Journal of the
Brazilian Computer Society

RESEARCH Open Access

An investigation of the predictability of the Brazilian three-modal hand-based behavioural biometric: a feature selection and feature-fusion approach

Julliana Caroline Goncalves de A. S. Marques[1], Tuany Mariah Lima Do Nascimento[1], Brenda Vasiljevic[1], Laura Emmanuella Alves dos Santos Santana[2] and Márjory Da Costa-Abreu[3]*

*Correspondence:
m.da-costa-abreu@shu.ac.uk
[3]Sheffield Hallam University (SHU), Sheffield, UK
Full list of author information is available at the end of the article

Abstract

New security systems, methods or techniques need to have their performance evaluated in conditions that closely resemble a real-life situation. The effectiveness with which individual identity can be predicted in different scenarios can benefit from seeking a broad base of identity evidence. Many approaches to the implementation of biometric-based identification systems are possible, and different configurations are likely to generate significantly different operational characteristics. The choice of implementational structure is, therefore, very dependent on the performance criteria, which is most important in any particular task scenario. The issue of improving performance can be addressed in many ways, but system configurations based on integrating different information sources are widely adopted in order to achieve this. Thus, understanding how each data information can influence performance is very important. The use of similar modalities may imply that we can use the same features. However, there is no indication that very similar (such as keyboard and touch keystroke dynamics, for example) basic biometrics will perform well using the same set of features. In this paper, we will evaluate the merits of using a three-modal hand-based biometric database for user prediction focusing on feature selection as the main investigation point. To the best of our knowledge, this is the first thought-out analysis of a database with three modalities that were collected from the same users, containing keyboard keystroke, touch keystroke and handwritten signature. First, we will investigate how the keystroke modalities perform, and then, we will add the signature in order to understand if there is any improvement in the results. We have used a wide range of techniques for feature selection that includes filters and wrappers (genetic algorithms), and we have validated our findings using a clustering technique.

Keywords: Hand-based biometrics, Keyboard keystroke dynamics, Touch keystroke dynamics, Handwritten signature, Feature selection, Feature fusion, Filters, Genetic algorithms, *k*-means

1 Introduction

The design of a biometric-based classification system is a particularly challenging pattern recognition task [40]. The fundamental nature of this type of data and the application domain make biometric data very specialised. The effectiveness with which individual identity can be predicted in different scenarios can benefit from seeking a broad base of identity evidence. Many approaches to the implementation of biometric-based identification systems are possible, and different configurations are likely to generate significantly different operational characteristics.

At the very least, it is generally necessary to include an appropriate strategy for exception handling in any significant biometric application scenario. Yet, the design of these systems normally focuses on one particular issue rather than analysing the problem as a whole. Exploiting a broader range of information about the task offers improved levels of accuracy, while also increasing the resilience of a system to, for example, spoofing attacks, or inclusiveness. Also, in order to test the accuracy and overall performance of security systems, it is necessary to subject them to similar conditions as what they would find in a real situation [7].

The issue of improving performance can be addressed in a number of ways, but system configurations based on integrating different information sources are a widely adopted means of achieving this [45]. Nevertheless, the often complex analysis required to choose an optimal modality (or many) for an application in the face of such conflicting demands has been a major factor, as has been trying to identify the best classification structures [25].

A very popular solution to deal with security issues in biometric-based systems is to use more than one modality. Multimodal systems are most often tested with a combination of different biometric databases—one for each modality. As these databases do not have the same subjects, the multimodal databases end up with 'fictional' subjects—each 'person' will have a mix of the biometric characteristics of two or more different databases (meaning different users). Such combinations can be done at the feature level, classification level or decision level, and there is no investigation that shows if their differences can have a great impact on the performance of the systems [47].

One approach that is being disregarded, specially with the advent of deep learning techniques, is the selection of features, since such approach uses all the raw data as input [18], but needs a huge quantity of data, which in the current research of biometrics is still non-realistic.

Thus, the present work aims to analyse several important differences in the feature-level fusion and if such a combination provides test conditions that are close enough to a real-life situation. Moreover, we have chosen to investigate the feature selection process of three hand-based biometric modalities. Two are very theoretically similar modalities: keyboard keystroke dynamics and touchscreen keystroke dynamics and the other one is the well-known handwritten signature. Our investigation aims to explore the benefits of feature selection by analysing the performance of filter and wrapper methods. We will then validate our findings by applying a clustering technique. We also have investigated how fundamentally different strategies for implementation can increase the degree of choice available in achieving particular performance criteria in the hand-based modalities.

This paper is organised as follows: Section 2 will present the main background necessary to understand how signature and keystroke dynamics work in their different acquisition approaches and characteristics. Section 3 will present the different ways in which we have performed feature selection. Section 4 will present our results for feature-level fusion with our filter and wrapper approaches as well as an in-depth analysis using our k-means and statistical tests, and, finally, Section 5 will present our final remarks about this work as well as open issues we believe are worth investigating.

2 Understanding hand-based behavioural biometric data: feature extraction and selection

According to [39], biometrics can be defined as a set of unique physical or behavioural characteristics. Physical biometrics make use of human body traits, for example, the iris, fingerprint, palmprint and retine [20]. Behavioural biometrics use patterns of human behaviour, such as handwritten signature, keystroke dynamics and voice [5].

The use of behavioural biometrics aiming to recognise fraud can be a complex task, since the manners and body language of users are a combination of social and psychological factors. Thus, it is quite common to use a combination of two or more biometric modalities [29]. Therefore, one of the most used biometric data fusions is the combination of biometric data of both modalities in one single database (feature-level fusion) [29]. The combination of biometric modalities must be done carefully because the merge between the databases can introduce irrelevant and/or redundant information. So, it is necessary to use tools that help on removing this kind of information.

The field of biometrics provides an extensive and diverse literature reporting studies on several approaches for data combination as well as multimodal systems. Discussion of various modalities, fusion techniques, sensor development, security issues and usability problems, as well as the use of revocable biometrics and soft biometrics, can be found in [46].

Despite all the work that has been carried out using biometrics, it is still quite often very difficult to make strong claims about performance which can be statistically significant, mostly because it can be difficult to find large enough databases [18]. Moreover, the experiments which are used to validate any proposed techniques are normally executed in a very controlled environment which does not represent real-world scenarios, while the complexity of some techniques would probably cause usability concerns in practical situations.

When dealing with very similar modalities, one tends to assume that the set of configurations, ranging from features selected to classification chosen, should be the same. So far, no other work has focused on investigating the real reliability of feature-level fusion (or any other fusion) of modalities from different users and its impact on the security of the system.

Thus, we will investigate the feature selection process in the context of similar generated and collected modalities, in our case, a three hand-based biometric modalities: keyboard keystroke dynamics, touchscreen keystroke dynamics and online handwritten signature. Thus, in the next two sections, we will present the most recent works focusing on the feature-level fusion of keystroke (touch and keyboard) dynamics and handwritten signature as well as their main databases available.

2.1 Keystroke dynamics

Keystroke dynamics analysis is a behavioural biometric-based technique. In [3], the authors introduce keystroke dynamics analysis as a low-cost, non-intrusive authentication method that has a clear advantage over password-based authentication: it cannot be lost, forgotten or stolen. While the same can be said about physiological techniques, these generally require expensive and, sometimes, intrusive hardware to collect the biometric data.

As explained in [40], the analysis of typing patterns can be made statically or dynamically. In the static approach, the system verifies the identity of the user before granting them access to the system, generally using typing features collected while the password is typed. In the dynamic approach, the typing patterns are analysed continuously during the work session with data being extracted from arbitrary text input (free-text).

According to Morales et al. in [37], the most popular typing pattern features are as follows:

- *Hold time* is the difference between the time of pressure and release of the same key.
- *Release-press latency, or RP-latency*, is the difference between the time of release of a key and the time of pressure of the next one.
- *Press-press latency, or PP-latency*, is the difference between the time of pressure of a key and the time of pressure of the next one.
- *Release-release latency, or RR-latency*, is the difference between the time of release of a key and the time of release of the next one.
- *Press-release latency, or PR-latency*, is the difference between the time of pressure of a key and the time of release of the next one.

As already mentioned, we will investigate two different ways of collecting keystroke dynamics data depending on whether the system uses keystrokes in physical or virtual keyboards (touchscreens). Sections 2.1.1 and 2.1.2 will present the main literature references of each case, focusing on when the authors use both data.

2.1.1 Keyboard keystroke dynamics

The interest on studying keyboard keystroke dynamics is not new, and it has started mostly with the popularisation of personal computers.

In [3], a static approach to keystroke dynamics authentication is used to improve the conventional login—password authentication method, analysing a combination of hold time, PP-latency and RP-latency. They found that the familiarity with the password impacts the performance, with the false rejection rate increasing when a password is imposed to the user instead of chosen by them. They collected data on three machines from 30 test subjects and the user's samples were collected in different periods of time, instead of all at once.

Ngugi et al. [38] obtained their data through the typing of a provided personal identifying number (PIN), '1234'. There were twelve participants, and they repeated the experiment two more times, a week and then a month after the first session, finding evidence that the hold time and RP-latency lowers over time, impacting negatively the classification as legitimate or impostor attempts, since the samples collected during enrolment no longer reflect the user's typing rhythm. In [37], the proposed system collects keystroke patterns from 63 users typing their personal information (such as family name

and ID) six times to enrol, and after at least 24 h, six more times to try and get access to their accounts, plus twelve times to try and get access to the account of another user. Their results proved the feasibility of the method and found a correlation between hold time and RP-latency, since their combination improved the performance of classifiers.

In order to compare 14 anomaly-detection algorithms, in [28], keystroke dynamics data was collected from 51 subjects, each providing 400 samples—the password was typed 50 times per session, during eight sessions. Loy et al. [33], similarly, provided all users with the same password and keyboard, this one modified to be sensitive to pressure, thus capturing typing pressure patterns along with the typing latency. A hundred users participated in the data collection, providing 10 samples each—a much more reasonable number of samples to require from an actual user without causing annoyance.

The work presented in [46] collected data from three typing activities: password entries, freestyle typing and transcription of fixed text. Among the discussed databases, this one had the largest amount of text per user. The BeiHang database [31] was acquired under real application assumptions: the user could choose their own password, repeating it during the enrolment phase four or five times; the data collection was done without supervision in two different environments, and the number of samples collected varied from subject to subject.

A dynamic approach can be found in [6], with keystroke dynamics extracted from free-text—or more specifically from recurring digraphs and trigraphs (combination of two and three letters) in free-text. Also investigating the use of continuous user authentication, [14] made an analysis of the performance of four keystroke dynamics features in the most frequent letters, digraphs and words in English, finding evidence of their relevance as features for classification algorithms. Corder and Foreman [10] studied the feasibility of using keystroke analysis to continuously authenticate users in mobile devices. The experiment was based on a physical numeric keyboard and involved 32 participants and two scenarios: entry of a 4-digit PIN and entry of an 11-digit 'telephone number'. Each participant repeated both entries 30 times in a single session. Another dataset was generated for the same work, this one requiring 30 subjects to enter a total of 30 'text messages' (longer passages of text) split during three sessions. While promising, the results of [10] were of questionable relevance to the present situation due to the vast changes in mobile devices since then.

Table 1 summarises the databases discussed in this section, including the number of test subjects each employed and the type of text entry required from them—whether the subjects had to type a password, a PIN or a longer text, if it was chosen by the user or if a

Table 1 Keystroke dynamics databases (physical keyboards)

Paper of origin	Subjects	Type of entry	Error rate
[28]	51	Fixed password	9.6% EER
[33]	100	Fixed password	11.8% EER
[46]	39	Many	0.75% FAR, 3.93% FRR
[31]	117	User's password	11.83% EER
[3]	30	User's password	1.6% EER
[38]	12	Fixed PIN	2.0% EER
[37]	63	Personal data	2.3% EER
[10]	32	Fixed PINs	12.8% EER
[10]	30	Fixed text	17.9% EER

fixed entry was provided to all users. The table also shows the equal error rate (EER), false rejection rate (FRR) and false acceptance rate (FAR) of the proposed system that used that database to measure their performance.

In none of the cited studies can we find an in-depth discussion about the feature selection or feature-level fusion in a two-modal keystroke-based database or even the use of three-modal databases with signature. In the following section, we will discuss works related to authentication systems in more modern devices. These systems are also based on keystroke dynamics, but make use of touchscreen technology.

2.1.2 Touch keystroke dynamics

Due to the increase in popularity and, thus, frequent use of mobile devices, it is possible to note a rise in the number of studies related to touch dynamics. The most recent works will be described in this section.

Jeanjaitrong and Bhattarakosol [26] have found that keystroke dynamics can be used as an authentication method in touchscreen devices with similar accuracy rates as in actual keyboards. The comparison was made using only features that can be measured in both virtual and physical keyboards, namely the hold time and RP-latency. Additionally, they had found that touch-specific features, like the distance between touch events, increased the performance of classifiers.

Touch-specific keystroke features were evaluated in [7], some of which often outperformed the RP-latency, one of the most used features in related works. The EER reported in Table 2 is the lowest equal error rate found when the user's hand posture was one during enrolment and another during authentication, a novel variable in literature that highly increased error rates. Their system was tested through 28 subjects typing fixed passwords.

Two of the listed works proposed classification methods based on vector proximity: [1] tested it on a database with keystroke dynamics from 17 experienced users of touchscreen-based smartphones; they typed a message of their choice five times each. On the other hand, [35] tested their method on the dataset used in [2]—collected from a fixed password given to 42 people, and typed 51 times by each one of them. The RHU Keystroke Database [16] was a benchmark dataset with features collected from touchscreen devices, but without touch-specific features. With the goal of analysing tapping behaviours during the input of PINs, [50] collected data from over 80 users in a Samsung Galaxy Nexus. Data collected from iPhones had already been gathered by [26] where 10 randomly selected subjects from iPhone users were asked to input a fixed symbol password.

Table 2 Keystroke dynamics databases (touchscreens)

Paper of origin	Subjects	Type of entry	Error rate
[26]	10	Fixed password	2.0% FAR
			1.8% FRR
[16]	53	Fixed password	None
[2]	42	Fixed password	7.0% EER
[50]	80+	Fixed PIN	3.7% EER
[7]	28	Fixed passwords	33% EER
[1]	17	User's text	12% EER
[35]	42	Fixed password	6.8% EER
[9]	100	User's password	6.9% EER

Finally, [9] asked 100 participants to enter a graphical-based password of their choice five times; the password was a sequence of taps on an apparently undivided image chosen by the user (both the image and the sequence). Then, 10 participants in possession of the other users' passwords tried to attack the system, entering each legitimate user's password five times. The database collected timing and pressure features through an interface that would remain the same even if used on different devices, like the one in [26].

Again, no work can be listed that compares touch with keyboard or combined them, or even included the combination of either with handwritten signature. Keystroke dynamics (keyboard and touch) is, perhaps, the very first case where the basic characteristics of each modality are seen as the same. Thus, our paper aims to investigate how well feature selection and feature-level fusion of the two different ways of collecting keystroke dynamics (keyboard and touchscreen) can be done.

2.2 Online handwritten signature

As opposed to the investigation of keystroke dynamics, the handwritten signature (specially the online version) has been under study due to the growing tendency to use devices to authenticate individuals. The most recent works will be described below.

In [21], the authors proposed a wearable device called Handwriting Watcher, a wrist-worn sensor to authenticate a user through his/her handwriting. From the device's accelerometer and gyroscope data, they extracted 364 features based on the x, y and z components. A selection of 60 features was calculated and classified using SVM and MLP classifiers, obtaining an error rate as low as 6.56%

In [19], a method based on self-taught learning was present. In this method, sparse autoencoders attempted to learn features called discriminative features from a dataset of unlabelled 17,500 signatures (ATVS dataset). In the next step, a model user was created by each original user signature where a one-class classifier was used to classify the features. The verification process was evaluated using the datasets SVC2004 and SUSIG containing original and forgery signatures. A significant error reduction and increase in accuracy was obtained compared with the state of the art. Xia et al. [48] presented two methods based on full factorial and optimal orthogonal experiment design to select discriminative features. The features were analysed to improve the robustness, and then, the more consistent were selected as discriminative candidates. A signature curve constraint was presented, in order to improve the verification step. The experiments based on MCYT and SVC2004 Task2 databases verified the effectiveness and robustness of the methods. Yang et al. [49] proposed a writer-dependent verification technique for online signature systems. The verification process consisted of two phases applied to each user: the training phase and the test phase. In the training phase, pairs of true and counterfeit signatures were created and these characteristics were combined for each pair that forms a group, then the most stable group was selected by the Relief algorithm during the test phase, the grouping was completed using the k-nearest neighbour. The experiments also used a private database, such as the SVC2004 database. As a result, lower FAR and FRR errors were obtained when compared to the state of the art.

The authors of [24] proposed an online signature verification approach using the principal component analysis (PCA) to select the features and the neural network Multilayer Perceptron (MLP) as the classifier. The proposed approach illustrated a reduction in error rate through discarded information by the PCA process. A total of 4000 signatures from

SIGMA database were used and generated results that showed an FAR error of 7.4% and an FRR of 6.4%.

Table 3 summarises the papers previously discussed, including databases used as well as the features extracted from each database, the feature selection approach used and the error rate of the experiments.

Although the works presented in [21, 49] do use the RELIEF filter, in none of the cited studies, we can find a significant discussion about the feature selection or feature-level fusion on a multimodal hand-based situation.

3 Methodology for feature selection

From what we have seen in the previous section, there is no work focusing on analysing how feature selection and feature-level fusion can affect systems with keystroke dynamics and handwritten signature biometric modalities. In order to perform this analysis, we will be using a specific database and a set of well-known classifiers.

Our experiments were conducted using a Brazilian hand-based behavioural biometric database [13] that collected both keyboard keystroke dynamics and touchscreen keystroke dynamics data as well as online signature samples.

The data used on this experiment was collected from 76 individuals in a controlled environment. For the keystroke data (keyboard and touch), they were asked to type the same carefully chosen text—a mix of frequently used words in Brazilian Portuguese that included English cognates and, when possible, important digraphs from both languages. The list of words used were *america, felicidade, internet, pequeno, coisa, primeiro, normal, zoom, fazer, selfie, homem, ultimo, carro, mulher, porque, cuba, case* and *mouse* [13].

For the keystroke databases, the collection protocol for both modalities was similar. For the keyboard mode, the users entered the given word set on a desktop keyboard. For the touchscreen mode, the same set of words was used, being entered by the users on the virtual keyboard of a tablet. From each digraph, the dwell time was extracted from the first letter of the digraph, the flight time between releasing the first one and pressing the second letter of the digraph and the dwell time of the second letter were collected. Thus, the extraction process of features occurred as follows:

Table 3 Handwritten signature main works

Paper of origin	Feature approach	Feature selection approach	Error rate
[24]	*x*, *y* and *z* coordinates	Relief	6.56% EER
[25]	Sparse autoencoders	PCA	0.83 EER
			0.77 EER
[26]	Full factorial and optimal orthogonal experiment design	Consistent and discriminative features	2.17% EER
			2.60% EER
[27]	Similarity matrix of features	PCA	7.4% FAR
			6.4% FRR
[28]	Signature writing time, total number of signatures, horizontal and vertical coordinates, pressure value, height angle, inclination angle, and length-width ratio of a signature *x* and *y* coordinates	Relief	1.25% FAR, 1.25% FRR 5.12% FAR, 5.5 FRR

- Initially all the times, of all the digraphs, of all the words of the protocol
- The words were separated into digraphs when possible
- As some digraphs were repeated, they were ordered by frequency of occurrence
- The samples were formed by the most frequent digraphs
- Each sample contained the time of each digraph's occurrence, without repetition

The keyboard used to capture the words was a QWERTY. This keyboard layout is widely used which facilitates the replication of experiments. The authors of [13] did not consider capital letters, special characters and accents as it would make the collection process more complex and could also interfere with data analysis. When typing errors occurred, their entries were discarded and the user had to re-type the word sequence again [13].

In order to obtain three samples from each user, the database gathered three occurrences of the same digraph in different words or considered two or three digraphs as one due to the proximity of their keys in the keyboard. In the end, a total of 14 digraphs was included in the database: *ME, ER, RI, IC, CA, IM, IR, SE, MO, OO, DE, EL, RM* and *UE*. The features extracted from each digraph were the RP-latency and hold time of both keys in the physical keyboard, and the RP-latency and hold time of the digraph's second key in the virtual keyboard.

In the handwritten database, the samples were collected by having the users signing their full name three times after their collection of both the keyboard and touch keystroke dynamics samples. The extracted features were obtained considering the x and y coordinates of the signature, the pressure of the pen and the time in each point of signature. Thus, 34 features were calculated in to form the database, which were *SIGDIST, TOTALTIME, AVXV, AVYV, VELXZERO, VELYZERO, VEL1Y, VEL1X, AVPRESS, VEL2, VEL3, PIXELCENX, PIXELCENY, VEL4X, VEL4Y, PTD, VEL5, VEL6, HWRATION, VELCOR, SET, INITDIR, DCHANGE, XSIZE, DIST1, YSIZE, DUR1, DUR3, DUR2* and *DUR4*.

In order to compare the impact of the feature selection techniques and the feature-level fusion of the datasets described in this work, we have decided to use the same three classification algorithms used in [13] all of which are available in the WEKA toolkit [22], as well as two variations of the SVM and MLP, named LibSVM and WiSARD. For each of the algorithms described below, a stratified 3-fold cross-validation was used, since there are only three samples per user in the database.

The *k-nearest neighbours classifier*, abbreviated to KNN, is an algorithm that determines the distance between two templates by representing them as vectors with n attributes (being n the number of features in a template) and placing them in a n-dimensional plane. KNN predicts if a template will belong to the same user as their closest neighbour(s) in the n-dimensional plane; after conducting empirical tests, our experiments were configured to consider only a single neighbour for classification purposes as well as the Euclidean distance.

The *Support Vector Machine* (or SVM) classifier builds a hyper-plane in an n-dimensional space to separate training data points by class; the hyper-plane is deliberately placed in the point where it will be furthest away from the nearest training data point—or in our case, the nearest user template—to diminish the risk of mis-classification. Multiclass problems such as the one presented in this paper can be solved by Weka's SVM algorithm using pairwise classification, that is, through dividing the problem into binary classification problems and classifying the user template as belonging to the class that it

was most often labelled as in the classification sub-problems. The settings used in this algorithm were $C = 10$ and kernel being the Puk.

The third classifier is the *Multilayer Perceptron Neural Network*, also known as MLP. The algorithm has multiple layers of processing nodes that map the inputs (user templates) to a set of outputs (users) through a neural network model. The algorithm uses backpropagation to train the neural network; when the input vector is propagated all the way to the output layer, the algorithm compares the user assigned to the template to the correct user, calculates the error value and goes backwards through all the layers, assigning each of the processing nodes in the network a weight based on their contribution to reaching that output. The settings used in this algorithm were learning rate $= 0.01$, momentum $= 0.9, 5000$ iterations and 72 neuron in the hidden layers.

The *Library for SVM* abbreviated to LIBSVM is a library developed by [8] for SVM. Differently from the SVM, the LIBSVM supports both the regression and distribution estimations. The library allows the selection of variants of the SVM algorithm, such as kernels, and its parameters. There are five variations implemented of the algorithm (C-SVM, v-SVM, one-class SVM, epsilon-SVR and v-SVR) and also four types of the kernel: linear, polynomial, Gaussian and Sigmoid [44]. The algorithm has both faster training and testing times [8].

The fifth classifier is the *Wilkes, Stonham, and Aleksander Recognition Device* (or WiSARD). At first, it was proposed to be applied for image-based applications, and later, it was adapted to perform pattern recognition. It is a neural network composed of binary inputs and outputs and implemented with Random Access Memory (RAM) [12]. There are no weights between the nodes, and the neuron functions are stored into tables. The process of adjusting weights consists of changing the table entries, making the algorithm more flexible and efficient. One can perform both supervised and unsupervised learning with this algorithm [34]. As settings used for LIBSVM and WiSARD, they were used as default settings.

A larger set of features in a database can, in theory, provide more information to the classifier. However, if the features are simply chosen without any analysis or consideration to the problem, this hypothesis is more often untrue. This has to do with the presence of some irrelevant or redundant features that can provide a biased model to the learning algorithm [42]. The computational time can also increase exponentially with the growth of the number of features, making the building of the model more difficult [17]. Hence, the selection of the features has been one of the main steps of preprocessing for posterior applications in tasks such as data mining or pattern recognition. Its main purpose is to select a subset of relevant features among all the available features without losing precision [41].

With this aim in mind, we have divided our investigation in two categories of experiments:

- Three filter-based feature selection techniques, which is a method used before the application of the learning algorithm to select a subset of features [17]
- One wrapper feature selection technique, which is characterised by being linked to a method of machine learning in an embedded way. In other words, the accuracy of the classifier is used to evaluate the aptitude of the solutions that were found [17]

In addition, in order to provide more information for our analysis, we have performed a statistical test. For this, we have chosen the non-parametric statistical test *Kruskal-Wallis* [11]. This test verifies if there is a statistical difference between the samples of the same group coming from the same distribution.

We have used also the *k*-means with the best subset of features, according to the accuracy of the classifiers tested to identify if features were sufficiently representative to allow the clustering algorithm to group samples of each individual together, as we will describe in Section 3.3.

3.1 Filter-based approach

The filter approach is a pre-processing step and it makes the feature selection independently from the learning algorithm [42]. This approach tends to select a larger number of attributes in their subsets and have a much lower computational cost compared to a wrapper approach discussed in Section 4 [27].

With the intent of analysing the impact of feature selection on the performance of each classifier when applied to the multimodal database, three different methods of feature selection based on [36] were explored, namely:

- Manual feature selection: It consists of removing each feature from the multimodal database, one at a time, and putting the remaining features that increased the classifiers accuracy rate as the input to the classifiers again.
- Correlation-based feature selector (CFS): It selects a subset of features with high individual predictive ability but low correlation with the other features in the subset, eliminating redundant features the may increase the processing time but not add to the accuracy of the classification.
- RELIEF: It ranks a feature based on how well its values differentiate among the nearest samples. The rationale is that the most discriminative features should have highly different values from those found in samples of different classes, and highly similar values to those found in samples of the same class.

These are three of the most used approaches for filter-based feature selection, and that is the reason we have chosen to investigate their performance in this work [27].

3.2 Wrapper-based approach

As previously mentioned, the wrapper approach is linked to a learning algorithm as a 'black box' serving to analyse every interaction for selecting attributes, which makes it more interesting than the filter approach that requires more computational power. This approach gets lower error rates due to its interaction between the sorting algorithm and the training set. This loop happens for each subset until the stop criterion is reached [43]. For this paper, we have chosen to analyse the performance of a genetic algorithm as the wrapper.

In our context, a binary vector represents a chromosome, where each index indicates the use of the feature (filled with 1) and the non-use of the feature (filled with 0) in the selection process. The vector size is equal to the number of features of the original set. Each chromosome of the initial population is generated randomly using a uniform probability of whether or not each feature is being used.

As operators, we have used the selection, mutation and crossover [30]. The selection operator was performed using the roulette method, where the chromosome with the lowest fitness value had more chance to be chosen. The crossover operator used one cut breakpoint method. This method selects a gene of the chromosome and generates two new chromosomes, the parents. The mutation operator works choosing randomly two chromosome alleles and reversing their values, replacing 0 with 1 and 1 with 0. Finally, the error is used as the fitness function; that is, the lowest value of the fitness function is the best. The parameters of the algorithm are a mutation rate of 3%, the crossover is always done, the population size was 30 individuals and the number of generations was 15.

Both approaches of filter and wrapper will be analysed by a clustering technique in order to make sure we have the correct samples being grouped together when using the respective selected features. Section 3.3 will explain how the *k*-means clustering technique was used.

3.3 The *k*-means algorithm

One possible way of analysing if a set of features is able to represent a class well is to use a clustering algorithm that can group the data following the classes that we already know. In this work, we have chosen to use the *k*-means for this end simply because it is the most popular technique used for clustering [15].

The *k*-means clustering is an algorithm of data partitioning that was first proposed by [32]. It works by partitioning n observations in k clusters defined by centroids where the k is chosen before the algorithm starts. The *k*-means version used in this work differs from the standard version because it initialises the centre of the clusters by *k-means++* as in [4].

In our version of the technique, found in MATLAB [23], we have an initial k cluster centre which is chosen (centroid). Our parameters for the *k*-means algorithm were 76 as the number of clusters, the distance had varied between *Euclidean Distance*, *CityBloc Distance*, *Correlation Distance* and *Cosine Distance* and 500 as the maximum number of algorithm iterations.

The distance from each point to each centroid was calculated by a distance measure. Next, each observation was assigned to the cluster with the closest centroid. New centroids were calculated based on the average of the already grouped observations to obtain k new centroid locations. This process formed new centroids, positioning the observations in other clusters.

4 Results and analysis

Since our main goal is to analyse feature selection and feature-level fusion for two keystroke datasets as well as their combination with online handwritten signature, we have tested if our results were satisfactory by applying a clustering technique to our fused selected features.

Thus, firstly, in order to explore this approach, we have tested the five previously cited classification algorithms in the multimodal database in three different configurations (keyboard + touch, signature + keyboard and signature + touch). Initially, the algorithms had as input the features selected by the filter- or the wrapper-based methods when on their own and a fused feature vector when combined (fused).

Table 4 shows the average accuracy (i.e. percentage of correctly classified instances) of the five classifiers, with their standard deviation and the mean of selected features when trained and tested with the features selected by each technique previously described.

According to the results presented in Table 4, we can observe that the average accuracy of the classifiers was higher with the features selected using the CFS, RELIEF and GA techniques in detriment of the sampling with all the features, indicating that the complete database has conflicting information for the classification algorithms. Only the Manual Selection technique had lower average performance than the complete database, which indicates that some isolated features may not make a difference to the classification, but when they were removed together, some important information was lost.

For the signature + keyboard and signature + touch bases, the classifiers obtained better performance with the subsets of features selected by the wrapper-based technique, GA, with an average difference of 27.83% in accuracy over the complete base, followed by the RELIEF technique. For the keyboard + touch base, the filter-based technique, RELIEF, selected the best subset of features with an average difference of 32.60% in accuracy over the complete base, followed by the GA technique.

Observing the fusion of the selected features for each database, we can verify that in three of the five cases analysed, the fusion of the databases of signature + keyboard resulted in better accuracy: CFS, RELIEF and GA. The fusion of the signature + touch databases obtained better accuracy only with the complete database. The fusion of the keyboard + touch databases obtained better accuracy for the subset of features selected by Relief.

On the other hand, we can see from Table 5 that by analysing the performance of the KNN classifier with subsets of features, it can be verified that its accuracy ranged from 56.71% (Manual) to 71.49% (GA), the accuracy being lower with a greater number of features. Meanwhile, the highest accuracy was with a small subset of only 9 features, supporting our hypothesis that for this classifier, having data that is noisy, irrelevant or redundant, can be damaging to the classification accuracy.

The SVM classifier had a performance ranging from 10.39% (Manual) to 66.67% (CFS), the worst with 40 features and the best with 4 features. It is important to note that with the Manual selection the performance was even lower than with no selection, where the accuracy was 20.61%. In addition, it was observed that the GA selected 16 features for this classifier, and its accuracy was 61.40%. In other words, for this classifier, the filter-based methods CFS and RELIEF selected fewer features, but the chosen features were more relevant for classification (providing higher accuracy).

Table 4 Average accuracy (Acc) rates in percentage, standard deviation (SD) and average of selected features (SF) for all the combinations of signature, touch and keyboard

Feature	Keyboard + touch		Signature + keyboard		Signature + touch	
selection	Acc (SD)	SF	Acc (SD)	SF	Acc (SD)	SF
No selection	28.21 (13.49)	66	45.26 (12.78)	74	45.96 (11.03)	54
Manual	28.63 (25.03)	28	11.31 (7.12)	17	18.24 (13.37)	18
CFS	49.00 (24.95)	3.2	51.31 (24.26)	10	50.00 (23.27)	10
RELIEF	**60.81 (9.80)**	16.2	62.89 (6.41)	18	58.17 (6.79)	12
GA	52.20 (34.04)	27.2	**78.94 (12.81)**	31.6	**67.94 (17.24)**	22

Table 5 Accuracy (Acc) rates in percentage and total of selected features (SF) for all the combinations of signature, touch and keyboard

Feature selection	Algorithm	Keyboard + touch Acc (FS)	Signature + keyboard Acc (FS)	Signature + touch Acc (FS)
No selection	KNN	20.17 (66)	37.71 (74)	43.85 (54)
	SVM	20.61 (66)	38.15 (74)	42.54 (54)
	MLP	28.07 (66)	48.68 (74)	42.54 (54)
	LIBSVM	20.61 (66)	35.52 (74)	35.96 (54)
	WISARD	**51.61 (66)**	**66.22 (74)**	**64.91 (54)**
Manual	KNN	**56.71 (27)**	9.64 (11)	10.08 (11)
	SVM	10.39 (40)	13.59 (5)	**32.89 (13)**
	MLP	54.11 (19)	**21.05 (26)**	16.67 (12)
	LIBSVM	03.50 (20)	01.31 (37)	01.31 (39)
	WISARD	18.42 (35)	10.96 (7)	30.26 (17)
CFS	KNN	**67.53 (4)**	64.47 (10)	62.28 (10)
	SVM	66.67 (4)	08.77 (10)	9.21 (10)
	MLP	41.99 (4)	60.08 (10)	58.77 (10)
	LIBSVM	08.33 (2)	55.26 (10)	53.50 (10)
	WISARD	60.47 (2)	**67.98 (10)**	**66.22 (10)**
RELIEF	KNN	**67.53 (5)**	60.96 (18)	57.01 (12)
	SVM	66.67 (5)	60.96 (18)	58.33 (12)
	MLP	59.31 (5)	60.52 (18)	56.67 (12)
	LIBSVM	44.29 (33)	57.89 (18)	50.00 (12)
	WISARD	66.26 (33)	**74.12 (18)**	**68.85 (12)**
GA	KNN	74.33 (20)	75.26 (25)	77.63 (19)
	SVM	48.02 (29)	63.41 (40)	41.83 (27)
	MLP	33.94 (35)	73.28 (40)	59.60 (32)
	LIBSVM	8.81 (20)	85.92 (22)	76.05 (8)
	WISARD	**95.91 (32)**	**96.84 (31)**	**84.60 (25)**

In turn, the MLP classifier presented an accuracy 63.59% (GA). This classifier demonstrated a different behaviour from the other two, since its performance was better with more features. In the best case, in the GA, 29 features were selected. In this database, this indicated that the MLP neural network needed more information, even though redundant, to perform a good classification.

When we compare the classifiers, we can see that the KNN presents the best results for all the feature selection techniques, concluding that this classifier is best applied in situations where the features are relevant and not redundant or noisy. In turn, despite the fact that the MLP had the best performance when trained and tested with all features, it had the lowest accuracy in most feature selection techniques (CFS, RELIEF and GA). Only in the Manual, where the SVM showed an expressively lower accuracy, this fact did not occur. In other words, the MLP can improve with feature selection but it still needs more features to perform well when compared to other classifiers, which indicates that it is good to remove noisy or irrelevant attributes, but possibly redundancy is important for the classifier, whereas this does not occur with KNN or SVM.

But more importantly, it is interesting to observe that the results from the LibSVM and the WISARD are directly related with the quantity and the quality of the features selected reaching an impressive accuracy of 85% and 96%, respectively. These techniques perform way better than the original implementations of SVM and MLP which now is more than justified to be used in problems that need to investigate feature selection and fusion.

After the classification, we have performed the statistical test to help us find which feature subset selection approach is the best to perform the identity prediction. We have applied the test using a confidence value of 95% (p value = 0.05). A p value $<$ 0.05 implies a statistical difference between the databases, that is, in our case, statistically a database is more accurate than another. Table 6 presents the p value for all combinations

Table 6 p value for all the combinations using keyboard + touch, signature + keyboard and signature + touch

	***p* value**		
Classifier	***keytouch vs sigkey***	***sigkey vs sigtouch***	***sigtouch vs keytouch***
	No selection		
SVM	0	0	< 0.0001
LibSVM	0	0	< 0.0001
KNN	0	0	< 0.0001
MLP	< 0.0001	0	0
WiSARD	< 0.0001	0	0
	Manual		
SVM	0	0	< 0.0001
LibSVM	< 0.0001	0	< 0.0001
KNN	< 0. 0001	0	0
MLP	0	0	< 0.0001
WiSARD	0	< 0.0001	0
	CFS		
SVM	< 0.0001	0	0
LibSVM	< 0.0001	0	0
KNN	0	0	< 0.0001
MLP	< 0.0001	0	0
WiSARD	< 0.0001	0	0
	ReliefF		
SVM	0	0	< 0.0001
LibSVM	< 0.0001	0	0
KNN	0	0	< 0.0001
MLP	0	< 0.0001	0
WiSARD	< 0.0001	0	0
	GA		
SVM	0	0	< 0.0001
LibSVM	0	< 0.0001	0
KNN	0	0	< 0.0001
MLP	< 0.0001	0	0
WiSARD	0	< 0.0001	0

using keyboard + touch (keytouch), signature + keyboard (sigkey) and signature + touch (sigtouch). Using only the combinations where the p value is < 0.05, we have created Table 7 that shows in percentage the number of times that a dataset was more accurate (+) and less accurate (−).

Observing Table 7, we can see that most of the approaches, keyboard + touch and signature + keyboard have achieved the highest accuracies. However, No selection and GA techniques reached the same performances (86.67%), using 66 features (No Selection) and 27 (GA), indicating that the feature subset selected by GA is the best to realise our experiments.

Since the genetic algorithm obtained the best results for most of the observed cases, we have used the k-means algorithm to verify if it is possible to uniquely identify each one of 76 individuals from the selected sets. For this, it was necessary that every triple sample of each user was allocated to a single cluster where each one of them represented an individual. As distance measure, four metrics were used:

- Square Euclidean distance: At this distance, each centroid represents the average of the points.
- Cityblock distance: This distance is obtained by the absolute sum of differences, where each centroid composes the average of the cluster points.
- Correlation distance: This distance is the correlation between one minus the points in the sample. After centralising and normalising the points and standard deviation, the centroids are the average of the cluster points.

Table 7 Accuracy rates when the accuracy was higher (+) and (−) lower

		No selection	
%	Keyboard + touch	Signature + keyboard	Signature + touch
+	33.34	**66.67**	**86.67**
−	**66.66**	33.33	13.33
		Manual	
%	Keyboard + touch	Signature + keyboard	Signature + touch
+	**63.33**	33.34	**53.33**
−	36.67	**66.66**	46.67
		CFS	
%	Keyboard + touch	Signature + keyboard	Signature + touch
+	**63.34**	**63.33**	36.66
−	36.66	36.67	**63.34**
		ReliefF	
%	Keyboard + touch	Signature + keyboard	Signature + touch
+	48.34	**73.33**	33.33
−	**51.66**	26.67	**66.67**
		GA	
%	Keyboard + touch	Signature + keyboard	Signature + touch
+	26.67	**86.67**	36.67
−	**73.33**	13.33	**63.33**

- Cosine distance: The cosine distance obtained by one minus the cosine angle existing between the points represented by vectors. After normalising the points according to the unit Euclidean length, the centroids are the average of the cluster points.

Since we know that the GA selected 9 features for the KNN, 16 for the SVM and 29 for the MLP, and there was a direct relationship between the amount of features and the ability of the *k*-means algorithm to correctly group the samples of each individual. However, this accuracy was relatively low, ranging from 38.59 to 50.63%. By comparing the distance measure adopted in the clustering algorithm (that can be seen in Table 8), we have found that the cityblock obtained the highest accuracy using the features selected by the MLP as well as the 9 features selected for the KNN. Only for the 16 features selected for the SVM is that the cosine distance obtained better accuracy (47.36%).

In this way, we can conclude that for the different distance measures applied to the *k*-means algorithm, the accuracy had a correlation with the quantity of available features. The cityblock distance measure was better able to group the individuals of the database among the other measure distances.

Although we have done very initial experiments with this new three-modal hand-based biometric database in order to investigate effectively the use of feature selection and feature-level fusion, we are indeed able to note that the techniques for feature selection have presented a considerate improvement in accuracy. Moreover, we have the KNN with a very significant accuracy when compared with a traditionally higher performer MLP and using less features.

Table 8 Accuracy rates of *k*-means algorithm

k-means algorithm	Algorithm	Keyboard + touch	Keyboard + signature	Touch + signature
Euclidean distance	KNN	57.01	62.71	57.01
	SVM	57.45	59.64	47.36
	MLP	58.77	43.85	60.52
	LIBSVM	53.07	63.59	65.35
	WiSARD	55.70	67.54	61.40
Cityblock distance	KNN	63.15	64.91	68.85
	SVM	59.64	64.03	50.00
	MLP	59.21	41.22	67.98
	LIBSVM	58.77	72.80	66.66
	WiSARD	62.28	66.66	65.35
Correlation distance	KNN	56.14	64.03	59.21
	SVM	49.12	59.64	46.05
	MLP	57.01	53.07	64.03
	LIBSVM	54.82	65.78	65.78
	WiSARD	42.98	65.78	61.48
Cosine distance	KNN	57.45	66.22	57.89
	SVM	57.01	60.08	46.05
	MLP	55.70	53.94	62.71
	LIBSVM	53.07	65.78	68.42
	WiSARD	54.38	64.03	59.64

5 Conclusion and future work

In this work, we have analysed the impact of two different kinds of feature selection on the performance of five different algorithms when classifying the instances of the multimodal and unimodal biometric hand-based databases. Also, we have tested if our results were satisfactory by applying clustering techniques on our selected features.

When we applied genetic algorithms as a selection technique, we have managed to reduce considerably the quantity of features for both modalities and still keep an acceptable accuracy level (when compared with the three filter techniques). We also confirmed that the selected features did indeed manage to group together a considerable amount of users when we verified it using *k*-means. Also, the classifiers presented a varied performance to each of the feature selection techniques being the LibSVM and the WiSARD the two with the best overall performance.

Through all experiments, it is possible to observe that the performance of the classification on the multimodal database can increase or decrease independently of whether the performance of its internal unimodal databases increases, decreases, or it can show differing behaviours. In such conditions, it is essential to understand the importance of each feature, specially when you have two very similar yet different biometric modalities.

We understand that the main importance of our work is the investigation and exploration of the very first three-modal biometric database that has both touch and keyboard keystroke dynamics and online handwritten signature from the very same users; and how the feature selection can really impact on the performance of traditional classification algorithms when applied at a more realistic dataset.

Acknowledgements
This work was supported by the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES).

Authors' contributions
All authors had an equal input in this paper. The authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The database used in this work is available by contacting the authors.

Competing interests
The authors of this manuscript declare that they have no competing interests.

Author details
[1]Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil. [2]Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, RJ, Brazil. [3]Sheffield Hallam University (SHU), Sheffield, UK.

Received: 13 March 2019 Accepted: 10 August 2020
Published online: 27 August 2020

References

1. Alghamdi SJ, Elrefaei LA (2015) Dynamic user verification using touch keystroke based on medians vector proximity. In: 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks. pp 121–126. https://doi.org/10.1109/cicsyn.2015.31
2. Antal M, Szabo LZ (2015) An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. In: 2015 20th International Conference on Control Systems and Computer Science. pp 343–350. https://doi.org/10.1109/cscs.2015.16
3. Araujo L, Sucupira L, Lizarraga M, Ling L, Yabu-Uti J (2005) User authentication through typing biometrics features. IEEE Trans Sig Process 53(2):851–855
4. Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms:1027–1035
5. Bailey KO, Okolica JS, Peterson GL (2014) User identification and authentication using multi-modal behavioral biometrics. Comput Secur 43:77–89

6. Bergadano F, Gunetti D, Picardi C (2003) Identity verification through dynamic keystroke analysis. Intell Data Anal 7(5):469–496
7. Buschek D, De Luca A, Alt F (2015) Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15. ACM, New York,USA. pp 1393–1402. https://doi.org/10.1145/2702123.2702252
8. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27
9. Chang TY, Tsai CJ, Lin JH (2012) A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices. J Syst Softw 85(5):1157–1165
10. Clarke NL, Furnell SM (2006) Authenticating mobile phone users using keystroke analysis. Int J Inf Secur 6(1):1–14
11. Richardson A (2010) Nonparametric statistics for non-statisticians: A step-by-step approach by Gregory W. Corder, Dale I. Foreman. Int Stat Rev 78(3):451–452
12. Cotta KP, Ferreira RS, França FM (2018) Weightless neural network wisard applied to online recommender systems. In: 2018 7th Brazilian Conference on Intelligent Systems (BRACIS). IEEE. pp 348–353. https://doi.org/10.1109/bracis.2018.00067
13. Da Silva VR, Silva JCGdA, Da Costa-Abreu M (2016) A new Brazilian hand-based behavioural biometrics database: data collection and analysis. In: The 7th IET International Conference on Imaging for Crime Detection and Prevention (ICDP-16). p 1. https://doi.org/10.1049/ic.2016.0085
14. Darabseh A, Namin AS (2015) On accuracy of classification-based keystroke dynamics for continuous user authentication. In: 2015 International Conference on Cyberworlds (CW). IEEE. pp 321–324. https://doi.org/10.1109/cw.2015.21
15. Dubey A, Choubey A (2017) A systematic review on k-means clustering techniques. Int J Sci Res Eng Technol (IJSRET, ISSN 2278–0882) 6(6)
16. El-Abed M, Dafer M, Khayat RE (2014) RHU Keystroke: a mobile-based benchmark for keystroke dynamics systems. In: 2014 International Carnahan Conference on Security Technology (ICCST). pp 1–4. https://doi.org/10.1109/ccst.2014.6986984
17. Faceli K, Lorena AC, Gama J, Carvalho ACPLF (2011) Artificial intelligence: a machine learning approach, LTC, Rio de Janeiro
18. Fairhurst MC, Abreu MCC (2009) Balancing performance factors in multisource biometric processing platforms. IET Signal Process 3(4):342–351
19. Fayyaz M, Hajizadeh_Saffar M, Sabokrou M, Fathy M (2015) Feature representation for online signature verification. arXiv preprint arXiv:1505.08153
20. Ghayoumi M (2015) A review of multimodal biometric systems: fusion methods and their applications. In: Computer and Information Science (ICIS) 2015 IEEE/ACIS 14th International Conference on. IEEE. pp 131–136. https://doi.org/10.1109/icis.2015.7166582
21. Griswold-Steiner I, Matovu R, Serwadda A (2017) Handwriting watcher: a mechanism for smartwatch-driven handwriting authentication. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE. pp 216–224. https://doi.org/10.1109/btas.2017.8272701
22. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I (2009) The WEKA data mining software: an update. SIGKDD Explor 11(1):10–18
23. Higham DJ, Higham NJ (2016) MATLAB guide. SIAM
24. Iranmanesh V, Ahmad SMS, Adnan WAW, Yussof S, Arigbabu OA, Malallah FL (2014) Online handwritten signature verification using neural network classifier based on principal component analysis. Sci World J 2014
25. Jain A, Ross A, Pankanti S (2006) Biometrics: A tool for information security. IEEE Trans Inf Forensics Secur 1(2):125–143. https://doi.org/10.1109/TIFS.2006.873653
26. Jeanjaitrong N, Bhattarakosol P (2013) Feasibility study on authentication based keystroke dynamic over touch-screen devices. In: 2013 13th International Symposium on Communications and Information Technologies (ISCIT). IEEE. pp 238–242. https://doi.org/10.1109/iscit.2013.6645856
27. Kawamura A, Chakraborty B (2017) A hybrid approach for optimal feature subset selection with evolutionary algorithms. In: 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST). pp 564–568. https://doi.org/10.1109/icawst.2017.8256521
28. Killourhy KS, Maxion RA (2009) Comparing anomaly-detection algorithms for keystroke dynamics. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks. pp 125–134. https://doi.org/10.1109/dsn.2009.5270346
29. Koong CS, Yang TI, Tseng CC (2014) A user authentication scheme using physiological and behavioral biometrics for multitouch devices. Sci World J 2014
30. Kumar M, Husian M, Upreti N, Gupta D (2010) Genetic algorithm: review and application. Int J Inf Technol Knowl Manag 2(2):451–454
31. Li Y, Zhang B, Cao Y, Zhao S, Gao Y, Liu J (2011) Study on the BeiHang keystroke dynamics database. In: 2011 International Joint Conference on Biometrics. pp 1–5. https://doi.org/10.1109/ijcb.2011.6117485
32. Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137
33. Loy CC, Lai WK, Lim CP (2007) Keystroke patterns classification using the ARTMAP-FD neural network. In: Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007). IEEE. pp 61–64. https://doi.org/10.1109/iih-msp.2007.218
34. Ludermir TB, Carvalho A, Braga AP, Souto M (1999) Weightless neural models: a review of current and past works. Neural Comput Surv 2:41–61
35. Mahmood Al-Obaidi N, Al-Jarrah MM (2016) Statistical median-based classifier model for keystroke dynamics on mobile devices. In: 2016 6th International Conference on Digital Information Processing and Communications (ICDIPC). IEEE. pp 186–191. https://doi.org/10.1109/icdipc.2016.7470816
36. Mendes BVS (2017) Analysis of feature selection on the performance of multimodal keystroke dynamics biometric systems Tech. rep

37. Morales A, Falanga M, Fierrez J, Sansona C, Ortega-Garcia J (2015) Keystroke dynamics recognition based on personal data. In: 2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS). pp 1–6. https://doi.org/10.1109/btas.2015.7358772
38. Ngugi B, Kahn BK, Tremaine M (2011) Typing biometrics. J Data Inf Qual 2(2):1–21
39. Sharif M, Raza M, Shah JH, Yasmin M, Fernandes SL (2019) An Overview of Biometrics Methods. In: Handbook of Multimedia Information Security: Techniques and Applications. Springer, Cham. pp 15–35
40. Polemi D (1997) Biometric techniques: review and evaluation of biometric techniques for identification and authentication, including an appraisal of the areas where they are most applicable. Reported prepared for the European Commision DG XIIIC:1–41
41. Santana LEAS (2012) Optimization classifiers committees: an approach based on filter for selecting subsets of attributes, Ph.D. thesis. Universidade Federal do Rio Grande do Norte, Natal
42. Santana LEAS, Canuto AMP (2014) Filter-based optimization techniques for selection of feature subsets in ensemble systems. Expert Systems with Applications 41(4, Part 2):1622–1631
43. Shen J, Xia J, Zhang X, Jia W (2017) Sliding block-based hybrid feature subset selection in network traffic. IEEE Access 186(18):179–18. https://doi.org/10.1109/ACCESS.2017.2750489
44. Sun F, Belatreche A, Coleman S, McGinnity YLM, Li Y, et al (2012) Evaluation of LibSVM and mutual information matching classifiers for multi-domain sentiment analysis. In: The 23rd Irish Conference on Artificial Intelligence and Cognitive Science, Dublin
45. Taouche C, Batouche MC, Berkane M, Taleb-Ahmed A (2008) Multimodal biometric systems overview. Electronics 49(3):39–44
46. Vural E, Huang J, Hou D, Schuckers S (2014) Shared research dataset to support development of keystroke authentication. In: IEEE International Joint Conference on Biometrics. pp 1–8. https://doi.org/10.1109/btas.2014.6996259
47. Wang S, Chen C, Yang W, Hu J (2015) Mutual dependency of features in multimodal biometric systems. Electron Lett 51(3):234–235
48. Xia X, Song X, Luan F, Zheng J, Chen Z, Ma X (2018) Discriminative feature selection for on-line signature verification. Pattern Recognition 74:422–433
49. Yang L, Cheng Y, Wang X, Liu Q (2018) Online handwritten signature verification using feature weighting algorithm relief. Soft Comput 22(23):7811–7823
50. Zheng N, Bai K, Huang H, Wang H (2014) You are how you touch: user verification on smartphones via tapping behaviors. In: 2014 IEEE 22nd International Conference on Network Protocols. pp 221–232. https://doi.org/10.1109/icnp.2014.43

Publisher's Note