Engineering Applications of Artificial Intelligence 99 (2021) 104135

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Adaptation of the idea of concept drift to some behavioral biometrics: Preliminary studies

Piotr Porwik *, Rafal Doroz

University of Silesia, Katowice, Faculty of Science and Technology, 41-200 Sosnowiec, Bedzinska str. 39, Poland

ARTICLE INFO

Keywords:
Concept drift
Biometrics
Classifiers
Ensemble of classifiers

ABSTRACT

In this paper we present a novel strategy that utilizes concept drift to improve some biometric procedures. The proposed method can be applied whenever behavioral signals change and those changes need to be detected. From a security point of view, this is important because detection of and appropriate response to change should result in some alteration in the operation of the biometric system. As one example, this allows for the detection of legitimate and illegitimate users.

Experiments performed on real biometric signals have demonstrated that the proposed techniques could be introduced into existing professional biometric systems based on behavioral features.

1. Introduction

The effective detection of changes in stream-based data is an important task from both the theoretical and practical points of view. Conventional classification methods often assume that input signals are static and are emitted from the same distribution, so that classifiers can be trained on historical data. Variation is assumed to occur only within a class of instances. Indeed, in many biometric tasks these assumptions are true. In retinal and fingerprint analyses, individual biometric features of a given individual are stable and can be employed in biometric recognition. In other cases and especially in behavioral biometrics, the features gathered can change over time. This can be observed in the analysis of keystroke-based biometrics where, for example, personal dynamic features change over various time intervals. Such phenomena are observed also during facial recognition procedures. Faces constantly change due to different haircuts, beards or moustaches being shaved, glasses, makeup. etc. Another typical example is weather prediction. The scenarios mentioned above indicate that previous data may be unsuitable and of insufficient quantity for recognition of a given object at the present time. Such changes should be monitored and classification procedures adapted to these changes. Problems with the detection of data changes over time can be overcome by the application of concept drift (Gonçalves et al., 2014).

Generally, concept drift assumes that some anomalies in the data stream will be observed. As the data stream changes, additional actions need to be undertaken to maintain classification quality. In these cases, either new classifiers should be included in the classification process or old classifiers re-programmed. In biometrics, concept drift describes changes in a user's behavior over time. The anomaly detection procedure applied must be capable of adapting to such changes if the user is legitimate, recognize hostile actions, but it must not adapt to illegitimate users. This means that in biometrics, data stream changes should be detected while simultaneously recognizing the reason for the observed changes. This additionally complicates a problem that does not occur in classical concept drift theory. In classical concept theory, only the phenomenon should be detected and the classifiers' appropriate reaction predicted.

Today, much concept drift research has been presented with both its theoretical and practical ideas being well known (Gonçalves et al., 2014; de Barros and de Carvalho Santos, 2019; Lu et al., 2014; Brzezinski and Stefanowski, 2014). The interested reader is encouraged to review this work. Here we briefly discuss concept drift with specific reference to some biometric tasks.

2. Concept drift vs behavioral biometrics

Concept drift detection strategies employ base classifiers to classify incoming instances as they endeavor to predict the true class of each given instance. Taking into account previous classification results, drift detection algorithms attempt to recognize whether data drift has occurred (Gonçalves et al., 2014; Pesaranghader et al., 2018). This allows classification error to be minimized during class prediction. We assume that all data streams are timestamped.

In some cases, human behavior can be analyzed as a data stream changing over time. Human behavior is frequently modified by disease, temporary contusions, fever, fatigue and other passing symptoms. From a biometric domain point of view, data fluctuations while not desirable are inevitable. Additionally, such phenomena can exhibit either

* Corresponding author.
E-mail address: piotr.porwik@us.edu.pl (P. Porwik).

https://doi.org/10.1016/j.engappai.2020.104135
Received 19 June 2020; Received in revised form 24 October 2020; Accepted 10 December 2020
Available online 16 December 2020
0952-1976/© 2020 Elsevier Ltd. All rights reserved.

a gently or rapidly changing character over time. While changes may originate from the person being studied, they may in addition be caused by an intruder. Legitimate users and intruders are detected on the basis of their biometric data, and these users should be treated separately.

There are two basic drift categories: "virtual drift" (drift in data distribution with decision boundaries remaining unaltered) and "real drift" (drift in the decision's concepts, with decision boundaries altered) (Lu et al., 2014; Hoens et al., 2012). In practice, both types occur. Patterns of drift are generally categorized into "sudden", "incremental", "gradual", and "reoccurring" (Gonçalves et al., 2014; Pesaranghader et al., 2018; Bonab and Can, 2018), but only some drift types should be recognized in the biometric domain. Demonstrations of these patterns are presented in Fig. 1.

In our previous investigation in which analysis of keystroke dynamics was presented, we assumed that the collected data were stable (Wesołowski et al., 2016) in time. In this study, we provide a solution for the recognition of legitimate (illegitimate) individuals based on the presence and characteristics of the observed data drift, an approach more realistic for the real world. We clearly demonstrate that our proposed modification raises biometric systems' classification quality. The concept drift scenarios presented in Fig. 1 can be related to biometrics:

- Drift appears incrementally (Fig. 1b) when many moderately slow concept developments occur between the first and last concept. In biometrics, this represents an individual's slowly (mildly) changing behavioral characteristic, but they are still a legitimate user. For such a person, classifiers should be learned again, taking into account the user's new behavioral data.
- Drift may occurs suddenly (abruptly) by replacing one concept with another as presented in Fig. 1c. In biometrics, this indicates that instead of a legitimate user, a counterfeiter has presented biometric characteristics (Banerjee and Woodard, 2012). A profiled user only very occasionally instantaneously displays biometric qualities of low similarity with their own past behavior. An appropriate alert should be generated and the system should halt without any classifiers being rebuilt. In contrast to the original concept drift strategy, the model of a classifier trained on a data set incorporating drift can maintain classification quality. In biometrics, sudden drift occurrence should always be alarmed.

In biometrics, if changes are short in time, these changes can be recognized as behavioral fluctuations (similar as in incremental drift). Otherwise, such drift is recognized as an intruder's data and the system should stop after a warning alert.

3. A statistical method for biometric drift data detection

As previously stated, unsupervised drift detection is very convenient from a practical point of view, because only the data's distribution needs to be checked. For behavioral-based biometrics, data flows as a stream. Consecutive data must be checked as to whether each data chunk (data window) comes from the same distribution as the reference data. In biometrics, reference data are the behavioral characteristics of a given user. In our approach, we will use the non-parametric univariate two-sample Kolmogorov–Smirnov statistical test for biometric drift detection (Massey, 1951; Sharipov, 2011; Andress, 2014). Some modification of this test will be used for outlier detection which will assist in the detection of incremental drift. Additionally, this test has the advantage of making no assumptions about the data distribution, so it can be employed on any data.

3.1. Detection of drift in the data distribution

Let $\{x_1, \dots, x_n\}$, be independent, identically distributed (i.i.d) ordered real random observations. For such assumption we can define function $F_n(t)$ as empirical cumulative distribution function (ECDF) of samples:

$$F_n(t) = \frac{number\ of\ elements\ in\ sample \le t}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(x_i \le t). \tag{1}$$

The Kolmogorov–Smirnov test may be applied to test whether two one-dimensional distributions of samples belongs or not to the same distribution. The Kolmogorov–Smirnov test statistic can be expressed as:

$$D_{n,m} = \sup_t |F_n(t) - F_m(t)|, \tag{2}$$

where F_n and F_m are the empirical distribution functions of the first and the second sample, respectively.

The null hypothesis H_0 assumes that the samples were taken from the same distribution. The null hypothesis is rejected at the level α upon the condition:

$$D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{n \cdot m}}, \tag{3}$$

where n and m are the sizes of first and second sample respectively, and $c(\alpha)$ can be taken from a table or from the formula:

$$c(\alpha) = \sqrt{\ln\left(\frac{\alpha}{2}\right) \cdot \frac{1}{2}}. \tag{4}$$

For $\alpha = 0.05$, we have $c(\alpha) \approx 1.36$. These parameters were applied in the experiments performed.

3.2. Counting outliers in the data

By means of the Kolmogorov–Smirnov test, outliers in the data can be also identified. We modified a basic method for outlier counting. Let a given set P consists of n points. If we want to check whether the point $p \notin P$ belongs to outliers, then we should determine the cumulative distribution functions, defined as follows:

$$\begin{aligned} F_n(t) &= \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}(d_E(p, p_j) \le t), \\ F_{n^2}(t) &= \frac{1}{n^2}\sum_{k=1}^{n}\sum_{j=1}^{n} \mathbb{1}(d_E(p_k, p_j) \le t), \end{aligned} \tag{5}$$

where $\mathbb{1}()$ is an indicator function, such that:

$$\mathbb{1}(d_E(p_k, p_j) \le t) = \begin{cases} 1 & if \quad d_E(p_k, p_j) \le t \\ 0 & otherwise \end{cases}, \tag{6}$$

and $d_E(p_k, p_j)$ is the Euclidean distance between $p_k \in P$ and $p_j \in P$.

Next, we define the value $D \in [0, 1]$ as the maximum distance between $F_n(t)$ and $F_{n^2}(t)$, respectively, so:

$$D = \max_t \left|F_n(t) - F_{n^2}(t)\right|, \tag{7}$$

From the Glivenko–Cantelli theorem, it follows that points located close to each other have a smaller D value compared to those points further apart (Sharipov, 2011). Based on the D value, the set O of outliers can be created:

$$O = \{p \notin P : D > S\}, \tag{8}$$

where S is a threshold value set to 0.4.

The proposed strategy has been empirically confirmed using the same biometric data as in our previous work (Wesołowski et al., 2016), in which keystroke dynamic analysis was also analyzed. Data drift detection algorithms fall into two basic categories:

- supervised methods, in which both the class' true labels as well as the classifier's prediction labels are available for concept drift detection. In this case, drift detection is assessed on the basis of changes in the classifier's accuracy values. However this approach is impractical in real-world scenarios (Gama et al., 2014),

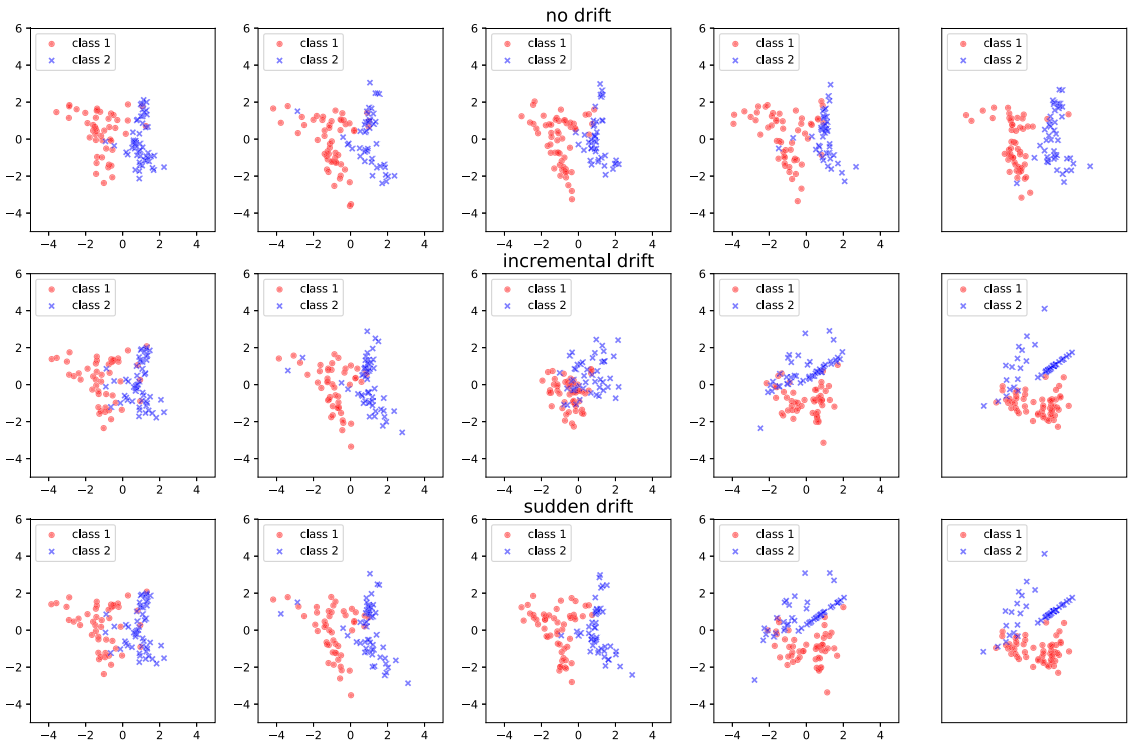

Fig. 1. Basic, real patterns of drift changes analyzed in this paper.

- unsupervised methods, in which access to the true class labels is not provided, similar to other classification approaches. In this case, drift is recognized based on the properties of the underlying data distribution.

In this paper we propose an unsupervised data drift detection strategy. This article's main contribution is the detection of drift phenomenon based on prior knowledge of the underlying data distribution, utilizing ensemble-based classification. Statistical tests will be used to evaluate this strategy.

We have assumed that knowledge about the behavioral characteristics of each user is represented in the data samples. Initially, the user's original behavioral characteristics are determined on the assumption that the data is impostor-free data and that the profiled data is available. Over time, the comparison of successive collections of incoming data with the stored user profile allows the determination of the similarity of the observed user to their earlier profiled data. A typical drift detection procedure in biometrics aims to check whether the data comes from the same distribution as the reference data:

- if the number of outliers (λ) in a set of incoming data is small (say $\lambda < a$), and if these data appear to come from the same distribution as the given user's reference data, then we assume that the original user's profile should not be changed. The user is recognized as legitimate,
- if the number of outliers (λ) is large (say $\lambda > a$) but the incoming data are seen to still originate from the same distribution as the reference data, then the primary behavioral characteristics of that user should be updated. An incremental data drift in the same, legitimate, user's data has been identified. The overall accuracy of the classification system has been improved.
- if the incoming data appears to come from a different distribution in comparison to the reference data for this given user, then we assume that an impostor of some type has taken over the originating computer. In this case, a warning alert is sent and the system stopped. This is a sudden data drift occurrence.

The events mentioned above can be analyzed by means of statistical tests. The empirical tests confirmed the effectiveness of the proposed approach.

4. Data dimensionality reduction

The classical Kolmogorov–Smirnov test was developed to evaluate activities undertaken within a one-dimensional space. Unfortunately, in biometrics, most tasks exist within a multidimensional domain. The reduction of dimensions can be performed by t-SNE (t-distributed Stochastic Neighbor Embedding) (van der Maaten and Hinton, 2008). This is a nonlinear, unsupervised dimensionality reduction technique resulting in a low-dimensional space of two or three dimensions. The algorithm starts by calculating the probability of the similarity of points in high-dimensional space and then determines the probability of similarity of the same points in the corresponding low-dimensional space. Points' similarities are calculated as the conditional probabilities that each point A would choose a specific point B as its neighbor, if neighbors were picked in proportion to their probability density under a Gaussian distribution centered at A. In the next step, the algorithm tries to minimize the difference between these conditional probabilities in higher-dimensional and lower-dimensional space in order to represent the data points in the lower-dimensional space. Fig. 2a shows a t-SNE visualization of the original biometric feature vectors $\mathbf{v} \in R^{113}$ of a legitimate user, whereas Fig. 2b depicts the case when, after some time, the input computer was taken over by an intruder. These examples come from realistic experiments undertaken specifically for this paper.

5. Ensemble based classification method

An ensemble of classifiers is a set of individual classifiers whose decisions are combined in order to produce a final decision. Most ensemble-based approaches use homogeneous structures of classifiers, though other methods do use heterogeneous classifiers. Each single classifier in an ensemble should deliver high accuracy and handle any possible high diversity (Kuncheva, 2004; Tronci et al., 2009).

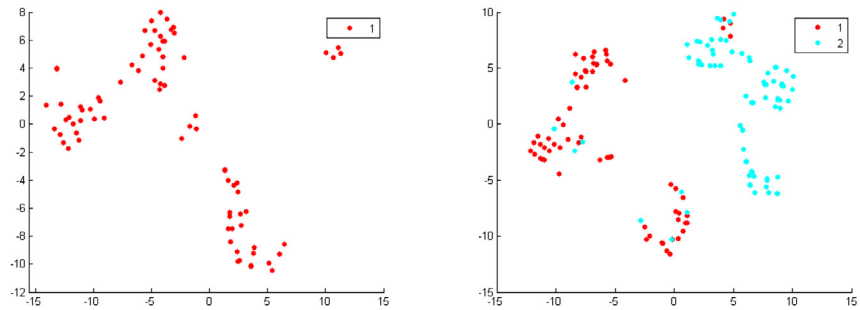

Fig. 2. Examples of legitimated (left) and intruder's injection data (right) in the reduced space.

In our previous research (Wesołowski et al., 2016) keystroke dynamics analysis was based on an ensemble of classifiers. This approach allowed us to achieve a good level of user recognition, thus the assumed level of computer security can also be accepted as relatively high. In our new strategy, we constructed an ensemble of classifiers each supported by the appropriate competences. It is demonstrated that the approach proposed here expands our previous investigations resulting in the achievement of better recognition of users when compared to our previous work (Wesołowski et al., 2016) and compared to other state-of-the-art approaches (Zhong and Deng, 2015; Teh et al., 2013; Tappert et al., 2010; Shimshon et al., 2010; Pilsung and Sungzoon, 2015). Additionally, we also introduce the Kolmogorov–Smirnov statistic to increase classification efficiency. Under the classical approach, the multi-classifier system consisted of a set of K base classifiers $Y = \{Y_1, \dots, Y_K\}$. Each base classifier realizes the function $Y_d(\mathbf{x}) : X^n \to C$, $d = \{1, \dots, K\}$ which maps from the feature space $\mathbf{x} \in X^n$ to a set of class labels $C = \{c_1, \dots, c_M\}$. For a given object $\mathbf{x} \in X^n$, a base classifier uses the function $Y_d(\mathbf{x})$ to calculate support coefficients for each class in the form of the vector $\lambda_d(\mathbf{x}) = [h_{dc_1}(\mathbf{x}), \dots, h_{dc_M}(\mathbf{x})]$ (Kuncheva, 2004). In other words, the $h_{dc_k}(\mathbf{x}) : R^n \mapsto R$ denotes a classifier's Y_d support factor for the hypothesis that the object $\mathbf{x} \in X^n$ belongs to the class $c_k \in C$. Additionally, for a given classifier Y_d some simple conditions should be fulfilled: $h_{dc_k}(\mathbf{x}) \in [0, 1]$ and $h_{dc_1}(\mathbf{x}) + \dots + h_{dc_k}(\mathbf{x}) + \dots + h_{dc_M}(\mathbf{x}) = 1$.

For the above assumptions, a single classifier produces a decision on the basis of the following formula:

$$Y_d(\mathbf{x}) \to c_k \Leftrightarrow h_{dc_k}(\mathbf{x}) = \max_{c_i \in C} h_{dc_i}(\mathbf{x}), \quad c_k \in C. \tag{9}$$

In the multi-classifier system we manage a set of classifiers. If we analyze a two-class problem, the classification process for the ensemble Y can be expressed in a compact form as follows:

$$Y(\mathbf{x}) = \begin{cases} c_1 & \text{if } \sum_{d=1}^{K} \mathbb{1}(\lambda_d(\mathbf{x}, c_1) \geq \lambda_d(\mathbf{x}, c_2)) \geq \lceil \frac{K}{2} \rceil \\ c_2 & \text{otherwise} \end{cases}, \tag{10}$$

where $\lambda_d(\mathbf{x}, c)$ stands for the support function value of the dth classifier Y_d in the ensemble Y for a given subject $\mathbf{x}$ belonging to the class c_1, or to c_2, respectively. $\mathbb{1}(con)$ is an indicator function that produces values from the set $\{0, 1\}$. The function $\mathbb{1}(con)$ returns the value 1 if the condition con is fulfilled, and 0 otherwise. The advantage of the formula (10) is its ability to be used for any type of classifier.

The ensemble of classifiers supplies advantages unavailable when a single classifier is used for classification. Additionally, the use of an ensemble has gained wide acceptance in the machine learning and statistics communities due to the significant improvement in accuracy an ensemble delivers (Kuncheva, 2004). Multiple classifier systems (often known as an ensemble or as a committee of classifiers) are convenient tools that help to decompose complex problems into sub-problems that are more easily to solve. This approach brings promising results in many domains and is widely described in the literature (Cruz et al., 2018; Souza et al., 2019).

Suppose that the available input dataset Ω will be divided into a training set $T \in \Omega$ and a validation set $V \in \Omega$, with each datum having a known class label. In machine learning, separating data into training and validating sets is an important part of evaluating a classifier model, so $T \cap V = \emptyset$ (Souza et al., 2019; Cruz et al., 2018). Let the set of some source data consist of $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in V \subset X^n$ multidimensional data points. The competence of the classifier will be performed on the set V such that the source competences of each classifier at each point $\mathbf{x}_e \in V$, $e = 1, \dots, n$ are determined.

There are many techniques which provide a measure of classifier source competence: logarithmic functions, exponential as well as piecewise linear functions (Cruz et al., 2018; Souza et al., 2019; Woloszynski and Kurzynski, 2009). In our method, we have binary classification only, so the source competence (C_s) of a given classifier Y_d can be defined as follows:

$$C_s^{Y_d}(\mathbf{x}_k, c_j) = (2 \cdot \mathbb{1}(Y_d(\mathbf{x}_k, c_j) = c_j) - 1) \cdot |\, (\lambda_d(\mathbf{x}_k, c_j) - \lambda_d(\mathbf{x}_k, c \neq c_j)\,|, \tag{11}$$

what can be simplified. Because $\lambda_d(\mathbf{x}_k, c \neq c_j) = 1 - \lambda_d(\mathbf{x}_k, c_j)$ it means that source competence can be formulated as follows:

$$C_s^{Y_d}(\mathbf{x}_k, c_j) = 2\lambda_d(\mathbf{x}_k, c_j) - 1. \tag{12}$$

$\lambda_d(\mathbf{x}_k, c_j)$ represents the support function value for the dth classifier Y_d for a given instance $\mathbf{x}_k$ that belongs to the class $c_j \in C$. In other words the instance $\mathbf{x}_k \in V$ has label $c_j \in C$.

Next, the global competence C_g of a given classifier Y_d, $d = 1, \dots, K$ at the new unknown instance $\mathbf{x}_n \in X$ and $\mathbf{x}_n \notin V$ and $\mathbf{x}_n \notin T$ is determined:

$$C_g(Y_d, \mathbf{x}_n) = \sum_{\mathbf{x}_k \in V} C_s^{Y_d}(\mathbf{x}_k, c_j) \cdot \exp\left(-dist(\mathbf{x}_n, \mathbf{x}_k)\right)^{\epsilon}, \tag{13}$$

where $dist$ is the Euclidean distance between $\mathbf{x}_n$ and $\mathbf{x}_k$.

Classifier Y_d is competent if for a given space point $\mathbf{x}_n$, $C_g(Y_d, \mathbf{x}_n) > 0$, otherwise the classifier is incompetent for the decision.

Finally, for the classifier Y_d with the greatest competence in ensemble Y its support function is calculated for the new, unknown point $\mathbf{x}_n \in X$ for authorized user:

$$Y_d \to \lambda_d^{best_1}(\mathbf{x}_n, c_1) = max\{C_g(Y_d \in Y, \mathbf{x}_n)\}, d = 1, \dots, K, \tag{14}$$

where $c_1 \in C$ stands for a label for an authorized person. Hence, $c_2 \in C$ means a label for an unauthorized user. For this case the support function for the class c_2 is formulated as follows: $\lambda_d^{best_2}(\mathbf{x}_n, c_2) = 1 - \lambda_d^{best_1}(\mathbf{x}_n, c_1)$.

The Gaussian potential function in (13) strongly promotes the $\mathbf{x}_n$ elements lying close to $\mathbf{x}_k$, while penalizing other elements. In the research, the parameter ϵ was set to 3.0 because, for this value, there is a balance between ACC, FAR and FRR. It should be noted that in practice, changes in the parameter ϵ affect the biometric system (Porwik et al., 2019). The FAR (False Acceptance Rate) and FRR (False Rejection Rate) factors will be defined later.

The ensemble of classifiers is dynamically formed. It means that a different, unknown new instance $\mathbf{x}$ will be always classified by the best classifier from a given classifiers pool Y.

$$
\begin{aligned}
\mathbf{v}_1^{uid} &= [\text{keyDown, 0,TAB}] \\
\mathbf{v}_2^{uid} &= [\text{keyUp, 55, TAB}] \\
\mathbf{v}_3^{uid} &= [\text{keyDown, 182, a}] \\
\mathbf{v}_4^{uid} &= [\text{keyUp, 445, a}] \\
\mathbf{v}_5^{uid} &= [\text{keyDown, 675, b}] \\
\mathbf{v}_6^{uid} &= [\text{keyUp, 1323, b}] \\
\mathbf{v}_7^{uid} &= [\text{keyDown, 1690, ENTER}] \\
\mathbf{v}_8^{uid} &= [\text{keyUp, 2203, ENTER}]
\end{aligned}
$$

Fig. 3. Example of vectors with user's activity.

6. Biometric profile of the user

A user's behavioral profile is built on the basis of keystroke analysis. Our strategy consists of three consecutive stages. In the first step, we construct a user profile from the volunteers who participated in our experiment in which real-life keystroke dynamics features were recorded. Each volunteer generated their own personal biometric profile based on a series of input data.

In the second step, by utilizing drift detection analysis, a pool of classifiers, and a classifier competence strategy, the best classifiers in the pool were selected based on their quality recognition of a given user's biometric profile.

In the third phase, the biometric model was evaluated using a 10-fold cross-validation strategy (Gareth et al., 2014).

6.1. User's profile parameters

As stated, for each new user with access to a computer, an individual behavioral keystrokes profile was formed. This profile's parameters have been broadly discussed in previous work (Wesołowski et al., 2016; Alsultan et al., 2018). The activity of each user's *uid* was separately monitored by an analysis of keystroke time dependencies. This is undertaken similarly to other classical measurements as described in, amongst others, Wesołowski et al. (2016) and Alsultan et al. (2018). The measurements are relatively simple. In this paper they will only be briefly discussed.

The parameters of each jth keyboard event for a given user *uid* are stored in the vector $\mathbf{v}_j^{uid}$:

$$\mathbf{v}_j^{uid} = [type_j, t_j, \omega_j], \tag{15}$$

where: $type_j \in \{keyDown, keyUp\}$ details the type of an event; t_j is the timestamp of a given event; ω_j is an encoded identifier for a key used. For simplification, in this example key are not coded for a better understanding of the method. All vectors $\mathbf{v}_j^{uid}$ constitute a user's activity dataset (see Fig. 3).

Based on vectors stored in containers $G^k(de), k = 1, \ldots, 113$, $de \in \{desc, desc^*\}$, a feature vector $\mathbf{F}^{uid} = [f_1, f_2, \ldots, f_{113}]$ is constructed as follows. Let k denote that the G^k container is full. Then the kth element f_k of the vector $\mathbf{F}^{uid}$ is calculated from those vectors stored in the container $G^k(de) = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{g^{uid}}\}$.

In the other 112 containers $G^j(de)$, $j \in \{1, 2, \ldots, i, i+1, \ldots, 112\}$ and $i \neq k$, the number of vectors $\mathbf{d}$ can be different, depending on the activity of the user controlling the computer keyboard. Let n_k be the number of vectors $\mathbf{d}$ in the container G^k. Features are calculated according to the formula:

$$f_k = \begin{cases} 0 & \text{when } n_k = 0 \\ \sqrt{\dfrac{1}{n_k} \displaystyle\sum_{c=1}^{n_k} \left((t_{j+1} - t_j) - \bar{t} \right)^2} & \text{when } n_k > 0 \end{cases} \tag{16}$$

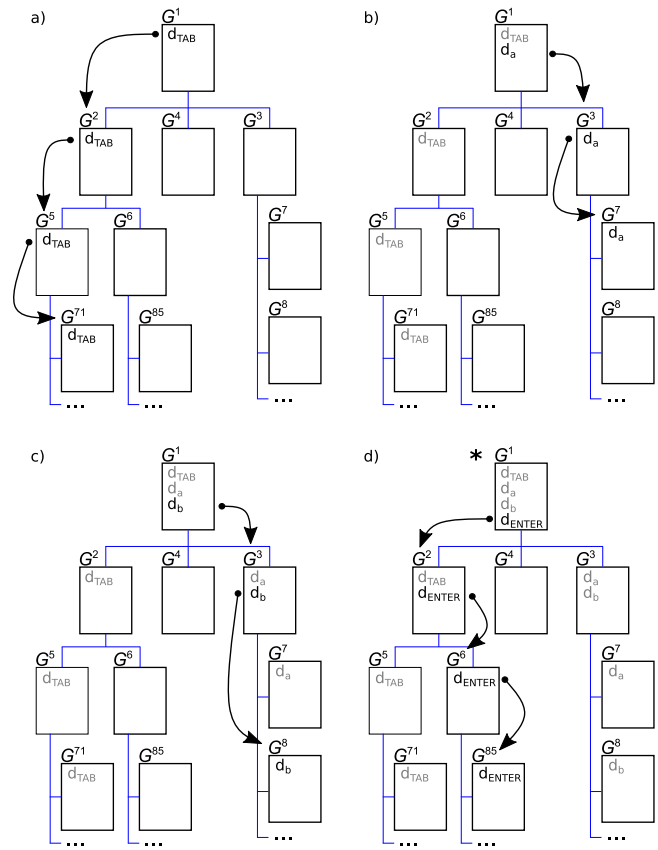

Fig. 4. The principle of storing vectors $\mathbf{v}_i^{uid}$ in containers G in steps (a), (b) (c) and (d). The asterisk (*) on the figure (d) informs us that the container is full and feature creation can be undertaken.

and $t_j, t_{j+1} \in \mathbf{d}_c = [t_j, t_{j+1}, \omega_j, \omega_{j+1}]$, $\bar{t} = \frac{1}{n_k} \sum_{c=1}^{n_k} (t_{j+1} - t_j)$.

It should be noted that the formula (16) is applied only in those cases where a given container G^k is full ($n_k == g^{uid}$). The principles of the containers' population are presented in Fig. 4.

If any container is populated, the vector $\mathbf{F}^{uid}$ of the user's features is created. The container that has a number of elements equal to g^{uid} is cleared and the process is continued until the required number of feature vectors $\mathbf{F}^{uid}$ is reached.

This process is presented in Algorithm 1. An analysis of this algorithm shows that, for small values of the parameter g, many containers are empty. So it is important to find a compromise between the optimal value of the #**F** parameter, the g parameter, and the accuracy (ACC) of the system. Based on the experiments described in Wesołowski et al. (2016), the number of vectors **F** may empirically be set to 100 and the capacity g of each container G set to 10.

7. Classification process

Changes in a given user's behavioral characteristics are evaluated using the Kolmogorov–Smirnov statistical test. In the learning, supervised phase, the profile of a given user is developed. This profile represents a person's behavioral characteristic. For the statistical purposes, multidimensional data are converted into one-dimensional data by application of the t-SNE algorithm (van der Maaten and Hinton, 2008). If the incoming data belong to the same distribution as a user's primary profile and any drift in the data is not radical, then the user's profile is modified and the classifier Y is learned again. A given user is recognized by their profile and by an outliers analysis. This ensures that classifier accuracy is maintained. Algorithm 2 presents the main steps of drift detection using a statistical method and an ensemble of classifiers.

Algorithm 1: Feature vectors' creation of each user's uid based on the containers' G content

```
/* #G - The current number of elements (vectors v)
   in a given container G.                          */
Input: vectors v stored inside of the containers G
r - the number of vectors F required for a user's uid profile
preparation.
g^uid - the common max. number of elements over all containers G.
Output: a set of a user's feature vectors F
i := 0;
while i < r do
    populate containers G with the vectors v as long as condition
    #G == g^uid is fulfilled by any container G;
    if any #G == g^uid then
        form the vector F_i from all vectors v stored in all
        containers G;
        clear only this container for which an event #G == g^uid
        has occurred.
        i:=i+1;
    end
end
```

These strategies enable high classification accuracy whenever the input data change. If only an authorized individual uses the computer, then the security system achieves its best performance. When unauthorized users utilize the device, the system is halted. In contrast to the original idea of concept drift in which drift detection classifiers are to be trained again while the system remains working, in a biometric system such phases have to be detected and an illegitimate user's operations halted. These steps have been shown in Algorithm 2. Authorized and soft behavioral changes do not raise an alarm. By combining statistical data distribution analysis and statistical-based outlier counting, we can effect a change in the quality of the biometric system.

8. Experiments performed

The evaluation of competence-based classifiers supported by the two-sample Kolmogorov–Smirnov test has been successfully empirically confirmed. A range of classifier models was compared. Unfortunately, a number of issues related to the keystroke dynamics data sets used for research into user profiling, authorization, and into verification make for a difficult comparison between the different approaches: datasets are unavailable, datasets vary widely in size, and the datasets are often private. In our comparative studies, our comparison of various keystroke dynamics strategies generated both critical and positive commentary.

8.1. Performance measures

Evaluation of the proposed method was checked in hybrid form using different methods:

- Accuracy (ACC). Accuracy is a classical and frequently used measure for classifier performance:

$$ACC = \frac{TN + TP}{TN + TP + FP + FN}, \tag{17}$$

where TN represents the number of negative instances classified correctly. If positive instances are classified as negative, this is taken to be a false negative (FN). FP represents the number of negative instances that are misclassified as positive. TP is the number of correctly classified positive instances.

- Three of the most commonly used measures in biometrics are the false acceptance rate (FAR), false rejection rate (FRR) and equal error rate (EER).

$$FAR = \frac{FP}{FP + TN} \tag{18}$$

$$FRR = \frac{FN}{TP + FN} \tag{19}$$

FAR represents the percentage of identification instances in which unauthorized persons are incorrectly accepted, whereas FRR is the percentage of identification instances in which authorized persons are incorrectly rejected. EER indicates that the proportion of false acceptances is equal to the proportion of false rejections.

- The Bayesian signed-rank test. In this test two classifiers are compared and the probability is computed that: (a) one method is better than the other, (b) vice versa, (c) that the methods are equivalent (Benavoli et al., 2017).

8.2. Datatsets

We conducted the experiments on our own database consisting of 150 various biometric profiles. These profiles were recorded during normal work by computer users. These profiles are imposter-free datasets. Each profile consists of the 100 vectors **F** (see previous sections of the paper). The preparation of these data sets was necessary because there are no other benchmark data sets devoted to the problem being analyzed. The data set created to conduct all the experiments here is openly accessible at the address http://dx.doi.org/10.17632/vfgn7dd2z4.1. This allows experimental reproduction and for others to conduct their own experiments and to compare the results they obtain. The experiments were carried out utilizing a stratified 10-fold cross-validation technique (Gareth et al., 2014).

8.3. Experiments

Set up (1): The aim of the first experiment was to investigate the behavior of various user recognition classifiers on the basis of keystroke analysis. Our results are presented in Table 2. It is difficult to anticipate which classifier is more or less useful for a given dataset analysis (Burduk and Baczyńska, 2016; Cruz et al., 2018; Souza et al., 2019). For this reason, we suggest using various tested algorithms to create a pool of classifiers. On the basis of this pool of classifiers, the accuracy of the biometric system can be estimated. The quality of classification was measured by various coefficients including Accuracy (ACC), Equal Error Rate (ERR), False Acceptance Rate (FAR) and False Rejection Rate (FRR).

In order to ensure the correctness of the observed results in different learning methodologies, we assessed the considered algorithms in different classification algorithms. In the comparative study, we evaluated canonical classifiers as demonstrated in the previous literature while our classifiers were implemented using many mathematical packages including Matlab, R and WEKA systems: Random Forests (RF) (Breiman, 2001), Bayes Net (BN) (Muramatsu et al., 2006), J48 - C4.5 decision trees (J48) (Quinlan, 1993), Support Vector Machine (SVM) (Ma and Guo, 2014), Random Tree (RT) (Breiman, 2001), RIDOR RIpple-DOwn Rule learner (Ridor) (Quinlan, 1993), Naive Bayes (NB) (Johnson and Bhattacharyya, 2019; Muramatsu et al., 2006).

Some classifiers, while attractive from the classification quality point of view, cannot be used in the proposed approach. For example, learning deep networks like CNN (Convolutional Neural Networks) because of the many internal layers and convolution operations always requires large datasets. Valueva et al. (2020). The main assumption of our method was to limit the necessary data for classifier training.

In our investigations different classifier configurations were checked: classifiers in which a classification rule was calculated on the basis of a majority voting scheme, and classifiers in which, for each

Algorithm 2: Classifier preparation and control of a user's behavior

Notation: DW - data window (20 vectors **F**) of incoming data.
λ - number of outliers in a dataset DW.
P_{uid} - profile of a user uid consists of 100 vectors **F**.
K–S - the Kolmogorov–Smirnov test on the two sets
P_{uid} and DW : K-S(P_{uid}, DW):
H_0: samples are drawn from the same distribution K-S()=*True*
otherwise K-S()=*False*
Input: Legitimate user's uid keystroke profile P_{uid} consisting of an imposter-free 100 vectors **F**
Output: alert if an unauthorized user is detected

```
/* Preparation for protection of the computer */
```

On the basis of the profile P_{uid}, an ensemble Y of single classifiers is trained, tested and validated to determine the selection of classifiers having the greatest competences.

```
/* Control of user's uid behavior via anomaly scores */
```

while *user's uid data stream is available* **do**
  The collect new data window DW
  if K-S(P_{uid}, DW)=*False* **then**
    ALERT "An unauthorized user has been detected" generated. Computer is blocked.
  else
    if $\lambda > 5$ **then**
      update the user's profile P_{uid}: replace the 20 oldest vectors **F** in the behavioral profile. Train the classifier again.
    end
  end
end

Table 1
Parameters of the classification and the algorithms used in experiments.

Algorithm	Parameters
RF	Number of trees: 10. Kernel = linear
BN	–
C4.5	Pruned, Decision trees, Number of trees: 100; Minimum number of instances per leaf: 2
SVM	$C \in (0.01, 0.05, \ldots, 1.0)$. Training procedure = SMO, Kernel = RBF
Random Tree	The number of randomly chosen attributes: log_2(number of attributes)
Ridor	Number of folds: 3; Number of shuffles: 1; The minimal weights of instances within a split: 2
Naive Bayes	–

single classifier in the pool, the classifier's competence had already been earlier computed and data drift monitored.

Throughout the conducted experiments, the proposed method was compared with selected, mentioned above, modern machine learning algorithms. The parameters of the reference methods used in the experimental studies are presented in Table 1.

Table 2 shows that a classification strategy with competent classifiers results higher ACC, EER, FAR and FRR values in comparison to a classifier ensemble based on majority voting. Using the same classifiers in "single-mode" we always obtained the worst results.

In Table 3 the same classifiers were employed, but now, we introduced a strategy of alerts based on the Kolmogorov–Smirnov statistic and outlier detection. The proposed modification for dynamic keystroke analysis worked better when classifiers having "competency" were used. This results forms the data presented in Table 2 and in Table 3. The analysis of Table 3 confirms that the best Accuracy (ACC) was achieved by a classification strategy supplemented by statistic-based alerts which ensure that the system can be also employed by users who do not have stable biometric behavior. This here is a core advantage of the proposed solution: Soft behavioral changes are automatically corrected due to data drift detection.

Set up (2): Different ensemble structures can be evaluated in more detail by taking into consideration the statistical characteristic of the various classification strategies using a Bayesian Signed-Rank Test (Benavoli et al., 2017). This test is the Bayesian equivalent of the Wilcoxon signed-rank test. In the reported Bayesian test, the Authors introduce the *rope* concept, the "Region Of Practical Equivalence" for every two classifiers. If this parameter was set to 0.01, then the two classifiers are considered equivalent if the difference in their performance is smaller than *rope*. The test results in three possible outcomes: (a) one method is better than the other, (b) vice versa, (c) they are equivalent.

Table 2
Classification results for various types of classifier. Comparison of single classifiers as well as classifiers with voting and with a competence-based strategy. The presented results are the average values after twenty experiments.

Classifier	ACC [%]	EER [%]	FAR [%]	FRR [%]
Ensemble[a]	98.43 ± 0.25	1.57 ± 0.01	1.23 ± 0.00	1.66 ± 0.00
Ensemble[b]	97.91 ± 0.15	2.09 ± 0.00	1.81 ± 0.00	2.67 ± 0.01
RF	87.99 ± 0.23	12.01 ± 0.02	9.73 ± 0.02	15.07 ± 0.02
BN	83.43 ± 0.17	16.57 ± 0.07	14.15 ± 0.03	20.21 ± 0.03
C4.5	82.59 ± 0.29	17.41 ± 0.01	14.63 ± 0.01	20.76 ± 0.01
SVM	80.38 ± 0.14	19.62 ± 0.02	14.55 ± 0.03	20.21 ± 0.02
Random Tree	78.27 ± 0.24	21.73 ± 0.04	19.52 ± 0.04	29.63 ± 0.05
Ridor	74.63 ± 0.20	25.47 ± 0.02	23.11 ± 0.02	33.27 ± 0.02
Naive Bayes	80.58 ± 0.17	19.42 ± 0.02	17.03 ± 0.01	24.08 ± 0.02

[a]Ensemble of classifiers with competences.
[b]Ensemble of classifiers with majority voting without competences.
Rest of classifiers are single classifiers.

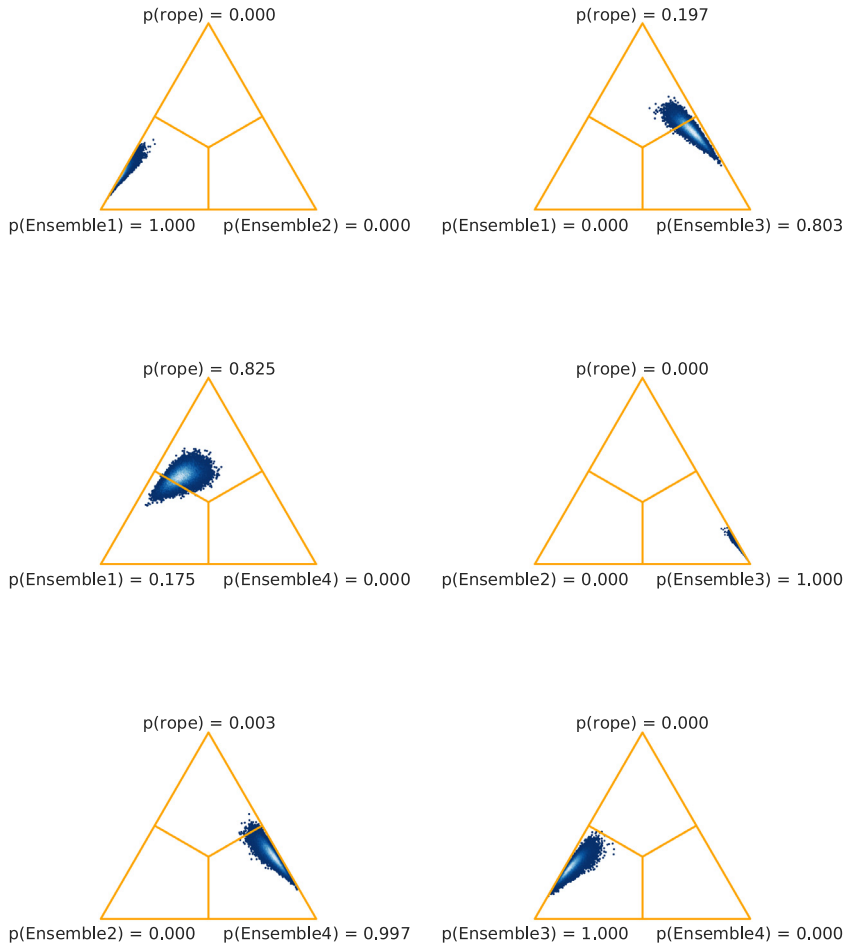

Fig. 5. Posteriors for the Bayesian sign-rank tests for Accuracy measure for various ensemble-based classification strategies. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Classification results for each type of classifier supported by each classifier's competences and by Kolmogorov–Smirnov alerts. The presented results are average values following twenty experiments.

Classifier	ACC [%]	EER [%]	FAR [%]	FRR [%]
Ensemble[c]	98.96 ± 0.22	1.04 ± 0.00	0.87 ± 0.00	1.65 ± 0.00
Ensemble[d]	98.37 ± 0.19	1.63 ± 0.00	1.37 ± 0.00	2.21 ± 0.00
RF	92.31 ± 0.15	7.69 ± 0.02	5.94 ± 0.01	14.33 ± 0.03
BN	90.64 ± 0.15	9.36 ± 0.02	7.28 ± 0.07	13.67 ± 0.02
C4.5	89.88 ± 0.19	10.12 ± 0.00	8.78 ± 0.04	15.65 ± 0.01
SVM	86.59 ± 0.20	13.41 ± 0.02	11.21 ± 0.02	16.05 ± 0.01
Random Tree	83.84 ± 0.19	16.16 ± 0.00	14.92 ± 0.01	19.91 ± 0.02
Ridor	84.28 ± 0.16	15.72 ± 0.03	10.42 ± 0.03	19.10 ± 0.01
Naive Bayes	86.80 ± 0.24	13.20 ± 0.02	11.47 ± 0.02	14.84 ± 0.02

[c]Ensemble of classifiers with competences and alerts.
[d]Ensemble of classifiers with majority voting without competences and with alerts.
The remainder of the classifiers are single classifiers having an alert strategy.

Fig. 5 shows the posteriors for the Bayesian sign-rank tests. In these triangles, the bottom-left and bottom-right regions correspond to the cases where one classification method is better than the other, and vice versa. The top region represents those cases where the equivalence between the methods is more probable. If all the points (here blue) fall inside one particular region, we conclude that the hypothesis ("that one method is better than the other") is true with a probability of ≈ 1 (Benavoli et al., 2017). Looking at Fig. 5, it is evident that the majority of cases support Ensemble 3 and Ensemble 4 more than *rope* and definitively more than Ensemble 1 and Ensemble 2. So, the strategy utilizing Kolmogorov–Smirnov statistical alerts results in more benefits than the other approaches.

The results of the proposed classification models were also compared with other methods. Comparative studies of this problem are presented in Table 4. There are a number of issues related to the data sets used in research into computer user profiling based on keystroke analysis. First of all, there are no standards for data collection and they are in different fields of biometrics. For this reason, it is difficult to compare different researchers' work. Further details on the issues concerning such data sets can be found in the literature (Wesołowski et al., 2016). The classification quality of the methods was represented by two popular measures: Accuracy (ACC) and Equal Error Rate (EER).

Conclusions

The method proposed enhances classification accuracy, without supervision, by providing additional information related to the possible data distributions of various individuals. The knowledge of the data distribution allows to use the Kolmogorov–Smirnov statistic and outliers counting. In this paper a novel biometric classification method based on classifiers having competences and statistics is introduced. Competences are computed separately for each classifier in a pool of base classifiers. The proposed solution applied to the analysis of keystroke dynamics helps to confirm the identity of an individual based on that individual's typing pattern and the speed of their typing on an input computer's keyboard. The advantages of the method can be summarized as follows:

Table 4
A comparison of intrusion detection methods.

Method	Accuracy [%]	EER [%]	Data type[a]
Our method	**98.96**	**1.04**	**Continuous activity**
Wesołowski et al. (2016)	97.83	2.17	Continuous activity
Lopatka and Peetz (2009)	84.00	20.00	Long-text (free text, not continuous)
Tappert et al. (2010), identification, optimal conditions	97.40	*	Long-text (retyping, 650 characters, not continuous)
Tappert et al. (2010), identification, optimal conditions	95.60	*	Long-text (free text, 650 characters, not continuous)
Tappert et al. (2010), identification, non-optimal conditions	60.75	*	Long-text (free text, 650 characters, not continuous)
Tappert et al. (2010), verification	90.20–95.6	*	Long-text (free text, 650 characters, not continuous)
Zhong et al. (2012)	*	8.40	Repeated short sequence (password)
Pleva et al. (2015)	97.03	*	Repeated short sequence (password)
Monaco et al. (2013)	91.50	*	Long-text (free text, not continuous)

[a]*continuous activity* - analysis of a user's activity while performing everyday tasks, *long text* – users type a text freely or retype a given text; acquisition is limited by the time taken to write or by the length of the input text, *repeated short sequence* – a single word (password) repeated many times.
* results are not available.

- Experimental results demonstrate that classifiers supported by competences and by statistical evaluation of the data distribution result in a better classification quality as compared to other techniques (see results presented in Tables 2 and 3). This was in addition confirmed by application of the statistical Bayesian sign-rank test.
- We report that results deliver additional advantages. The Accuracy (ACC) coefficient is higher and the FAR/FRR coefficients are lower. A system equipped with classifiers having competences supported by Kolmogorov–Smirnov tests ensures sufficient effectiveness.
- Based on the modified Kolmogorov–Smirnov test, outliers in data can also be discovered. This is a novel technique applicable also in other domains.
- The proposed approach can be implemented in parallel mode so that classifiers can work concurrently and separately. This can significantly reduces computation time.
- Investigations were carried out using the databases made freely available at http://dx.doi.org/10.17632/vfgn7dd2z4.1. These source databases can be used in the future by any persons interested in undertaking comparative studies.

The solution proposed employs only primary data. This new approach allows us to detect impostors based on statistical tests and an ensemble of classifiers. The method does not depend on any knowledge of the typed text, which makes it very convenient for users. In competitive methods, the text must be processed in an open form. In our approach, each user's data was encoded, making the technique appropriate for systems dealing with sensitive data. If the user works slowly (e.g. only presses a few keys), the system cannot react, but the user will not damage the system.

CRediT authorship contribution statement

Piotr Porwik: Conceptualization, Supervision, Formal analysis, Data preparation, Methodology, Writing - original draft. **Rafal Doroz:** Conceptualization, Software, Visualization, Data curation, Investigation, 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.

References

Alsultan, A., Warwick, K., Wei, H., 2018. Improving the performance of free-text keystroke dynamics authentication by fusion. Appl. Soft Comput. 70, 1024–1033.
Andress, J., 2014. The Basics of Information Security, second ed. Syngress, Boston.
Banerjee, S.P., Woodard, D.L., 2012. Biometric authentication and identification using keystroke dynamics: A survey. J. Pattern Recogn. Res. 7 (1), 116–139.
de Barros, R.S.M., de Carvalho Santos, S.G.T., 2019. An overview and comprehensive comparison of ensembles for concept drift. Inf. Fusion 52, 213–244.
Benavoli, A., Corani, G., Demšar, J., Zaffalon, M., 2017. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. J. Mach. Learn. Res. 18 (77), 1–36.
Bonab, H.R., Can, F., 2018. GOOWE: Geometrically optimum and online-weighted ensemble classifier for evolving data streams. ACM Trans. Knowl. Discov. Data 12 (2).
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
Brzezinski, D., Stefanowski, J., 2014. Combining block-based and online methods in learning ensembles from concept drifting data streams. Inform. Sci. 265, 50–67.
Burduk, R., Baczyńska, P., 2016. Dynamic ensemble selection using discriminant functions and normalization between class labels – approach to binary classification. In: Artificial Intelligence and Soft Computing. Springer International Publishing, Cham, pp. 563–570.
Cruz, R.M., Sabourin, R., Cavalcanti, G.D., 2018. Dynamic classifier selection: Recent advances and perspectives. Inf. Fusion 41, 195–216.
Gama, J., Žliobaite, Bifet, A., Pechenizkiy, M., Bouchachia, A., 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46 (4).
Gareth, J., Daniela, W., Trevor, H., Robert, T., 2014. An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated.
Gonçalves, Jr., P., Santos, S., Barros, R., Vieira, D., 2014. A comparative study on concept drift detectors. Expert Syst. Appl. 41, 8144–8156.
Hoens, T.R., Polikar, R., Chawla, N.V., 2012. Learning from streaming data with concept drift and imbalance: an overview.. Prog. AI 1 (1), 89–101.
Johnson, R.A., Bhattacharyya, G.K., 2019. Statistics: Principles and Methods. Wiley.
Kuncheva, L., 2004. Combining Pattern Classifier: Methods and Algorithms. Wiley Interscience, New Yersey.
Lopatka, M., Peetz, M.H., 2009. Vibration sensitive keystroke analysis. In: Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning. pp. 75–80.
Lu, N., Zhang, G., Lu, J., 2014. Concept drift detection via competence models. Artificial Intelligence 209, 11–28.

Ma, Y., Guo, G., 2014. Support Vector Machines Applications. In: Springer Science & Business Media, Springer International Publishing.

van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605.

Massey, F.J., 1951. The Kolmogorov-Smirnov test for goodness of fit. J. Amer. Statist. Assoc. 46 (253), 68–78.

Monaco, J.V., Stewart, J.C., Cha, S.-H., Tappert, C.C., 2013. Behavioral biometric verification of student identity in online course assessment and authentication of authors in literary works. In: Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on. pp. 1–8.

Muramatsu, D., Kondo, M., Sasaki, M., Tachibana, S., Matsumoto, T., 2006. A Markov chain Monte Carlo algorithm for bayesian dynamic signature verification. IEEE Trans. Inf. Forensics Secur. 1 (1), 22–34.

Pesaranghader, A., Viktor, H.L., Paquet, E., 2018. McDiarmid drift detection methods for evolving data streams. In: 2018 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–9.

Pilsung, K., Sungzoon, C., 2015. Keystroke dynamics-based user authentication using long and free text strings from various input devices. Inform. Sci. 308, 72–93.

Pleva, M., Kiktova, E., Juhar, J., Bours, P., 2015. Acoustical user identification based on MFCC analysis of keystrokes. Adv. Electr. Electron. Eng. 13 (4), 309–313.

Porwik, P., Doroz, R., Wrobel, K., 2019. An ensemble learning approach to lip-based biometric verification, with a dynamic selection of classifiers. Expert Syst. Appl. 115, 673–683.

Quinlan, R.J., 1993. C4.5: Programs For Machine Learning. In: The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, San Mateo, Calif..

Sharipov, O.S., 2011. Glivenko–Cantelli theorems. In: International Encyclopedia of Statistical Science. Springer Berlin Heidelberg, pp. 612–614.

Shimshon, T., Moskovitch, R., Rokach, L., Elovici, Y., 2010. Continuous verification using keystroke dynamics. In: 2010 International Conference on Computational Intelligence and Security. pp. 411–415.

Souza, M.A., Cavalcanti, G.D., Cruz, R.M., Sabourin, R., 2019. Online local pool generation for dynamic classifier selection. Pattern Recognit. 85, 132–148.

Tappert, C.C., Villani, M., Cha, S.-H., 2010. Keystroke biometric identification and authentication on long-text input. Behav. Biometr. Human Identif.: Intell. Appl. 342–367.

Teh, P.S., Teoh, A.B., Yue, S., 2013. A survey of keystroke dynamics biometrics. Sci. World J. 2013.

Tronci, R., Giacinto, G., Roli, F., 2009. Designing multiple biometric systems: Measures of ensemble effectiveness. Eng. Appl. Artif. Intell. 22 (1), 66–78.

Valueva, M., Nagornov, N., Lyakhov, P., Valuev, G., Chervyakov, N., 2020. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Math. Comput. Simulation 177, 232–243.

Wesołowski, T.E., Porwik, P., Doroz, R., 2016. Electronic health record security based on ensemble classification of keystroke dynamics. Appl. Artif. Intell. 30 (6), 521–540.

Woloszynski, T., Kurzynski, M., 2009. On a new measure of classifier competence in the feature space. In: Computer Recognition Systems, Vol. 3. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 285–292.

Zhong, Y., Deng, Y., 2015. A Survey on Keystroke Dynamics Biometrics: Approaches, Advances, and Evaluations. Science Gate Publishing, pp. 1–22.

Zhong, Y., Deng, Y., Jain, A.K., 2012. Keystroke dynamics for user authentication. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 117–123.