probas

Fig. 4: Modelling approach for each user using OvR data.

Fig. 4 illustrates the training pipeline used to select features and tune hyperparameters. The proposed approach applies to both Wiki and Game models, wherein models are trained on the first session(s) of data for the individual user as previously demonstrated in Fig. 3. Since we are learning from relatively small sets of training data, ten times repeating 5-fold cross-validation is used to minimise bias between each feature and parameter test. Within each test, training data is standardised to have one standard deviation with zero mean and scaled to a min-max range in $[0, 1]$, as in [8], [9]. Pre-processing adjusts the data for feature selection and hyperparameter tuning. Features are then included or excluded based on performance rank together with hyperparameter tuning of the classifier using the selected features in each test.

In this work, we adopted two classifiers: KNN and SVM similar to the work of others [7], [8], [9], [12]. We evaluate KNN setting $k = 3, 5, 7, 9$ with neighbour weight estimated by the inverse of their Euclidean distance or uniformly distributed. For SVM, a Radial Basis Function is used as a kernel and all combinations of γ and C values of 0.0001, 0.001, 0.01, 1, 10, 100, 1000. For all cases, models are selected and optimised to maximise the *Area Under Curve* (AUC) since this score is threshold independent while also allowing identification of the best error trade-off between both classes [16].

IV. Feature Selection and Analysis

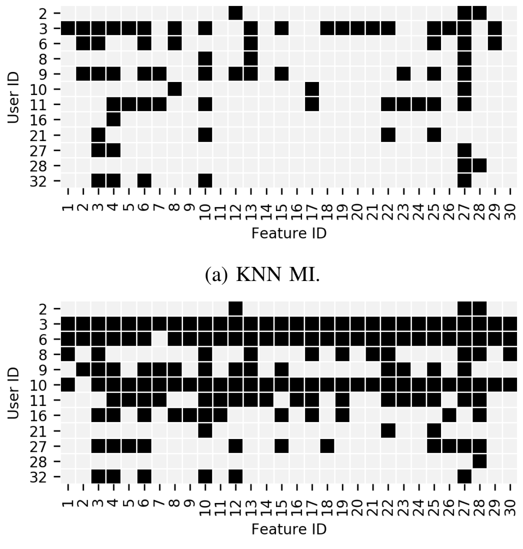
In this section, we present several feature selection methods together with our results for each while analysing the different outcomes amongst the approaches. Feature selection is a type of dimensionality reduction that aims to determine the smallest feature set required to predict a target class. It not only allows faster computation but also reduces model complexity. When modelling user behaviour, it may be necessary to consider the feature importance concerning the target user dynamically. In the case of CA, the positive class usually consists of the data produced by the owner of a device. In contrast, other users are collectively considered as the negative class. In these experiments, the selected features returned by all selection techniques for both Wikipedia and Game interaction are always identical. We report only one feature set for brevity.

A. Expert Knowledge

During feature engineering, features such as the change of finger orientation may logically provide valuable information. However, none of the included users changed their finger orientation. Thus, the feature does not add any further information and is removed. Similarly, phone orientation merely identifies a few users orienting their phone differently from others. However, removing this feature is not recommended as such behaviour may be highly discriminate for specific users. Therefore, this feature is included and empirically tested as part of the implemented feature-selection techniques. In this work, all features except the change of finger orientation remain included for empirical testing by the selection algorithms. For all users, features for Wikipedia and Game models are always the same. Thus, we present feature maps that are valid for both models.

B. Univariate Feature Selection

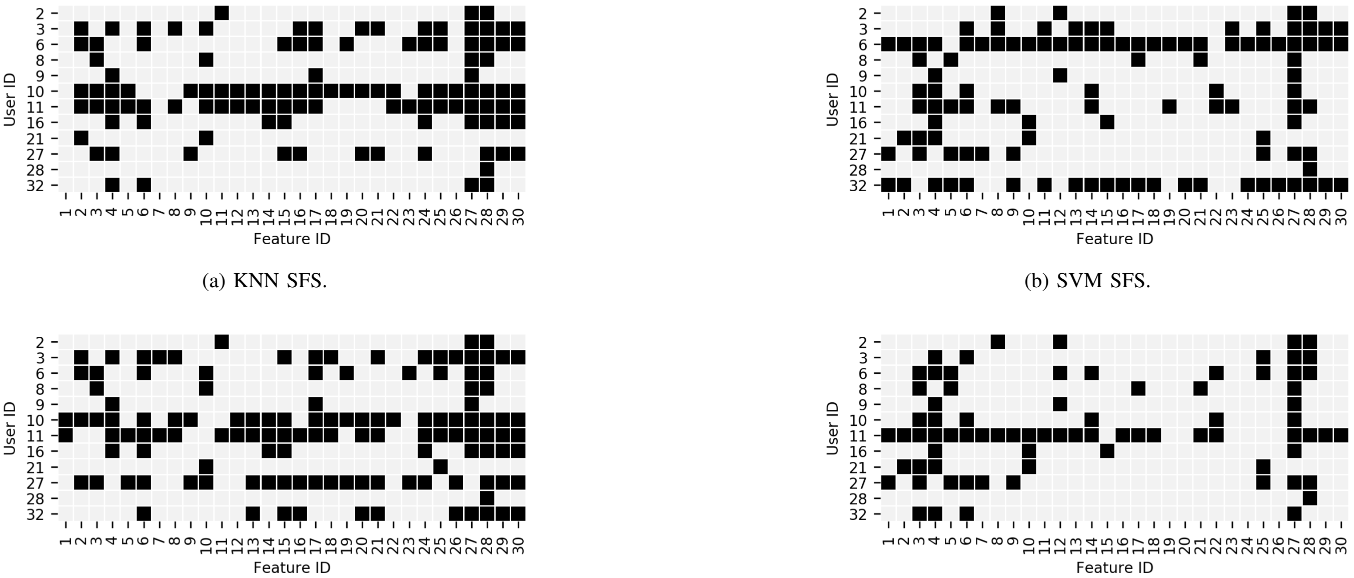
Filtering techniques, also known as univariate selection, work by ranking each feature by applying a scoring function. In the related work, MI is often applied as the scoring function [7], which returns a statistical measure of information gained between an individual feature and the class label. MI [17] is fast to compute since it does not apply a classification algorithm, but at the same time is also unable to describe how features interact with a classifier. Therefore, features are tested using the modelling approach in Fig. 4 by iterating and including k highest-ranked features for hyperparameter tuning in the range of k between 1 and 30. Our results are shown in Fig.5, which presents the selected features by applying MI for both KNN and SVM classification where included features



(a) KNN MI.

(b) SVM MI.

Fig. 5: Selected features using Mutual Information (MI).



(a) KNN SFS.

(b) SVM SFS.

(c) KNN SFFS.

(d) SVM SFFS.

Fig. 6: Selected features using Forward Selection methods

are marked with a black square. Overall, in Fig.5a, it can be observed that KNN has selected fewer features than SVM as shown in Fig. 5b.

C. Sequential Feature Selection

To overcome the drawbacks of univariate selection such as the inability to measure feature-interaction, applying *Sequential Feature Selection* provides insight into such interaction both between features and classification algorithm while testing different subsets. Two modes are available, allowing for inclusion or exclusion of features, namely forward and backwards, respectively. For each mode, a binary float option controls whether the sequence is allowed to reverse between inclusion/exclusion for as long as the decision function improves or maintains performance. This section presents four sequential selection techniques, including Sequential Forward Selection (SFS), Sequential Floating Forward Selection (SFFS), Sequential Backwards Selection (SBS) and Sequential Floating Backwards Selection (SFBS) [18].

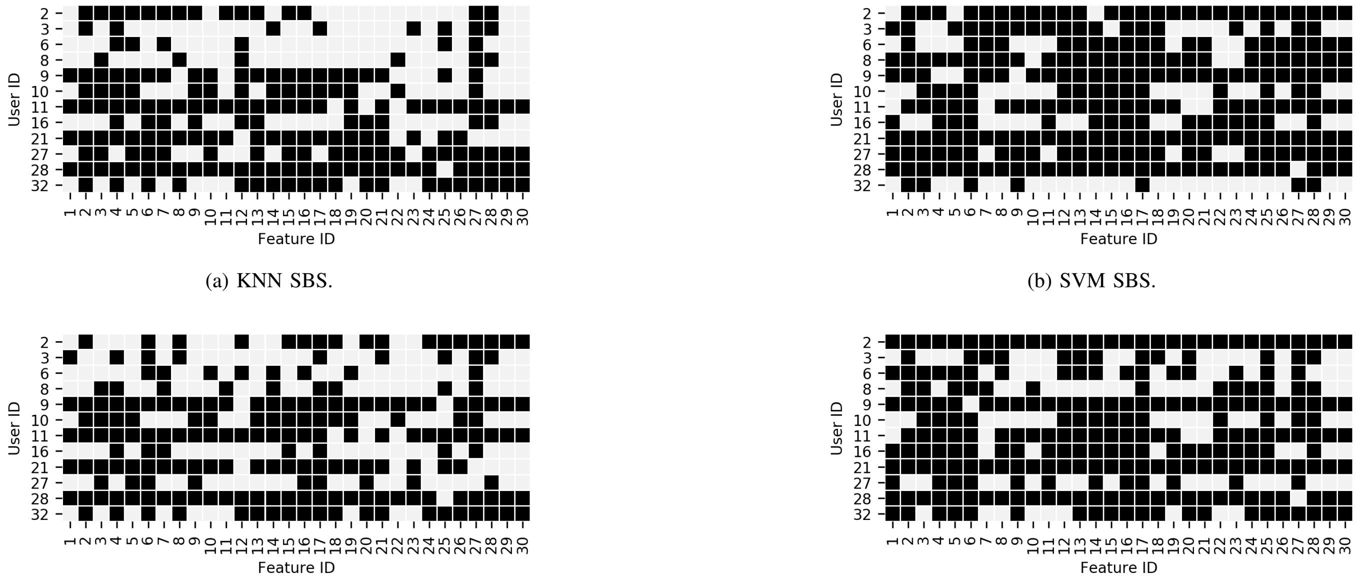
Forward Selection: Using SFS, the feature selection technique begins with an empty feature set and iteratively tests the performance of each feature for inclusion in the forward selection step. If performance persists or increases, then the feature remains; Otherwise, the feature is marked as insignificant and excluded in the final user model. As such, this approach attempts to find the smallest feature set possible. Figs.6a and 6b present the selected features in search of the optimal AUC score for each user. Similar to the SFS approach, Figs.6c and 6d present the impact of allowing the forward selector to float backwards. As such, the number of selected features increases only if previously excluded feature positively interacts with selected features. Selected features remain intact for eight out of 12 users. In contrast, the remaining four users are significantly affected, such as seen with user #32, reducing the selected features from 22 to four when comparing SFS with SFFS, respectively.

Backwards Selection: Contrary to the forward selection, SBS begins with a full feature set while iteratively testing and excluding insignificant features. As such, this approach aims to reduce a feature set by identifying noise. Fig.7 presents the selected features using backwards selection techniques, which, in comparison with forward selection such as seen in Fig. 6, a significant increase in included features can be observed. Similar to SFS, SBS allows for floating operations, which allow previously excluded features to be included in each step; thus, the exclusion list is considered as part of the floating stage until the decision function decays.

Both SFFS and SFBS are computationally more expensive since the methods reintroduce features previously excluded. However, the techniques also provide better coverage in terms of feature interaction and generally produce smaller feature sets. Therefore, touch behaviour of different user can be described with distinct sets of features which confirm research question (ii) *Different users may expose individual behaviour through distinct feature-sets.*

V. Experimental results and discussion

In this section, we present the average results of all user models concerning the selected features using the selection techniques presented in Section IV. To allow for comparison with related work, Figs.8a and 8b present the average EER scores across all individual users, while Figs.9a and 9b present the average AUC scores. The results are separated into inter-session and inter-week to show model stability over time.



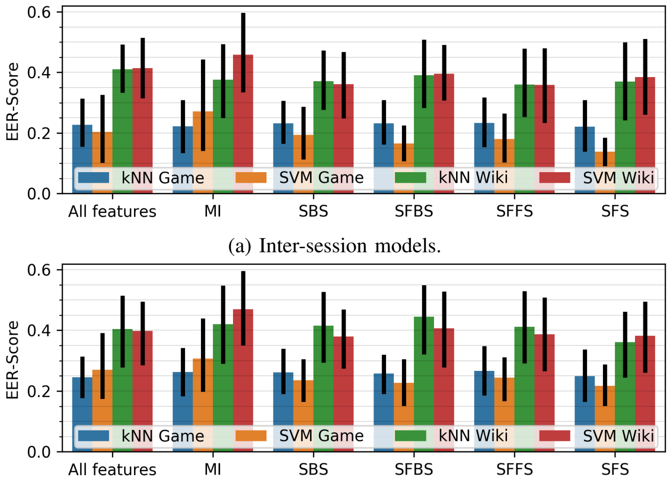
(a) KNN SBS.

(b) SVM SBS.

(c) KNN SFBS.

(d) SVM SFBS.

Fig. 7: Selected features using Backwards Selection methods



(a) Inter-session models.

(b) Inter-week models.

Fig. 8: Mean EER scores for each feature selection technique with 95% confidence interval

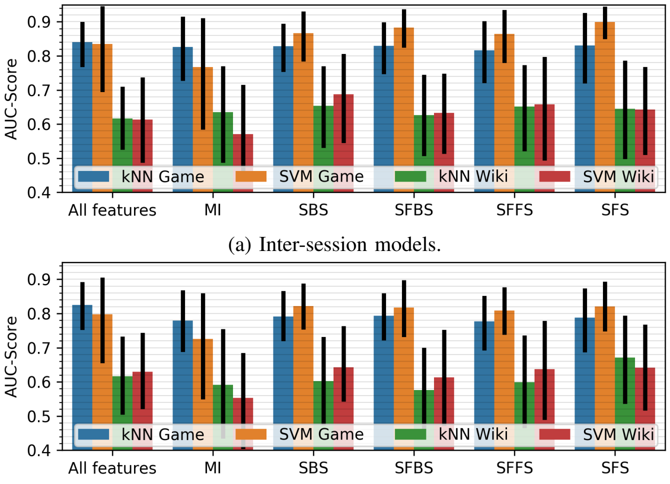
All figures include error bars signifying the 95% confidence interval. We find that *Sequential Forward Selection* maintains or outperforms all other selection methods when applied in an SVM classifier.

Personalised Behaviour: As shown in Section IV, different features are selected for different users when applying our modelling approach as previously illustrated in Fig. 4. Our approach highlights that users express behaviour through different features, and it is possible to reduce model complexity without affecting model performance. Interestingly, we observe in Figs. 9a and 9b that SFS generally outperforms all other feature-selection techniques either by maintaining or improving model performance, even over time. Thus, some features can be removed as they likely introduce noise as users may not conform with specified behaviour calculated by some features. Furthermore, floating options (SFFS and SFBS) do not improve model performance despite consuming more computational resources. As such, it is not advisable to use floating options when applying sequential feature selection on the selected users.

Cross-model Performance: We support the general hypothesis [9], [8] that horizontal strokes are more discriminate. However, we extend the work by observing that all feature-selection techniques identify the same feature-sets when measuring horizontal and vertical strokes independently. As such, we compared model performance by testing predictive Game behaviour against a trained Wiki model and confirm that Game models can predict Wiki behaviour and vice versa. Thus, in this case it may not be important to train two models as they could be interchangeable.

Stability over Time: The selection of features for each user may affect model stability over time. Figs. 9a and 9b compare the AUC score over time, with an expectation of reduced performance because human behaviour tends to change over time and the proposed method is limited to one-off training. Despite the expectation, the majority of applied feature selection techniques sustain performance over time with a limited reduction.

Shared Feature Importance: Fig. 10 presents an overview of selected features across all 60 models, 12 for each selection technique. The lowest occurrence of a single feature is 15 times across all models, whereas the most a feature was included is 50 times. Interestingly, features 10, 23, 25 were removed by Frank et al. in their work [7]; however, the empirical evaluation shows that these features may be significant to specific users. Feature 25, *average velocity*, appears to be a robust generic feature across all the selected users. Besides being robust, certain unique features such as those selected infrequently might help identify specific people. Therefore, models should be trained on a mixture of robust and unique features while selected using an empirical technique. i.e. SFS.



(a) Inter-session models.

(b) Inter-week models.

Fig. 9: Mean AUC scores for each feature selection technique with 95% confidence interval

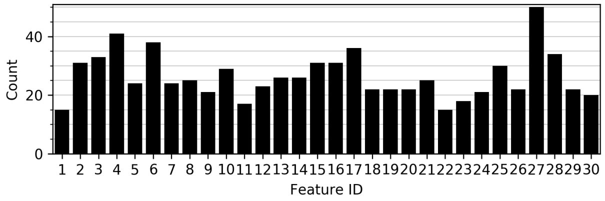


Fig. 10: Frequency of feature across total of 60 models trained.

VI. Conclusion

This research work carried out an empirical evaluation of standard features computed and used in touch-based continuous authentication. Applying the proposed method confirms that features should be considered individually for each user while feature selection techniques both reduce complexity and often improve performance. On average, the best feature selection technique is Sequential Forward Selection in combination with an SVM classifier, especially over time. The final approach results in a horizontal (Game) average EER score of 15% and 22% for inter-session and inter-week, respectively, while vertical (Wiki) EER reached 37% for both inter-session and inter-week. The EER scores are higher than related work since each stroke is evaluated independently. As such, combining strokes as seen in related work suggests that the error rates are conservative results.

The most common features amongst the selection techniques are mid-stroke pressure and mid-stroke area covered appearing in 81% and 73% out of 60 models tested, respectively. On the other hand, inter-stroke time was rarely included but not necessarily insignificant. In the future, it would be interesting to include a more extensive selection of features as well as excluding those that are screen-size dependant.

References

[1] F. Rieger, “CCC | Chaos Computer Club breaks Apple TouchID,” 2013. [Online]. Available: https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid

[2] Bkav, “Bkavs new mask beats Face ID in ”twin way”.” 2017. [Online]. Available: bkav-s-new-mask-beats-face-id-in-twin-way-severity-level-raised-do-not-use-face-id-in-business-transactions

[3] M. Harbach, A. D. Luca, and M. Smith, “Its a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception,” *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, p. 18, 2014.

[4] Z. Sitova, J. Sedenka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, “HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, May 2016.

[5] C. Shen, Y. Li, Y. Chen, X. Guan, and R. A. Maxion, “Performance Analysis of Multi-Motion Sensor Behavior for Active Smartphone Authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 48–62, Jan. 2018.

[6] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, “Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges,” *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, Jul. 2016.

[7] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, “Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[8] A. Serwadda, V. V. Phoha, and Z. Wang, “Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms,” in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Arlington, VA, USA: IEEE, Sep. 2013, pp. 1–8.

[9] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, “Benchmarking Touchscreen Biometrics for Mobile Authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2720–2733, Nov. 2018.

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

[11] W. Meng, W. Li, D. S. Wong, and J. Zhou, “TMGuard: A Touch Movement-Based Security Mechanism for Screen Unlock Patterns on Smartphones,” in *Applied Cryptography and Network Security*, M. Manulis, A.-R. Sadeghi, and S. Schneider, Eds. Cham: Springer International Publishing, 2016, vol. 9696, pp. 629–647.

[12] Z. Syed, J. Helmick, S. Banerjee, and B. Cukic, “Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability,” *Journal of Systems and Software*, vol. 149, pp. 158–173, Mar. 2019.

[13] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated Machine Learning: Concept and Applications,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[14] Q. Gu, Z. Li, and J. Han, “Generalized Fisher Score for Feature Selection,” *arXiv:1202.3725 [cs, stat]*, Feb. 2012, arXiv: 1202.3725.

[15] Y. Yang, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, “BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics,” *Ad Hoc Networks*, vol. 84, pp. 9–18, Mar. 2019.

[16] Z. Wang and Y.-C. I. Chang, “Marker selection via maximizing the partial area under the ROC curve of linear risk scores,” *Biostatistics*, vol. 12, no. 2, pp. 369–385, Apr. 2011, publisher: Oxford Academic.

[17] B. C. Ross, “Mutual Information between Discrete and Continuous Data Sets,” *PLOS ONE*, vol. 9, no. 2, p. e87357, Feb. 2014, publisher: Public Library of Science.

[18] S. Raschka, “MLxtend: Providing machine learning and data science utilities and extensions to Pythons scientific computing stack,” *Journal of Open Source Software*, vol. 3, no. 24, p. 638, Apr. 2018.