Knowledge-Based Systems 244 (2022) 108544

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Combining user behavioural information at the feature level to enhance continuous authentication systems

Alejandro G. Martín *, Isaac Martín de Diego, Alberto Fernández-Isabel, Marta Beltrán, Rubén R. Fernández

Rey Juan Carlos University, Department of Computing, ETSII, C/ Tulipán, s/n, 28933, Móstoles, Madrid, Spain

ARTICLE INFO

Article history:
Received 10 December 2021
Received in revised form 9 February 2022
Accepted 3 March 2022
Available online 15 March 2022

Keywords:
Anomaly detection
Behavioural information combination
Continuous authentication
User and Entity Behaviour Analytics

ABSTRACT

The scientific and business communities are proposing new authentication methods more robust than traditional solutions relying on a single security point such as passwords (i.e. "something you know"). User and Entity Behavior Analysis (UEBA) has postulated as an excellent solution to improve authentication systems by performing continuous authentication to extend the authentication process over time. UEBA is based on detecting anomalies in the intrinsic behaviour of each user or entity (i.e. it is based on "something you are/do"). This paper presents a method for performing continuous authentication using UEBA techniques that allows combining information from multiple sources at the feature level. This combination is achieved through a novel Symbolic Aggregate approximation (SAX) using Random Trees Embeddings for each information source, producing a sequence of symbols. Then, these sequences of symbols are combined into a single sequence using temporal information. The resulting sequence of symbols feeds a density-based clustering model that uses a distance based on DNA sequence alignment techniques to extract behavioural cores. Finally, new samples are compared against these cores to detect anomalies using a risk model that evaluates if a behaviour is anomalous (suspected user impersonation). The model has been extensively tested and evaluated against well-known state-of-the-art datasets.

1. Introduction

The theft of credentials using phishing, or session hijacking, are only two examples of spoofing or impersonation attacks that allow adversaries to access personal information or to act in the name of a legitimate user. Applications and services usually have a single point of authentication where users enter their passwords. From that moment onward, the system trusts that the users are whom they claim to be. Sometimes, these applications and services include a second authentication factor, by sending the user a One-Time Password (OTP) to the smartphone or by using a hardware token to perform the initial login or to perform high-risk activities [1]. However, a session hijacking attack, among other patterns, can make these additional security mechanisms insufficient, and they should be complemented with continuous authentication systems to provide the desired security levels.

Continuous authentication consists of extending the authentication process over time, not considering it a one-time action performed just once at the beginning of the session. This enables the systems to authenticate a user as many times as necessary during the session. An effective way of performing continuous authentication is analysing behavioural information [2]. Integrating this behavioural information into systems increases the security by moving from a one-time use of "something the user knows" (passwords) or "something the user has" (OTP on a smartphone or hardware token) to a continuous checking of "something the user is or does" (the exhibited behaviour), making it even more difficult to impersonate a user.

UEBA is a scientific discipline that appears as a good solution for improving authentication systems. It is focused on understanding, modelling and predicting past, present and future behaviours of users and entities [3]. UEBA relies on Machine Learning (ML) to define and extract behavioural features, characterise the normal behaviours for each user, and detect anomalous behaviours that could indicate a cyber-attack. Subsequently, this problem can be framed as an anomaly detection problem [4].

Moreover, thanks to the evolution of technology and the increase of computational resources, the use of any device, service, or application generates a multitude of new behavioural dynamics that can be analysed. Thus, keystroke dynamics, mouse

* Corresponding author.
E-mail addresses: alejandro.garciam@urjc.es (A.G. Martín), isaac.martin@urjc.es (I. Martín de Diego), alberto.fernandez.isabel@urjc.es (A. Fernández-Isabel), marta.beltran@urjc.es (M. Beltrán), ruben.rodriguez@urjc.es (R.R. Fernández).

https://doi.org/10.1016/j.knosys.2022.108544

dynamics, the use of sensors in smart devices such as smartphones or even the dynamics of the sensors themselves with the advent of Internet of Things (IoT) are in the spotlight today. These new behavioural dynamics are integrated into solutions for companies or even for the end-user to improve the security of technological systems [5,6]. In addition, these solutions are starting to be used in federated identity management solutions, which further increases the interest in them [7].

Several approaches analyse the best behavioural features to use depending on the information source (keystrokes, mouse interactions, smartphone sensors, etc.) or evaluate different ML models to identify the one showing the best performance. However, few proposals combine information from different sources due to the complexity of the problem, despite the consolidated benefits in terms of efficiency, effectiveness, and performance [8].

The combination of this behavioural information can be performed at two levels [9]. First, at the decision level, a ML model is generated for each of the information sources. Then, the outputs of these models are combined to give a prediction at a given time. Second, at the feature level, the behavioural characteristics extracted from each information source are combined to feed a unique ML model. Thus, the interactions when a user switches from one information source to another can also be considered simultaneously.

The main motivation of this work is to increase the security levels that any agent (i.e. service or application) provides to end-users in terms of continuous authentication, using UEBA techniques. More specifically, nowadays, there are some approaches in this area to solve continuous authentication. However, there is still a long way to go, especially using information combination techniques to develop more accurate methods for better integration in real environments.

This paper proposes a method to accomplish continuous authentication that allows combining user behavioural information at the feature level. Furthermore, this method is evaluated through multiple experiments on well-known state-of-the-art datasets obtaining promising results. The main contributions of this paper are summarised as follows:

- The combination of information from heterogeneous information sources using a novel representation of behavioural information.
- The proposal of DNA sequence alignment techniques to calculate similarities between behavioural dynamics.
- The proposal of density-based clustering techniques to extract behavioural cores.
- The development of a risk model based on the proposed techniques to detect behavioural anomalies and enable continuous authentication.

Thus, this method provides an agent with the necessary mechanisms to use behavioural analysis combining information gathered from heterogeneous data sources to improve end-user security levels. This is in line with passwordless authentication methods, which are currently being developed and implemented by academia and companies such as Microsoft and Google [10]. A clear use case implementing these methods can be seen in [7]. This approach proposes a workflow to integrate methods such as the one presented here, in the main federated identity management schemes like OpenID Connect [11], which are becoming today the most used architectures to perform identity management.

The rest of the paper is organised as follows. Section 2 presents an overview of the related work. Section 3 details the proposed method. Section 4 addresses the experiments and analyses the obtained results describing the strength and weaknesses of the proposal. Finally, Section 5 concludes and introduces future lines of research.

2. Related work

This section introduces the approaches from the literature related to the proposal. The adopted UEBA techniques, the ML models and the feature extraction processes are exhibited. Three categories are analysed depending on the selected information source. First, the keystroke dynamics, which compile behavioural information retrieved from the keyboard interactions made by users. Then, the mouse dynamics, which use behavioural information retrieved from the mouse movements. Subsequently, it is addressed the combination of behavioural information retrieved from multiple sources. Lastly, the main challenges addressed related to the proposal are analysed.

2.1. Keystroke dynamics

The analysis of behavioural information for authentication purposes began in the 1980s with the study of keystroke dynamics [12]. A multitude of works arose that tried to solve the problem using Bayesian qualifiers [13], Neural Networks (NNs) [14] and clustering techniques [15].

These models have been evolving, and new techniques and more exhaustive experiments continue to be carried out. For instance, in [4], 14 ML detectors, including Mahalanobis distance, Manhattan distance, K-Nearest Neighbors (KNN), NNs, K-means, Fuzzy logic, and SVM, are compared. In [16], an Ant Colony (AC) is used to perform a feature selection step, allowing choosing non-conventional features in the scope. Later, an SVM is implemented, obtaining quite good results. In [17], a novel method to extract features adaptively for each user to feed later a Gaussian density estimation model, Parzen window density estimation, One-Class SVM (OC-SVM), KNN, and K-means is accomplished. In [18], it is shown that heterogeneous (i.e. non-aggregated) feature vectors are more discriminating in distinguishing keyboard dynamics. Subsequently, they use Naïve Bayes (NB), Tree Augmented Naïve Bayes (TANB), KNN, and ridge logistic regression to detect impostors obtaining much better results by combining both types of vectors. In [19], the feature vectors are transformed into frequency spectrograms to transform both frequency and time data into an image. Then, a Gauss–Newton-based Neural Network classifier is used to classify each image into genuine users or impostors. In [20], convolutional and recurrent NNs are combined to build a model for the Buffalo Dataset [21] that obtains encouraging results. In [22] a Kernel Density Estimation (KDE) is considered and compared to other well-known state-of-the-art algorithms using the Clarkson, Torino, and Buffalo Datasets. In [23], an Instance-based Tail Area Density (ITAD) metric is proposed to reduce the number of keystrokes required to authenticate users. This method improves efficiency and reduces latency.

In the case of the feature extraction process for keystroke dynamics, it can be organised into two types of events that are retrieved for each user interaction [24]. Firstly, the *KeyDown* event, which contains the timestamp of when the keystroke is initiated, and secondly, the *KeyUp* event, which includes the timestamp when the key is released. Thus, the interactions are joined together with a sliding window of size 2, forming di-graphs. Note that each di-graph is formed by four events that include four timestamps (for each interaction a *KeyUpX* and *KeyDownX*, where X represents the number of interactions). In this way, six features are usually extracted:

1. H1 (*keyUp1-KeyDown1*): Time elapsed for the first interaction.
2. H2 (*KeyUp2-KeyDown2*): Time elapsed for the second interaction.

3. RP (*KeyDown2-KeyUp1*): Time elapsed between the release of the first interaction and the initiation of the second.
4. PP (*KeyDown2-KeyDown1*): Time elapsed between the initiation of the two interactions.
5. RR (*KeyUp2-KeyUp1*): Time elapsed between the release of the two interactions.
6. PR (*KeyUp2-KeyDown1*): Time elapsed between the initiation of the first interaction and the release of the second.

2.2. Mouse dynamics

The analysis of mouse dynamics for authentication purposes emerged in the 2000s [25]. In [26], movement speed, movement direction, action type, travelled distance, and elapsed time features are used to feed a NNs obtaining good results. In [27], convolutional NNs, recurrent NNs, and a hybrid model which combines both types of NNs using the layer-wise relevance propagation algorithm are used to detect impostors across two well-known state-of-the-art datasets. In [28], mouse features are categorised into holistic and procedural. Then, the performance for the authentication task of both types of features is compared through an OC-SVM. The research in [29] uses Progress-Adjusted Dynamic Time Wraping (PADTW) algorithm, along with a segmentation algorithm to tune the features that later feed an SVM classifier. In [30], the mouse dynamics are converted to images using their mapping function to perform data augmentation then. Later, these images feed a convolutional NNs.

Regarding the mouse dynamics feature extraction process, each mouse interaction is formed by the coordinates, timestamps, and events. The coordinates represent the x and y pixel positions on the screen. The timestamp represents the time at which the event occurs. Regarding the retrieved events, the most common ones are the *MouseMovement* and the *MouseClick*. In this way, as in keyboard dynamics, interactions are linked via a sliding window to form di-graphs. Thus, the elapsed time, distance, velocity, angle, and angular velocity are usually calculated. Nevertheless, these di-graphs usually have high variability, causing many users to generate very similar values. For this reason, a series of di-graphs spaced in time are usually grouped to form strokes [31]. Thus, a mouse stroke is an aggregation of di-graphs where more aggregated features can be calculated. The length of a stroke can be selected according to a time interval (e.g. every 5 s) or by the occurrence of a given event (e.g. from one *MouseClick* to the next one). The mean, median, maximum, minimum, and standard deviation (among other aggregation measures) can be calculated for each one of the above features. Moreover, velocities, distances, or the number of di-graphs for each direction (i.e. categorising the angle feature) can also be calculated.

2.3. Combination of behavioural information

A new line of research is currently emerging that attempts to combine data from multiple information sources simultaneously. Generally speaking, there are two ways to combine information: at the decision and feature levels. Both ways of combining information have a multitude of advantages over the use of a single dynamic. For example, it allows detecting impostor users over a wider time spectrum. In [32], the basis for behavioural biometrics information fusion are established. They addressed the feature level and decision level information fusion for face, fingerprint, and hand verification, improving the results of previous works.

At decision level combination, the objective is to have an independent model for each information source. Subsequently, the predictions of all models are combined to give a single prediction at a given time. In [33], it is implemented a Trust Model (TM) based on combining the outputs of different ML models for keystroke and mouse dynamics using weights adjusted by genetic algorithms. The models used are NNs and Counter-Propagation Artificial Neural Networks. A SVM is used to achieve the combination of both models. It is also proposed another scenario in which they used different distance metrics to achieve their objective without using impostor data in the training phase. In [34], a combination of NB to map each behavioural dynamic to the decision space is proposed. Then, a SVM for the classification task is considered. In [35], a Bayesian network (BN) is trained for each behavioural dynamic, and then Bayes Fusion Scheme (BFS) is used to combine the outcomes of each independent model. In [36], RF, SVM, Decision Trees (DTs), and BN are evaluated for the same purposes. In [8], it is proposed to combine session context information with behavioural features to enhance authentication performance using the The Wolf of SUTD (TWOS) dataset [24]. To achieve this, first, a model that predicts using only session context information is implemented. Then, a model that combines both keyboard and mouse behavioural features during a session is developed. The combination of both models is accomplished using three different methods; a Parametric Linear Combination (PLC), a RF classifier, and an SVM classifier.

Regarding the feature level combination, in [37] it is used a Multi-kernel Learning Method (MKL) for combining keystroke and mouse dynamics features, and it is evaluated against DT, RF, NB, OC-SVM, and SVM models obtaining very promising results. In [9], it is compared both types of combinations. First, for the decision level, it is used the BN for keystroke dynamics and SVM for mouse dynamics to perform an ensemble J48 decision tree model that combines the obtained results. Regarding the feature level combination, Principal Component Analysis (PCA) is used to train and test the performance of BN, J48, and SVM models independently. Note that this approach combines keystroke, mouse movements, and the graphical user interfaces interactions sources. Moreover, the fusion at the feature level has also been accomplished recently using smartphone sensors, obtaining excellent results and providing a new line for researchers improving the results, again, of previous works in the scope [38]. This reaffirms that combining information is a major advance in improving the performance of these types of behavioural-based authentication systems.

The method proposed in the present paper combines information at the feature level. RTEs are used to transform the behavioural features extracted of all available information sources into comparable sequences of characters achieving a novel SAX. Then, DNA sequence alignment techniques are used to measure the distance between these sequences. Subsequently, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is used to obtain behavioural cores for each user. Finally, a risk model is developed to measure new samples with the behavioural cores previously extracted.

In this sense, the present work combines information at the feature level. This type of combination is uncommon in the approaches of the state-of-the-art, although it brings important advantages. Examples of these advantages include increased accuracy in detecting security breaches and system up-time, which allows security breaches to be detected over a longer period of time. In addition, to the best of the author's knowledge, DNA sequence alignment techniques have not been used to detect anomalous behaviour in the area of continuous authentication. The use of this type of techniques opens up new lines of future research that can, as the proposed method does, increase the performance in detecting anomalous behaviour for some scenarios. The use of density-based algorithms to filter information from trusted information sources is another novel aspect. This technique improves the method's efficiency by establishing specific behavioural cores that narrow down the search for atypical behaviours when training and predicting. Moreover, the method is

Table 1
Comparison of previous works. *C* refers to context information. *KD* and *MD* represent keystroke dynamics and mouse dynamics respectively. *RM* is risk model.

Work	Behavioural dynamic	Method	Combination level	Dataset	Free interaction
[4]	KD	14 classifiers	–	.tie5Roanl	No
[16]	KD	DT+SVM+AC	–	Own	Yes
[17]	KD	Gauss+Parzen+ OC-SVM+k-NN+K-means	–	Own	Yes
[18]	KD	NB+TANB+KNN+RLR	–	Own	No
[19]	KD	Spectograms+NNs	–	Own	No
[20]	KD	NNs	–	Buffalo	Yes
[22]	KD	KDE	–	Buffalo+Clarkson+Torino	Yes
[23]	KD	ITAD	–	Buffalo	Yes
[26]	MD	NNs	–	Own	Yes
[27]	MD	NNs	–	Balabit+TWOS	Yes
[28]	MD	OC-SVM	–	Own	No
[29]	MD	PADTW	–	Own+[28]	No
[30]	MD	NNs	–	Balabit	Yes
[33]	KD+MD	TM+NNs+SVM	Decision	Own	Yes
[34]	KD+MD	NB+SVM	Decision	Own	Yes
[35]	KD+MD	BN+BFS	Decision	Own	Yes
[36]	KD+MD	RF+SVM+DT+BN	Decision	Own	Yes
[8]	KD+MD+C	RF+SVM+PLC	Decision	TWOS	Yes
[37]	KD+MD	MKL+DT+RF+NB+ OC-SVM+SVM	Feature	Own	Yes
[9]	KD+MD	BN+J48+SVM	Decision+Feature	Own	Yes
Our	KD+MD	RTE-SAX+DNA-SA+ DBSCAN+RM	Feature	[39]+TWOS	Yes

scalable, that is, the method can be adapted to consider more information sources by simply increasing the number of characters to be used during the SAX process.

On the other hand, the main disadvantage of the proposed method consists of the development of the method may be more complex, which may lead to overhead when implementing it. In addition, the use of DNA sequence alignment techniques may introduce some added latencies that can be addressed by using parallel processing techniques.

Table 1 shows the most relevant works related to the current approach. Those proposals that only use one of the dynamics (i.e. keystroke or mouse dynamics) are bounded due to their high number in the scope. Few works that combine information using other biometric information or behavioural dynamics are also considered. As it can be seen, the number of approaches that combine information sources is scarce.

3. Proposed method

In this section, the proposed method to perform continuous authentication is detailed. It has been specifically designed to combine temporal data from heterogeneous information sources. Furthermore, the method does not make assumptions regarding the temporal distribution of the data. That is, it does not require the information to have a fixed frequency or pattern.

3.1. Research challenges and overview

The analysis of the related work allows us to identify some challenges in the area. The main one is to obtain an accurate representation of behavioural information from heterogeneous information sources. This representation may be based on categorising the information using multivariate SAX. Note that SAX allows representing temporal information into sequences of symbols, but any process of discretisation induces a loss of information. In this way, improving the obtained representation will lead to better results.

A novel technique to archive multivariate SAX using RTEs is proposed. This technique is based on using a tree-based classifier to perform the discretisation process. This technique is selected since tree-based techniques have been demonstrated to outperform other techniques such as clustering techniques (e.g. K-means) and also require less computational time [40].

Another challenge detected is to achieve an accurate comparison between behavioural dynamics. Note that, thanks to SAX, these behavioural dynamics are represented into sequences of symbols. Thus, this challenge can be translated as a problem of comparison between character sequences. Multiple distance metrics could be used in this scope, such as the Levenshtein distance or the Hamming distance [41]. Nevertheless, it is required to detect intrinsic patterns in the sequences in this proposal. In this way, DNA sequence alignment techniques have been demonstrated to fit the desired requirements [42].

Finally, the last challenge is to reduce response time, given the performance requirements of the domain application. To achieve this, the number of samples to be considered by the model is reduced. The idea behind this approach is that the method only considers the information relevant to the model (i.e. behavioural cores), and therefore discards the non-relevant information that introduces unnecessary latencies when training and predicting. Any clustering technique is suitable for this purpose. However, density-based clustering techniques easily determine which are the relevant clusters based on the distribution of the distances of the samples. Therefore, the DBSCAN [43] algorithm has been selected in this proposal.

The whole method is illustrated in Fig. 1. The method is composed of four main tasks. The first task combines the information from each information source into a sequence of n-grams. This task is described in Section 3.2 and it is based on a novel SAX implementation using RTEs. Then, the next task, detailed in Section 3.3, constructs a distance matrix from the n-grams using DNA alignment techniques. This distance matrix is used to train a density-based clustering model to define the behavioural cores of each user in Section 3.4. Next, the risk model, described in Section 3.5, detects anomalous behaviour by weighting new samples against the cores of each user. Finally, in Section 3.6 it is detailed how to set the necessary parameters to achieve reproducibility.

3.2. Representation of information from multiple sources

First, the raw data coming from the keyboard and mouse interactions are processed into features. As previously stated, the

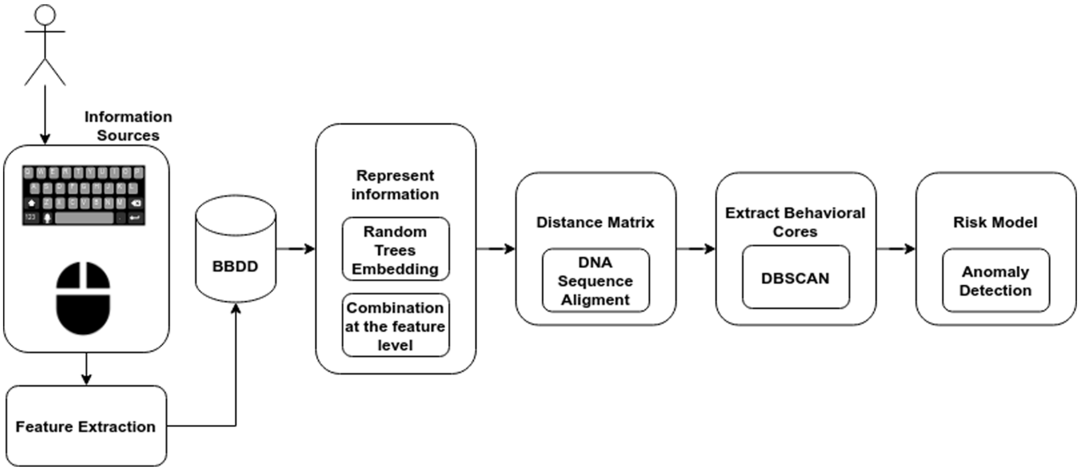


Fig. 1. Overview of the proposed method for combining behavioural information.

keystroke dynamics are grouped into di-graphs. In this way, H1, H2, RP, PP, RR, and PR are calculated (see Section 2.1). Regarding the mouse dynamics, strokes of length 5 s are selected based on practical experience. The number of interactions in the stroke is retrieved. Also, the mean, minimum, maximum, and standard deviation values of the elapsed time, distance, velocity, angle and angular velocity features are calculated. Thus, a vector of 21 features is obtained for each mouse stroke (i.e. the number of iterations in the stroke and four dispersion measures calculated for each one of the five features).

Once the features have been extracted and selected, the next task is to transform them into an appropriate representation. In this proposal, an approach to combine temporal data from heterogeneous sources (e.g. keyboard and mouse) into a tabular representation suitable for traditional ML models is presented. The new combination procedure is based on multivariate SAX. In particular, it is implemented using RTEs [44,45]. RTEs consist of a vector that indicates, for each tree, the number of the leaf where an observation belongs. That is, the position n in the embedding vector indicates the number of the leaf (counting from left to right) where the observation falls in the DT number n. Alternatively, this embedding vector is often represented by using a binary vector whose length is the number of leaves in the RTE and whose value is 1 if the observation falls in that leave and 0 otherwise.

The embeddings from the RTEs are mapped into symbols to ease the comparisons and processing. However, the number of required symbols for each RTE grows exponentially with its number of leaves. Despite this fact, most symbols will rarely occur. Consequently, this limitation can be overcome by using the symbols based on the number of occurrences of that embedding in the training set and assigning the "rare" embeddings to a particular symbol. For example, the embedding with more occurrences gets character *A*, the second *B*, and so on, until all symbols are used. Thus, if two samples have fallen in the same leaves, they are assigned to the same symbol. When all the determined symbols are used, the remaining embeddings are mapped to an arbitrary special symbol.

The process explained above is executed for each available information source. That is, each information source has its RTE and unique set of symbols. The information extracted from the keyboard is represented by using uppercase symbols and lowercase in the case of the mouse. Once all the data from the information sources are converted into sequences of symbols, they are merged based on the temporal information of each symbol (e.g. when a particular key was pressed or the mouse was moved). Finally, the merged sequence of symbols is converted in n-grams whose size is defined by the parameter $n - gramlength$ (ngl). The feature combination process for multiple information sources is illustrated in Fig. 2.

3.3. Distance matrix generation

This task is focused on obtaining a distance matrix that represents the dissimilarities between the behavioural dynamics generated by each user. Precisely, the pairwise distances between the n-grams that represent the behaviour of a user is calculated. This results in a NxN matrix where N is the number of n-grams for a specific user.

In the present method, the distance between two n-grams is calculated using DNA sequence alignment techniques [42], more precisely, the global sequence alignment algorithm. This algorithm is applied to all available sequences for each user in pairs. The alignment algorithm returns a score denoting the similarity between two sequences taking into account the length of the target sequence and the obtained match, miss and gap penalty scores. Then, this value is transformed into distances by first normalising it in the range [0, 1], with 1 being the maximum similarity value (i.e. a sequence compared to itself) and 0 being the minimum. Subsequently, 1-*similarity score* is applied to obtain the distance value. The distance value obtained is also in the range of values [0, 1], being 0 the minimum distance (i.e. a sequence compared with itself) and 1 the maximum distance.

3.4. Extracting the behavioural cores

Once the distance matrix has been calculated, the next task is to define the core behaviour of each user. Correctly identifying these behavioural cores is one of the most critical tasks to be able to perform continuous authentication since it is the source of knowledge that will be used to compare the new samples arriving. Most of the state-of-the-art works use all the information available for each of the users. However, in the users' behaviour, there may be outliers that can weigh down the classifier's performance. In addition, using all the available data will also generate latency when training and predicting since each sample has to be compared with a more extensive set of observations, making the classifier less efficient. Thus, the ideal would be to obtain behavioural cores as small as possible, as long as these cores correctly represent the user's behaviour so that the classifier can generalise correctly.

In this proposal, the DBSCAN algorithm [43] is used to obtain the behavioural cores for each user. In order to train the DBSCAN,

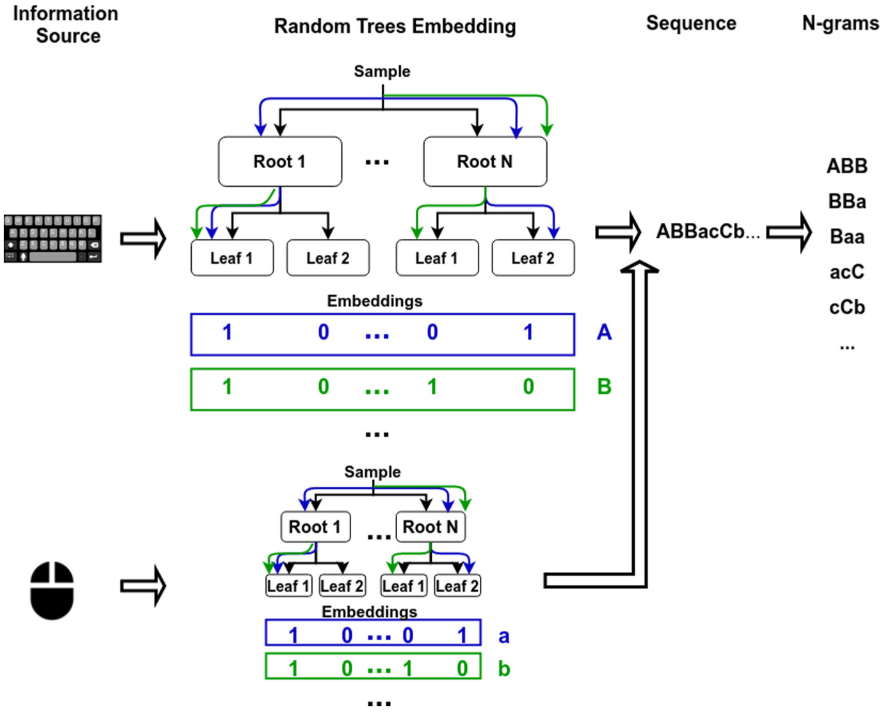


Fig. 2. Summary of the SAX process using RTEs for multiple information sources.

two hyper-parameters have to be fixed. First, the maximum distance between two samples to be considered as neighbours. This hyper-parameter is named *EPS* and is the most critical parameter of DBSCAN. Secondly, the minimum number of points that a neighbourhood must have to be considered as a cluster. This hyper-parameter is named *MINS*. Note that DBSCAN does not need information about the number of desired clusters. This is consistent with the fact that a priory, it is impossible to know how many behavioural cores (i.e. clusters) will be obtained for a specific user.

The distance matrix (which contains only genuine behaviour) feeds the DBSCAN. The high-density regions obtained represent the core behaviour, while the low-density regions represent atypical behaviours performed by the user. Atypical behaviours are discarded (for the moment, but not removed) so that new observations cannot be compared.

In Fig. 3, an excerpt of the cores extracted (blue points) and atypical behaviours (red points) for a specific user is illustrated. These points are represented using the first two components of Multidimensional Scaling [46]. It can be seen that as the points move away from the central core (displayed at the bottom), the distribution becomes more heterogeneous (i.e. it contains more outliers). This corroborates that DBSCAN is correctly discarding behaviours that deviate from the assumed distribution for this user.

Above, it was mentioned that the atypical behaviours are discarded to establish the core behaviour, but they are not removed. This is due to combat the ageing behaviour (i.e. the behavioural habits of the same user can be modified over time [47]). For example, this is caused by users becoming accustomed to using a system in the way they consider optimal or changing the system interface, among others. Therefore, continuous authentication systems must retrain or update their parameters over time in order to continue to maintain their performance. Thus, these atypical behaviours may not be a behavioural core at a given time but may form a behavioural core in the future.

3.5. Risk model

At this point, the only thing remaining is to predict the new observations. A risk model is built with this objective. First, these new observations are processed in order to obtain the sequence of n-grams. Then, a vector of distances is calculated for each new sequence. This vector contains the distance between the new sequence and all the sequences retrieved in the behavioural cores.

The risk associated with the observation is calculated as the mean of the retrieved vector distance. A low-risk value denotes that the new sequence is close to the cores and, therefore, it is likely to belong to the genuine user. In the opposite case, a high-risk value is likely to correspond to an impostor user. This allows comparing each new observation individually. However, multiple samples will arrive, and they do so in a temporally ordered manner. The obtained risk values are sorted temporally to analyse the changes in risk over time, building a risk buffer (see Fig. 4(a)). As can be seen, the risk value is very changeable over time, resulting in many false positives and false negatives. For this reason, the risk curve is smoothed calculating the EMA as follows:

$$EMA_t = \begin{cases} Y_1, & \text{if } t = 1 \\ \alpha Y_t + (1-\alpha) \cdot EMA_{t-1}, & \text{if } t > 1 \end{cases}$$

where EMA_t is the exponential moving average at time t, Y_t is the value at a given time t, and α is the smoothing coefficient between [0, 1]. A low value of α weights higher older observation, while high values of α overpasses older observations faster.

The α coefficient is usually calculated according to the number of observations to be taken into account [48]. In this case, it is calculated based on the *WinSize* parameter as follows:

$$\alpha = 2/(WinSize + 1)$$

where *WinSize* is the number of values in the risk sequence. Once EMA is applied, the values change from being highly changeable to being more stable over time (see Fig. 4(b)). This means that a

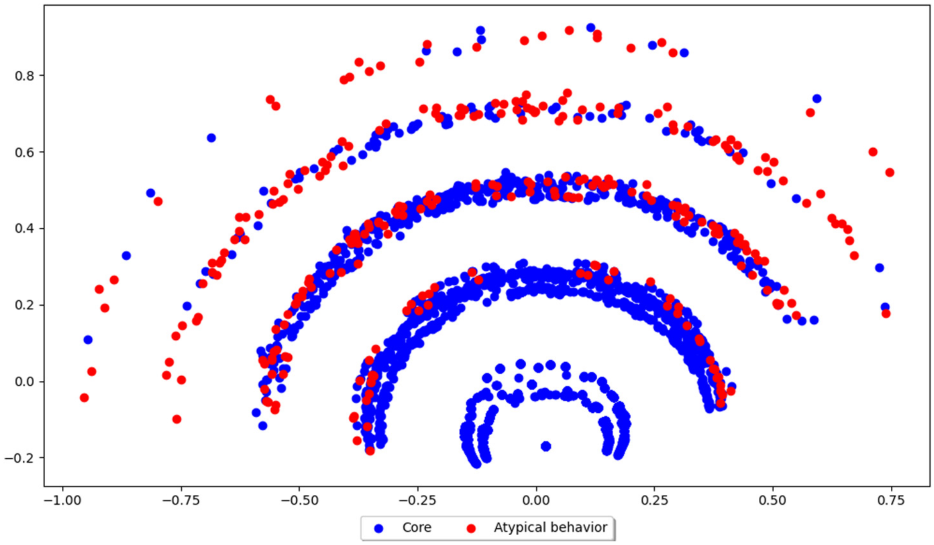
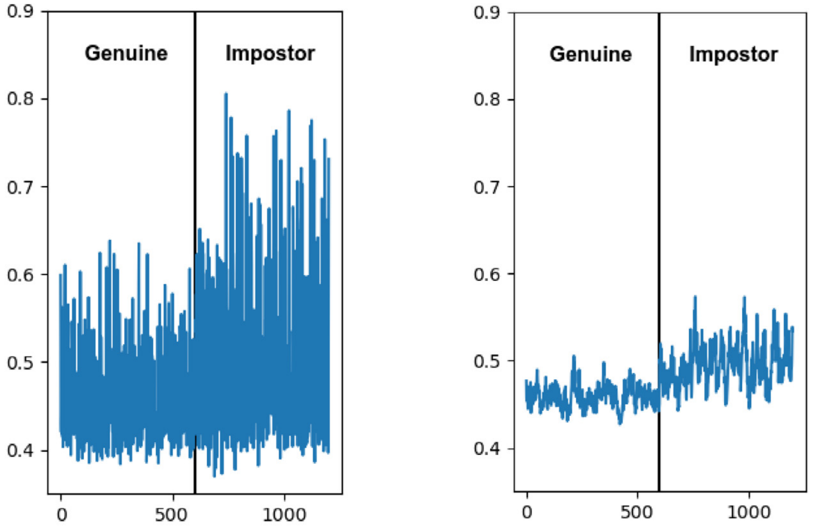


Fig. 3. Excerpt of cores extracted for a specific user using Multidimensional Scaling.



(a) Instance of raw risk values.

(b) Instance of risk values after EMA is applied.

Fig. 4. Instance of risk values for an user.

single observation does not decide whether a user is who claims to be, but the set of *WinSize* observations that does (i.e. the history of the user's behaviour). Thereby, in selecting the *WinSize* parameter, there is a trade-off between the performance of the method and its usefulness in a real environment. A high value of this *WinSize* smooths the risk curve better, considering more information, which translates into better accuracy in detecting impostors. On the other hand, a low value of *WinSize* makes it worse at detecting impostors, but it can make predictions considering less information and, therefore, be more useable in a real environment.

Once the risk curve is calculated, the last task of the risk model is to detect anomalous behaviour to classify the new samples into genuine or impostors. A decision barrier (i.e. threshold) is calculated by taking into account the values of the risk curve (see Fig. 5). Thus, the risk values lower than the threshold are classified into genuine samples, while those above are considered impostor samples.

The way to set an optimal threshold is by setting it to the value that produces the Equal Error Rate (EER) value, which is the point

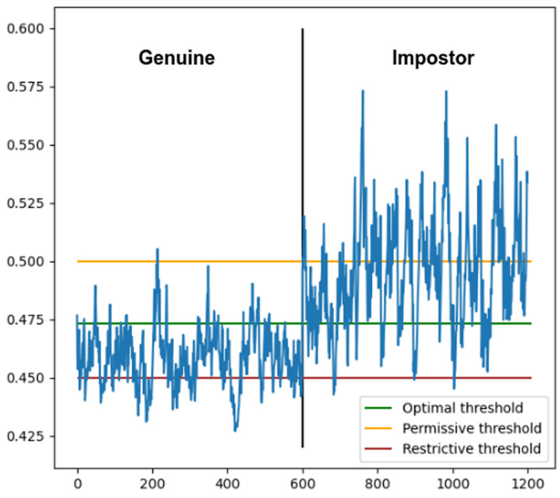


Fig. 5. Thresholds across risk values over time for a specific user using the EMA.

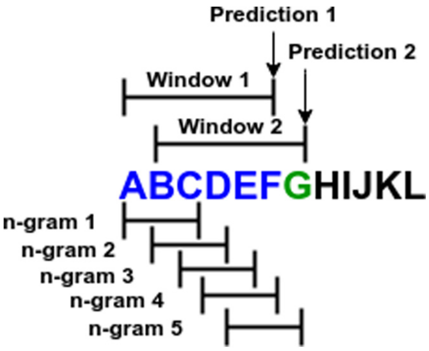


Fig. 6. Example of the predictions for $n - gramlength = 3$ and $WinSize = 4$.

defined by the intersection between the False Acceptance Rate (FAR) and the False Rejection Rate (FRR). The FAR refers to the probability that an unauthorised user is accepted. On the other hand, the FRR is defined as the probability that an authorised user is rejected when trying to access. Multiple FAR and FRR values can be obtained depending on the determined threshold. Thus, an extreme value of the threshold results in obtaining more restrictive outcomes (lower FAR but higher FRR), or more permissive outcomes (lower FRR but higher FAR) but never both lower simultaneously. Thus, the EER can be considered as the optimal point to set the threshold because it is the point where a minimum value is obtained between the pair FAR and FRR simultaneously.

Finally, it should be noted that when building the n-grams, the adjacent sequence is the same as the previous one except for the first and last characters. Thereby, the present method can make a prediction for each single user interaction (i.e. a single keystroke or a single mouse movement) once there is at least *WinSize* number of interactions (see Fig. 6).

3.6. Parameter selection

The method needs to define a total of two parameters and four hyper-parameters (see Table 2). The correct selection of these parameters is linked to the results to be obtained in the experiments. For this reason, it is necessary to make special attention to understanding and defining each of them correctly. To summarise, the parameters required are the *ngl* and the *WinSize*.

The value of the *ngl* defines the number of interactions (keyboard, mouse, or both) to be compared at each instant to obtain a prediction. An extreme selection of the value of the *ngl* can result in underfitting (for low values) or overfitting (for high values). This value has been limited to the values [5, 10, 20, 30] based on practical experience.

The *WinSize* defines how much historical information (in the form of sequences) is taken into account to make a prediction. The purpose of this parameter is to smooth the risk curve. The search for this parameter has been narrowed to the values [5, 10, 20, 50, 100] based on practical experience.

Regarding the RTE algorithm, two hyper-parameters have to be fixed: the *number of trees* and the *maximum depth*. These hyper-parameters decide the length of the alphabet in which each observation is categorised. Thus, a high value of both can lead to overfitting, as an extended alphabet will be generated. Consequently, the sequences generated will be very different from each other. On the other hand, a low value of both results in a very limited alphabet; for example, with one tree of depth two, the observations will be mapped only to two characters, which makes all the sequences generated very similar to each other and, therefore, non-discriminatory (underfitting). To map the observations to lower and upper cases, the search of these hyper-parameters is fixed to the range of [2, 5] based on practical experience.

For the DBSCAN algorithm, two hyper-parameters have to be also set: *EPS* and *MINS*. These hyper-parameters are set based on a grid search. The search for the parameters has been narrowed down to make the training more efficient. Thus, the hyper-parameter *EPS* will be in the range $[0 : f_{100}]$, where f_{100} represents the first 100 non-zero lowest distances calculated for the distance matrix. The *MINS* parameter is limited to the values [2, 10, 50, 100] based on practical experience.

Fixing these parameters correctly will depend largely on the particular requirements of the data on which the method is applied. However, the larger the *ngl* and *WinSize* parameters selected, the better results will be obtained in detecting anomalous behaviour. This is because more information will be considered in a row to make a prediction. Note that this also increases the time interval needed to make a prediction.

On the other hand, a grid search is recommended to set the hyper-parameters of the RTE and DBSCAN. First of all, for the RTE, the search values should be set to avoid overfitting and underfitting. Regarding the DBSCAN, the hyper-parameters must be set based on the final clusters obtained. For example, assumptions can be made to discard a fixed percentage of the data, or it can be deeply analysed the distributions of the distance matrix to narrow down the search values.

4. Experiments

This section details the experiments that have been carried out to validate the proposal. In this case, two experiments are considered. Firstly, the Keystroke and Mouse Dynamics from the UEBA Dataset [39] are used to analyse and validate the method. Secondly, the TWOS dataset [24] is used to evaluate the method and compare the results with other well-known proposals of the state-of-the-art.

Three use cases are performed for each experiment. The first two use cases consider each source of information independently (i.e. keystroke and mouse dynamics separately). The third use case evaluates the keystroke dynamics and the mouse dynamics simultaneously (i.e. the combination of both information sources). Note that the use of multiple information sources redounds in considering more interactions from users. This is an advantage over methods that only cover an information source independently. For instance, generally speaking, it could be better a 5% error method that can make predictions during 100% of user interactions than a 1% error method that can only make predictions during 10% of the user interactions.

The FAR, FRR, EER, Accuracy, Specificity, Negative Predictive Value (NPV), and F1^{-} score metrics are used to evaluate the results. These metrics are defined as follows:

- FAR: FP/(FP + TN)
- FRR: FN/(FN + TP)
- EER: Value of the intersection of FAR and FRR considering multiple classification thresholds.
- Accuracy: (TN+TP)/(TP+TN+FP+FN)
- Specificity: TN/(TN+FP)
- NPV: TN/(TN+FN)
- $F1^{-} = 2 \cdot \frac{\text{NPV}\cdot\text{Specificity}}{\text{NPV}+\text{Specificity}}$

where TP is the True Positives value (genuine users authenticated correctly), TN is the True Negatives value (impostor users correctly kicked out), FP is the False Positives value (impostor users that are incorrectly authenticated as genuine users), and FN are the False Negatives value (genuine users that are incorrectly considered impostors and they are kicked out). Specificity

Table 2
Summary of parameters and hyper-parameters of the proposed method. f_{100} represents the first 100 non-zero lowest distances calculated for the distance matrix.

Name	Description	Values
ngl	N-gram length	[5, 10, 20, 30]
WinSize	Historical information considered in the form of sequences	[5, 10, 20, 50, 100]
Number of trees	Number of trees to be generated in the RTE	[2, 5]
Maximum depth	Maximum depth of each tree in the RTE	[2, 5]
EPS	Maximum distance between two samples to be considered as neighbours in DBSCAN	$[0 : f_{100}]$
MINS	Minimum number of points that a neighbourhood must have to be considered as a cluster in DBSCAN	[2, 10, 50, 100]

Table 3
Obtained results for the keystroke dynamics use case for each combination of parameters for the UEBA Dataset.

ngl	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.457	0.457	0.457	0.442	0.397
5	10	0.423	0.423	0.423	0.439	0.392
5	20	0.356	0.356	0.366	0.420	0.287
5	50	0.268	0.239	0.282	0.323	0.207
5	100	**0.000**	**0.000**	0.048	0.250	0.062
10	5	0.384	0.381	0.386	0.303	0.453
10	10	0.343	0.333	0.343	0.189	0.414
10	20	0.276	0.276	0.286	0.122	0.250
10	50	0.059	0.059	0.074	0.015	0.167
10	100	**0.000**	**0.000**	0.047	**0.000**	0.186
20	5	0.304	0.301	0.305	0.297	0.141
20	10	0.226	0.228	0.226	0.230	0.030
20	20	0.145	0.145	0.147	0.163	0.019
20	50	0.050	0.051	0.050	0.043	0.027
20	100	**0.000**	**0.000**	0.011	0.003	0.026
30	5	0.267	0.267	0.270	0.236	0.254
30	10	0.177	0.175	0.178	0.142	0.097
30	20	0.091	0.090	0.091	0.038	0.038
30	50	0.006	**0.000**	**0.008**	**0.000**	**0.002**
30	100	**0.000**	**0.000**	0.012	**0.000**	0.009

is the proportion of impostors correctly identified, that is, the recall for the group of impostors. NPV is the effectiveness of the method when predicting impostors, which is the precision for the group of impostors. $F1^{-}$ score is a trade-off metric between NPV and Specificity [49]. Henceforth, the val suffix on these metrics, (e.g. FAR_val or FRR_val) indicates that the metric has been obtained over the validation set.

4.1. Keystroke and Mouse Dynamics for UEBA Dataset

The Keystroke and Mouse Dynamics for UEBA Dataset was collected through a web chat application. A total of 11 members of a research team with different ages, gender and usage profiles were analysed interacting with the application for five days. This interaction produced data that contained behavioural dynamics from the keyboard and mouse peripherals. It consists of 113,471 records for keystroke dynamics and 29,220 records for mouse dynamics.

First, the feature extraction process is accomplished. Subsequently, the train, test, and validation sets are generated for each user. The train sets are composed of only genuine samples and represent the 70% of the complete information. The test sets are composed of genuine and impostor samples. In the case of the genuine samples (i.e. 30% of the remaining data), a 60% is selected for testing (i.e. 18% of all genuine data). For the impostor samples, the same number is randomly selected from the other users (i.e. the other users act as impostors for a genuine user). Note that it is ensured that these random samples come from more than a single source of information. Finally, the validation sets are obtained from the remaining genuine data (i.e. 12% of all genuine data) and from the impostor data. As above, the same number of impostor samples are randomly selected.

The train set is used for each user to calculate the mean and standard deviation of each extracted feature to scale the other corresponding sets (i.e. test and validation sets). Then, the train set feeds the RTE training process, obtaining a sequence of characters. This sequence is separated into n-grams whose length is determined by the *ngl* parameter. These n-grams represent the retrieved behavioural dynamics, and they are used to build a distance matrix. The DNA sequence alignment algorithm is adopted to pairwise evaluate the distance between the n-grams. The obtained distance matrix feeds the DBSCAN algorithm. Thus, the result is a set of behavioural cores which represents the genuine behaviour of the user.

Once the training process concludes, each test set is evaluated through the RTE previously trained and separated into n-grams. Next, the distance between the obtained n-grams and the extracted behavioural cores is calculated. These distances are grouped according to the *WinSize* parameter. This allows obtaining the risk values of a sequence of n-grams. Lastly, the risk values are analysed to set an optimal threshold (see Section 3.5).

Each validation set is used to test the method against samples that have never been considered before. This step allows analysing the robustness, the generalisation capability, and the performance of the method. This set is evaluated using the previously obtained threshold.

Once the train, test and validation sets are built, the method is evaluated through the three use cases. The results of these use cases are shown in Tables 3, 4, and 5. All the possible combinations of the parameters are illustrated. The displayed results correspond to the mean of all users. The best results are highlighted in bold for each metric considered.

In all the use cases, the values of EER, FAR, and FRR decrease (i.e. improves the performance) when the *ngl* or *WinSize* parameters increase. Notice that the method uses more information for the predictions as the value of the parameters increases. Thus, the maximum values of EER for each use case (0.457, 0.438 and 0.397 respectively) are obtained for $ngl = 5$ and $WinSize = 5$. On the other hand, nearly perfect predictions are obtained for $ngl = 30$ and $WinSize = 100$. In this way, if the *WinSize* is set to 100, good results are obtained for any *ngl*. However, these results become more robust for the validation sets as *ngl* increases.

The best performance for the test and validation sets simultaneously is obtained for combining both information sources. This corroborates that the combination of information enhances the performance of the method in the continuous authentication task.

4.2. TWOS dataset evaluation

In this section, the TWOS dataset is used to evaluate the method. This dataset was collected during the competition organised by the Singapore University of Technology and Design.

Table 4
Obtained results for the mouse dynamics use case for each combination of parameters for the UEBA Dataset.

ngl	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.438	0.438	0.438	0.308	0.390
5	10	0.392	0.392	0.396	0.234	0.359
5	20	0.347	0.343	0.351	0.146	0.280
5	50	0.265	0.265	0.265	0.031	0.076
5	100	0.034	0.024	0.065	0.506	0.513
10	5	0.354	0.354	0.358	0.329	0.306
10	10	0.306	0.302	0.306	0.261	0.176
10	20	0.202	0.198	0.202	0.129	0.129
10	50	0.090	0.090	0.094	8.000	**0.000**
10	100	**0.000**	**0.000**	0.031	0.500	0.500
20	5	0.147	0.139	0.151	0.133	0.223
20	10	0.077	0.073	0.081	0.112	0.050
20	20	0.034	0.030	0.034	0.053	**0.000**
20	50	0.000	**0.000**	0.016	**0.000**	**0.000**
20	100	**0.000**	**0.000**	0.026	**0.000**	**0.000**
30	5	0.090	0.090	0.090	0.080	0.049
30	10	0.025	0.025	0.025	0.013	0.006
30	20	0.004	0.000	0.012	**0.000**	0.011
30	50	**0.000**	**0.000**	**0.007**	**0.000**	0.041
30	100	**0.000**	**0.000**	0.011	**0.000**	0.092

Table 5
Obtained results for the combination of information sources use case for each combination of parameters for the UEBA Dataset.

ngl	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.397	0.397	0.397	0.342	0.486
5	10	0.364	0.364	0.366	0.294	0.501
5	20	0.320	0.317	0.320	0.273	0.484
5	50	0.229	0.223	0.232	0.131	0.489
5	100	0.112	0.115	0.112	0.049	0.448
10	5	0.289	0.284	0.292	0.322	0.329
10	10	0.219	0.222	0.219	0.284	0.296
10	20	0.141	0.138	0.144	0.154	0.263
10	50	0.029	0.024	0.032	0.019	0.100
10	100	**0.000**	**0.000**	0.007	**0.000**	0.019
20	5	0.167	0.167	0.169	0.117	0.146
20	10	0.080	0.075	0.085	0.052	0.070
20	20	0.006	0.006	0.010	0.016	0.015
20	50	**0.000**	**0.000**	0.006	**0.000**	0.008
20	100	**0.000**	**0.000**	0.008	**0.000**	0.011
30	5	0.128	0.128	0.131	0.217	0.116
30	10	0.054	0.068	0.054	0.165	0.030
30	20	0.020	0.014	0.023	0.098	0.001
30	50	**0.000**	**0.000**	0.006	**0.000**	0.008
30	100	**0.000**	**0.000**	**0.004**	**0.000**	**0.000**

Table 6
Obtained results for the keystroke dynamics use case for each combination of parameters for the TWOS Dataset.

ngl	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.414	0.413	0.415	0.407	0.414
5	10	0.386	0.386	0.386	0.375	0.384
5	20	0.341	0.343	0.341	0.346	0.353
5	50	0.269	0.269	0.269	0.268	0.233
5	100	0.214	0.219	0.214	0.189	0.131
10	5	0.362	0.362	0.362	0.380	0.382
10	10	0.337	0.335	0.339	0.340	0.320
10	20	0.305	0.304	0.307	0.282	0.237
10	50	0.217	0.217	0.217	0.180	0.126
10	100	0.133	0.133	0.134	0.044	0.052
20	5	0.317	0.318	0.317	0.346	0.302
20	10	0.259	0.257	0.259	0.305	0.245
20	20	0.189	0.189	0.191	0.227	0.172
20	50	0.080	0.078	0.082	0.110	0.072
20	100	0.008	0.008	0.008	0.018	0.019
30	5	0.299	0.299	0.299	0.293	0.288
30	10	0.231	0.231	0.232	0.222	0.224
30	20	0.156	0.156	0.156	0.139	0.139
30	50	0.055	0.054	0.055	0.060	0.052
30	100	**0.007**	**0.007**	**0.007**	**0.000**	**0.015**

Table 7
Obtained results for the mouse dynamics use case for each combination of parameters for the TWOS Dataset.

ngl	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.433	0.433	0.434	0.400	0.432
5	10	0.404	0.403	0.404	0.375	0.400
5	20	0.371	0.370	0.372	0.313	0.351
5	50	0.317	0.316	0.318	0.252	0.286
5	100	0.266	0.265	0.266	0.205	0.260
10	5	0.420	0.419	0.420	0.406	0.394
10	10	0.392	0.392	0.393	0.369	0.356
10	20	0.355	0.354	0.355	0.290	0.315
10	50	0.285	0.285	0.285	0.263	0.209
10	100	0.180	0.176	0.180	0.250	0.126
20	5	0.377	0.376	0.379	0.362	0.388
20	10	0.336	0.335	0.337	0.311	0.347
20	20	0.286	0.285	0.288	0.255	0.287
20	50	0.174	0.173	0.174	0.153	0.211
20	100	0.088	0.086	0.101	0.086	0.117
30	5	0.359	0.357	0.361	0.338	0.352
30	10	0.313	0.312	0.314	0.308	0.302
30	20	0.250	0.250	0.251	0.237	0.243
30	50	0.136	0.136	0.137	0.141	0.146
30	100	**0.051**	**0.049**	**0.089**	**0.047**	**0.073**

The data comes from six information sources: keystrokes, mouse, host monitor, network traffic, SMTP logs, and logon. These data are completed with additional information from a psychological personality questionnaire. All the behavioural dynamics were collected in a free environment.

Regarding the information sources, the behavioural dynamics gathered from the keystrokes and mouse have been considered analogously to the first experiment. The 24 users are considered for five days.

The train, test and validation sets are obtained following the same steps achieved in the previous experiment. In conclusion, the behavioural cores are obtained, and a risk model for each user is produced. Once these tasks are accomplished, the three use cases are evaluated.

Tables 6–8 show the results obtained for each use case respectively for the test and validation sets. It is considered all the possible combinations of parameters shown in Section 3.6. The displayed results correspond to the mean of all users. The best results are highlighted in bold for each metric considered.

The results for the independent sources (i.e. keystroke and mouse dynamics) and their combination are promising and robust. The method's performance increases when more information is considered (i.e. *ngl* or *WinSize* increase). Regarding the keystroke dynamics use case, the EER values go from 0.41 to 0.007, obtaining FAR and FRR values in the validation set of 0.400 and 0.432 respectively for the worst combination of parameters, and 0 and 0.015 for the best combination of parameters. On the other hand, the results of the mouse dynamics use case are slightly worse. The EER values range between 0.433 and 0.051, obtaining FAR and FRR values in the validation set of 0.400 and 0.432 for the worst case, and 0.047 and 0.073 for the best case. The best results are obtained for the combination of both information sources. EER values are in the range between 0.407 and 0.006 achieving FAR and FRR values in the validation set of 0.409 and 0.410 respectively for the worst parameter selection, and 0.014 and 0.004 for the best case.

In the particular case of the keyboard dynamics, each character of the sequence (i.e. di-graph) has an average execution time of 0.227 s. In the case of the mouse dynamics, each character

Table 8
Obtained results for the combination of information sources use case for each combination of parameters for the TWOS Dataset.

ngl	WinSize	EER	FAR	FRR	FAR_val	FRR_val
5	5	0.407	0.407	0.407	0.409	0.410
5	10	0.380	0.380	0.380	0.381	0.375
5	20	0.335	0.335	0.335	0.343	0.321
5	50	0.257	0.257	0.257	0.270	0.228
5	100	0.177	0.177	0.176	0.197	0.150
10	5	0.359	0.359	0.359	0.371	0.349
10	10	0.320	0.319	0.319	0.335	0.307
10	20	0.263	0.262	0.263	0.289	0.245
10	50	0.169	0.168	0.169	0.213	0.159
10	100	0.082	0.082	0.082	0.140	0.099
20	5	0.295	0.295	0.296	0.310	0.306
20	10	0.226	0.226	0.226	0.267	0.230
20	20	0.156	0.156	0.157	0.206	0.157
20	50	0.066	0.065	0.067	0.098	0.080
20	100	0.022	0.022	0.022	0.041	0.018
30	5	0.268	0.268	0.268	0.282	0.285
30	10	0.201	0.200	0.201	0.227	0.211
30	20	0.121	0.122	0.121	0.168	0.110
30	50	0.038	0.037	0.038	0.076	0.020
30	100	**0.006**	**0.006**	**0.006**	**0.014**	**0.004**

represents a stroke of 5 s. This means that for *ngl* of size 10, approximately 2.27 and 50 s of information are used to make a keyboard and mouse prediction, respectively. Thus, 22.7 and 500 s of historical information are used to predict a *WinSize* value of 100. In the case of the combination of the information sources, the proportion of keystroke dynamics is 73%, while mouse dynamics represent the 27% of the data on average for all users. With *ngl* equals to 10, a mean vector would have 7 characters from the keyboard and 3 characters from the mouse. This vector represents an average time of 16.589 s. A *WinSize* value of 100 would cover on average 151.571 s of historical information. Although all this information is considered, predictions are made every 0.227 and 5 s on average. This is because the evaluated information consists of n-grams in which the adjacent sequence is equal to the subsequent one except for the first and last characters (see Fig. 6).

Finally, SVM and RF have been selected as representative algorithms of the state-of-the-art to compare the effectiveness of the proposed method. Thus, to train these algorithms, a preprocessing of the raw data has been performed to use the parameters *ngl* and *WinSize*, as the proposed method does. Firstly, the raw data is clustered into subgroups of size *ngl*. Then, the models are trained by performing a grid parameter search. The obtained predictions are clustered into time windows using the *WinSize* parameter and applying EMA. To obtain a model that combines information to compare with the combination of the proposed method, the best model was selected independently for each information source (i.e. keystroke and mouse dynamics) and its predictions were combined at the decision level. For this purpose, the predictive probability values have been normalised in the same range so that they can fill the same risk buffer. Furthermore, the proposed method is also compared with two well-known approaches that use the TWOS dataset [8,27].

The results are shown in Table 9. Analogously to previous tables, the displayed results for the proposed method correspond to the mean of all users. Thus, the results for the method are the ones obtained for a pair of parameters that achieved better results than the approaches for each use case in the validation set. That is, the pairs (30, 100), (20, 100), (5, 100), and (20, 50) are selected respectively for the *ngl* and *WinSize* parameters. These sets of parameters are also used to train the SVM and the RF. Note that the proposed method improves the results of state-of-the-art proposals when enough information is considered (i.e. large *ngl* and *WinSize* values). Note that as mentioned in Section 3.5, a trade-off has to be made between the information to be considered and the usability in a specific environment.

Comparing the results in [27] with the ones in Table 7, $ngl=30$ and $WinSize=50$ values get similar results for the mouse use case. Moreover, considering more information, such as a selection of the pairs (20, 100) or (30, 100) considerably improves the results. Analysing the [8] approach, the results are comparable to a selection of the pairs, (5, 100), (10, 50), (20, 20), and (30, 20) (see Table 8). If context information (i.e. another source of information) is taken into account, the results can be evened out by setting the parameters to (20, 50) or improved by the presented method using only keystroke dynamics and mouse dynamics by setting the parameters to (20, 100), (30, 50) and (30, 100). On the other hand, the algorithms of SVM and RF obtain satisfactory results. However, these results are always improved by the proposed method in terms of EER on equal conditions.

Regarding the efficiency of the proposed method, it can make predictions for each user interaction. As mentioned above, that is 0.27 and 5 s on average for keystroke and mouse dynamics respectively. Making predictions in such short time intervals can lead to significant computational overhead that decreases the efficiency of the method. However, the method can be trivially adapted to perform predictions at more spaced time intervals by decreasing the windows to be processed. Note that the latter could also lead to a loss in the accuracy of detecting impostors because less information will be considered. Other proposals, such as [8] perform predictions for each session, that is, from the time the user clicks the log-in button until he clicks the log-out button. In the [27] approach, they use different sequence lengths. For example, a fixed time interval of 10 s or the traversed distance to determine the size of the sequences, so that this distance represents at least a percentage of the display resolution. Finally, they also considered 2D-windows of a certain size. In summary, they use both fixed and variable sequence sizes. Taking all this into account, it could be said that the proposed method is within the efficiency standards of the state-of-the-art proposals.

It is also worth mentioning that the results obtained for the TWOS dataset are, in general, worse than those obtained for the UEBA dataset. This is because the UEBA dataset is much smaller and therefore, for this particular case, the data distributions are a priory more differentiating, thus obtaining better results. Note that this does not mean that as the volume of data increases, the method performs worse, but that when testing, taking a larger number of impostor users to evaluate a genuine user, it is more likely that a particular impostor is more similar to the genuine sample, lowering the results in consequence.

5. Conclusions

A method to combine information from multiple sources devoted to enhancing continuous authentication solutions has been presented. It includes a novel technique to represent temporal information based on SAX implementing through RTEs. This enables the production of a sequence of symbols that discretise the information. DNA sequence alignment techniques have been used to compare them accurately. Then, behavioural cores are extracted using a density-based clustering model to discard outliers samples. Finally, a risk model has been specified to evaluate new behavioural dynamics.

The proposed method has empirically shown that the use of the RTE technique provides a very accurate representation despite the loss of information during the discretisation process. It has been demonstrated that combining information at the feature level enhances the results of previous approaches and provides new insights for further research. However, the implementation

Table 9
Results of the several approaches used to compare the proposal. The results displayed for [27] are the ones obtained for the 2D-CNN, while for the combination in [8] are the ones obtained for the SVM classifier. *C* refers to context information retrieved from an external dataset. *KD* and *MD* represent keystroke dynamics and mouse dynamics respectively. RF+RF represent a combination model at the decision level using for both information sources the RF algorithm. Acc is Accuracy and Spec is Specificity.

Work	Source	EER	FAR_val	FRR_val	F1^{-}_val	Acc_val	NPV_val	Spec_val
SVM	KD	0.136	0.158	0.151	0.850	0.845	0.858	0.842
RF	KD	0.084	0.093	0.174	0.860	0.865	0.819	0.907
Our	KD	**0.007**	**0.000**	**0.015**	**0.968**	**0.979**	**0.945**	**0.993**
SVM	MD	0.171	0.180	**0.063**	0.882	0.877	**0.955**	0.820
RF	MD	0.109	0.141	0.081	0.890	0.888	0.925	0.859
[27]	MD	0.130	0.136	0.149	–	–	–	–
Our	MD	**0.088**	**0.086**	0.117	**0.900**	**0.909**	0.888	**0.914**
RF+RF	KD+MD	0.180	0.228	0.169	0.814	0.800	0.861	0.772
[8]	KD+MD	–	–	–	0.806	0.751	**0.932**	0.710
Our	KD+MD	**0.177**	**0.197**	**0.150**	**0.828**	**0.826**	0.856	**0.803**
RF+RF	KD+MD	0.121	0.126	0.152	0.860	0.860	0.848	0.874
[8]	KD+MD+C	–	–	–	**0.914**	**0.915**	0.874	**0.912**
Our	KD+MD	**0.066**	**0.098**	**0.080**	0.912	**0.915**	**0.921**	0.902

of this method may introduce some overhead in the training of the proposed models and in the predicting tasks, which may result in added latencies in the operation of the system in which the solution is integrated. Nevertheless, this overhead translates into a considerable improvement in the security levels, so it is likely that many services or applications will choose to implement it.

Future research will focus on testing more information sources and improving the scalability and performance of the proposed methods. In this sense, IoT devices, smartphone sensors and wearables are interesting to consider. Regarding the performance, a cost-sensitive measure for weighting the penalty of mismatch of the characters could be considered in the global alignment technique. A variation of the Breiman proximity measure [50] could be used, for example, to provide different levels of similarity.

Furthermore, categorising behavioural information may simplify the complexity of the problem without affecting performance. Fuzzy logic techniques and NNs to produce embeddings are noteworthy options to test in the future.

Finally, it should be noted that the techniques presented in the proposal are very specific to the application domain of continuous authentication. However, special mention should be made of SAX. This technique based on RTE has proven to discretise different longitudinal data satisfactorily. This fact has led to including it in many other domains such as time series. In this sense, detection of temporal anomalies in the stock market, or in the field of health to monitor changes in heart rate are well-known examples of its usability.

CRediT authorship contribution statement

Alejandro G. Martín: Software, Validation, Methodology, Formal analysis, Investigation, Data curation, Writing – original draft. **Isaac Martín de Diego:** Conceptualization, Writing - review & editing, Funding acquisition. **Alberto Fernández-Isabel:** Methodology, Validation, Writing - review. **Marta Beltrán:** Conceptualization, Validation, Investigation, Writing – review & editing, Supervision, Funding acquisition. **Rubén R. Fernández:** Software, Validation, Methodology, Writing - review.

Declaration of competing interest

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

Acknowledgements

Research supported by grants from Madrid Autonomous Community, Spain (ref: IND2019/TIC-17169); and from the Spanish Ministry of Economy and Competitiveness, Spain under the Retos-Investigación program: MODAS-IN (ref: RTI-2018-094269-B-I00); and donation of the Titan V GPU by NVIDIA Corporation, Spain; and by grants from Rey Juan Carlos University, Spain (Ref: C1PREDOC2020)

References

[1] K. Aravindhan, R. Karthiga, One time password: A survey, Int. J. Emerg. Trends Eng. Dev. 1 (3) (2013) 613–623.

[2] S. Eberz, K.B. Rasmussen, V. Lenders, I. Martinovic, Evaluating behavioral biometrics for continuous authentication: Challenges and metrics, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017, pp. 386–399.

[3] A.G. Martín, A. Fernández-Isabel, I.M. de Diego, M. Beltrán, A survey for user behavior analysis based on machine learning techniques: current models and applications, Appl. Intell. (2021) 1–27.

[4] K.S. Killourhy, R.A. Maxion, Comparing anomaly-detection algorithms for keystroke dynamics, in: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, IEEE, 2009, pp. 125–134.

[5] Forcepoint, 2021, https://www.forcepoint.com/product/ueba-user-~entity-behavior-analytics, accessed: 2021-12-10.

[6] Fortinet, 2021, https://www.fortinet.com/products/ueba, accessed: 2021-12-10.

[7] A.G. Martín, M. Beltrán, A. Fernández-Isabel, I.M. de Diego, An approach to detect user behaviour anomalies within identity federations, Comput. Secur. (2021) 102356.

[8] J. Solano, L. Camacho, A. Correa, C. Deiro, J. Vargas, M. Ochoa, Combining behavioral biometrics and session context analytics to enhance risk-based static authentication in web applications, International Journal of Information Security 20 (2) (2021) 181–197.

[9] K.O. Bailey, J.S. Okolica, G.L. Peterson, User identification and authentication using multi-modal behavioral biometrics, Comput. Secur. 43 (2014) 77–89.

[10] S.G. Lyastani, M. Schilling, M. Neumayr, M. Backes, S. Bugiel, Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication, in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 268–285.

[11] OIDF, Ope?id connect 1.0, 2022, http://openid.net/connect/, accessed: 2022-01-18.

[12] R.S. Gaines, W. Lisowski, S.J. Press, N. Shapiro, Authentication by Keystroke Timing: Some Preliminary Results, Tech. rep., Rand Corp Santa Monica CA, 1980.

[13] S. Bleha, C. Slivinsky, B. Hussien, Computer-access security systems using keystroke dynamics, IEEE Trans. Pattern Anal. Mach. Intell. 12 (12) (1990) 1217–1222.

[14] S. Cho, C. Han, D.H. Han, H.-I. Kim, Web-based keystroke dynamics identity verification using neural network, J. Organ. Comput. Electron. Commer. 10 (4) (2000) 295–307.

[15] F. Monrose, A. Rubin, Authentication via keystroke dynamics, in: Proceedings of the 4th ACM Conference on Computer and Communications Security, 1997, pp. 48–56.

[16] A. Alsultan, K. Warwick, H. Wei, Non-conventional keystroke dynamics for user authentication, Pattern Recognit. Lett. 89 (2017) 53–59.

[17] J. Kim, H. Kim, P. Kang, Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection, Appl. Soft Comput. 62 (2018) 1077–1087.

[18] K.S. Balagani, V.V. Phoha, A. Ray, S. Phoha, On the discriminability of keystroke feature vectors used in fixed text keystroke authentication, Pattern Recognit. Lett. 32 (7) (2011) 1070–1080.

[19] O. Alpar, Frequency spectrograms for biometric keystroke authentication using neural network based classifier, Knowl.-Based Syst. 116 (2017) 163–171.

[20] L. Xiaofeng, Z. Shengfei, Y. Shengwei, Continuous authentication by free-text keystroke based on CNN plus RNN, Procedia Comput. Sci. 147 (2019) 314–318.

[21] Y. Sun, H. Ceker, S. Upadhyaya, Shared keystroke dataset for continuous authentication, in: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2016, pp. 1–6.

[22] J. Huang, D. Hou, S. Schuckers, T. Law, A. Sherwin, Benchmarking keystroke authentication algorithms, in: 2017 IEEE Workshop on Information Forensics and Security (WIFS), IEEE, 2017, pp. 1–6.

[23] B. Ayotte, M. Banavar, D. Hou, S. Schuckers, Fast free-text authentication via instance-based keystroke dynamics, IEEE Trans. Biometrics Behav. Identity Sci. 2 (4) (2020) 377–387.

[24] A. Harilal, F. Toffalini, I. Homoliak, J.H. Castellanos, J. Guarnizo, S. Mondal, M. Ochoa, The Wolf Of SUTD (TWOS): A dataset of malicious insider threat behavior based on a gamified competition, J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 9 (1) (2018) 54–85.

[25] R.A. Everitt, P.W. McOwan, Java-based internet biometric authentication system, IEEE Trans. Pattern Anal. Mach. Intell. 25 (9) (2003) 1166–1172.

[26] A.A.E. Ahmed, I. Traore, A new biometric technology based on mouse dynamics, IEEE Trans. Dependable Secure Comput. 4 (3) (2007) 165–179.

[27] P. Chong, Y. Elovici, A. Binder, User authentication based on mouse dynamics using deep neural networks: A comprehensive study, IEEE Trans. Inf. Forensics Secur. 15 (2019) 1086–1101.

[28] C. Shen, Z. Cai, X. Guan, Y. Du, R.A. Maxion, User authentication through mouse dynamics, IEEE Trans. Inf. Forensics Secur. 8 (1) (2012) 16–30.

[29] D. Qin, S. Fu, G. Amariucai, D. Qiao, Y. Guan, MAUSPAD: Mouse-based authentication using segmentation-based, progress-adjusted DTW, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 425–433.

[30] T. Hu, W. Niu, X. Zhang, X. Liu, J. Lu, Y. Liu, An insider threat detection approach based on mouse dynamics and deep learning, Secur. Commun. Netw. 2019 (2019).

[31] H. Gamboa, A. Fred, A behavioral biometric system based on human-computer interaction, in: Biometric Technology for Human Identification, Vol. 5404, International Society for Optics and Photonics, 2004, pp. 381–392.

[32] A. Ross, A. Jain, Information fusion in biometrics, Pattern Recognit. Lett. 24 (13) (2003) 2115–2125.

[33] S. Mondal, P. Bours, A study on continuous authentication using a combination of keystroke and mouse biometrics, Neurocomputing 230 (2017) 1–22.

[34] L. Fridman, A. Stolerman, S. Acharya, P. Brennan, P. Juola, R. Greenstadt, M. Kam, Multi-modal decision fusion for continuous authentication, Comput. Electr. Eng. 41 (2015) 142–156.

[35] I. Traore, I. Woungang, M.S. Obaidat, Y. Nakkabi, I. Lai, Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments, in: 2012 Fourth International Conference on Digital Home, IEEE, 2012, pp. 138–145.

[36] S. Salmeron-Majadas, R.S. Baker, O.C. Santos, J.G. Boticario, A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios, IEEE Access 6 (2018) 39154–39179.

[37] X. Wang, Q. Zheng, K. Zheng, T. Wu, User authentication method based on MKL for keystroke and mouse behavioral feature fusion, Secur. Commun. Netw. 2020 (2020).

[38] Y. Li, B. Zou, S. Deng, G. Zhou, Using feature fusion strategies in continuous authentication on smartphones, IEEE Internet Comput. 24 (2) (2020) 49–56.

[39] “Keystroke and mouse dynamics for UEBA dataset”, mendeley data, v2, 2021, http://dx.doi.org/10.17632/f78jsh6zp9.2, accessed: 2021-12-10.

[40] M.G. Baydogan, G. Runger, Learning a symbolic representation for multivariate time series classification, Data Min. Knowl. Discov. 29 (2) (2015) 400–422.

[41] W. Cohen, P. Ravikumar, S. Fienberg, A comparison of string metrics for matching names and records, in: Kdd Workshop on Data Cleaning and Object Consolidation, Vol. 3, 2003, pp. 73–78.

[42] H. Li, N. Homer, A survey of sequence alignment algorithms for next-generation sequencing, Briefings Bioinform. 11 (5) (2010) 473–483.

[43] K. Khan, S.U. Rehman, K. Aziz, S. Fong, S. Sarasvady, DBSCAN: Past, present and future, in: The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), IEEE, 2014, pp. 232–238.

[44] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.

[45] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, in: Twentieth Annual Conference on Neural Information Processing Systems (NIPS'06), MIT Press, 2006, pp. 985–992.

[46] H. Abdi, Metric multidimensional scaling (MDS): analyzing distance matrices, Encyclopedia Meas. Statist. (2007) 1–13.

[47] P.H. Pisani, R. Giot, A.C. De Carvalho, A.C. Lorena, Enhanced template update: Application to keystroke dynamics, Comput. Secur. 60 (2016) 134–153.

[48] F. Klinker, Exponential moving average versus moving exponential average, Math. Semesterber. 58 (1) (2011) 97–107.

[49] I.M. De Diego, A.R. Redondo, R.R. Fernández, J. Navarro, J.M. Moguerza, General performance score for classification problems, Appl. Intell. (2022).

[50] L. Breiman, Manual on setting up, using, and understanding random forests v3. 1. 2002, 2002, p. 1, URL: http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.