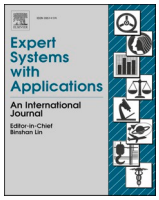
Expert Systems With Applications 235 (2024) 121201

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Distance-based one-class time-series classification approach using local cluster balance

Toshitaka Hayashi [a,1], Dalibor Cimr [a,2], Filip Studnička [a,3], Hamido Fujita [b,c,d,*,4], Damián Bušovský [a,5], Richard Cimler [a,6], Ali Selamat [b,7]

[a] Faculty of Science, University of Hradec Kralove, Hradecká 1285, Hradec Kralove, Czech Republic
[b] Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur, 54100, Malaysia
[c] Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain
[d] Regional Research Center, Iwate Prefectural University, Takizawa 020-0693, Japan

ARTICLE INFO

Keywords:
One-class classification
Timeseries classification
Clustering
Cluster Balance

ABSTRACT

Deciding the signal length is an important challenge for one-class time-series classification (OCTSC). This paper aims to develop an OCTSC algorithm that does not require model retraining for different signal lengths. For this purpose, a distance-based one-class time-series classification approach using local cluster balance (OCLCB) is proposed. OCLCB extracts feature vectors, namely, local cluster balance (LCB), from the clustering results of sliding windows. K-means clustering is applied to the sliding windows extracted from the training signal. Then, the local prototype (LP) is calculated as the average of the local cluster balance (LCB) in the training data. Unseen scores are computed as the distance metrics between LP and LCBs in the testing data. Since the sliding window size is independent of the entire signal size, OCLCB does not need to retrain the model. This aspect gives the benefit of reducing the parameter tuning costs. The source code is uploaded at https://github.com/ToshiHayashi/OCLCB.

1. Introduction

Time-series data are a set of observations obtained in chronological order, represented as a sequence or signal, and processed in various areas, such as biometric signals (Merdjanovska and Rashkovska, 2022), financial records (D'Urso et al., 2020), or weather records (Wang et al., 2021). In the context of time-series data, time-series classification (TSC) is a supervised classification problem. Several algorithms have been proposed for TSC (Rjoob et al., 2022). However, classification results can be affected by issues with training data, such as data imbalance (Hernandez-Matamoros et al., 2020), noisy labels (Karmy et al., 2021), and outliers (Alimohammadi and Chen, 2021). In addition, the model cannot classify data into unseen classes that were not included in the training data. The challenge of unseen classes is critical because some data are rare or nonexistent.

One-class classification (OCC) (Schölkopf et al., 2001) is a special type of supervised classification problem in which training data have only one class. The goal is to classify data into one seen class or other unseen classes. A solution to this problem is unsupervised learning because training data from only one class is not helpful for supervised classification. OCC is also known as anomaly detection (AD) in wider terms. AD methods are classified into supervised learning, where training data are fully labeled, unsupervised learning, where data are unlabeled and normal and abnormal data are mixed, and semisupervised

* Corresponding author at: Kotorizawa, 2-27-5, Morioka, Iwate 020-0104, Japan.
E-mail addresses: toshitaka.hayashi@uhk.cz (T. Hayashi), dalibor.cimr@uhk.cz (D. Cimr), filip.studnicka@uhk.cz (F. Studnička), fujitahamido@utm.my, HFujita-799@acm.org (H. Fujita), damian.busovsky@uhk.cz (D. Bušovský), richard.cimler@uhk.cz (R. Cimler), aselamat@utm.my (A. Selamat).
[1] 0000-0002-7599-4404.
[2] 0000-0003-2197-8553.
[3] 0000-0001-6721-8678.
[4] 0000-0001-5256-210X.
[5] 0000-0001-8853-8224.
[6] 0000-0001-6712-9894.
[7] 0000-0001-9746-8459.

https://doi.org/10.1016/j.eswa.2023.121201
Received 8 April 2023; Received in revised form 12 August 2023; Accepted 12 August 2023
Available online 15 August 2023

learning, where labeled and unlabeled data are mixed. OCC algorithms are applicable in all settings, as unsupervised solutions do not rely on annotations.

The main advantage of OCC algorithms is their ability to detect unseen classes. This aspect can support supervised learning with the strategy to train data class by class; OCC does not have data imbalance problems because the class label is only one. Moreover, noise and outliers can be detected by cross-validation. Accordingly, OCC is a promising approach to solving a wide range of training dataset problems in supervised classification.

One-class time-series classification (OCTSC) is a type of OCC for time-series data. OCTSC is classified into three levels (Li and Jung, 2022): time-point level, time-interval level (subsequence of the signal), and time-series level (the entire signal). However, existing OCTSC methods (Mauceri et al., 2020, Blázquez-García et al., 2021, Hayashi et al., 2022) have a common weakness in that the algorithms need to retrain the machine learning model in order to analyze signals of different lengths. This aspect increases the parameter tuning cost.

In summary, the motivation of this paper is to develop an OCTSC algorithm that does not require model retraining for signals of different lengths. One of the gaps in the literature is that current models process the entirety of a signal simultaneously, which limits their applicability to signals of varying lengths. Instead, the models should learn from arbitrary inputs that are robust to changes in signal length. This will enable a more flexible and versatile application of the models to a wider range of signals.

For this purpose, a novel OCTSC algorithm, namely, a distance-based one-class time-series classification approach using local cluster balance (OCLCB), is proposed by considering three key questions. The first question is how to ensure that a model input is robust to changes in signal length. In this study, sliding windows are used as model input because the window size is independent of the overall signal length. The second question concerns the algorithm used to create the model. An unsupervised clustering model is applied since sliding windows do not have annotations. The third question is how to use clustering results to analyze an entire signal. This study uses cluster balance, which represents the amount of data for all clusters, as a feature vector to approximate the balance of shapes in the signal since sliding windows that belong to the same cluster should have similar shapes. Moreover, the feature dimension is equal to the number of clusters, making it robust to changes in signal length.

In the end, unseen scores can be computed using the distance metric between cluster balances for training and testing data. To accomplish this, local cluster balance (LCB) is proposed as a feature vector of the local area of the signal. The size of the local area corresponds to the signal length. LCB can be computed for various signal lengths without requiring retraining of the clustering model because the model inputs are sliding windows.

The contributions of this paper are as follows:

- A distance-based one-class time-series classification approach using local cluster balance (OCLCB) is proposed. The main novelty is using the clustering results of sliding windows for feature extraction from the signal. In particular, LCBs are proposed as feature vectors in the local area of the signal.
- Evaluation was performed by authentication and change detection tasks using synthetic signals (created by connecting two sine waves) and two real-world datasets. OCLCB displayed fair performance in terms of both accuracy and processing speed.
- OCLCB does not need model retraining to process different lengths of signals. Such an advantage reduces parameter tuning costs.
- OCLCB does not require a graphics processing unit (GPU).

The organization of the paper is as follows. Section 2 describes related work and its shortcomings. Section 3 presents the proposed OCLCB algorithm, followed by Sections 4 and 5, which present the experimental results and discussions, respectively. Finally, Section 6 concludes the paper and provides avenues for future work.

2. Related work

2.1. Time-series classification

The first stage of time-series classification (TSC) is preprocessing, which aims to transform raw signals into a suitable form for classification. Several preprocessing methods have been applied by researchers, including wavelet transform (Barua et al., 2023), adaptive thresholding (Sadek et al., 2018), Cartan curvature (Cimr et al., 2020), and Euclidean arc length (Cimr et al., 2021). The main challenge in preprocessing is how to divide the signal. Generally, signals from sensor data are split into arbitrary-sized subsignals. One of the common approaches is to use sliding windows, which are subsequences of the signal extracted by sliding. Sliding windows are obtained by two parameters: window size and stride. A smaller stride can increase the amount of data, which can provide higher accuracy but also increase processing time.

Afterward, the classification stage aims to classify the data into the classes defined by expert annotations. Classification techniques can be roughly divided into four groups: distance-based, feature-based, deep learning (DL)-based, and ensemble approaches (Cheng et al., 2021). Distance-based methods (Lee et al., 2012; Abanda et al., 2018) classify data into the same class as the most similar signals. These methods are known as nearest neighbor approaches (Lee et al., 2012.), and several distance metrics are applied to compute neighboring samples (Abanda et al., 2018).

In contrast, the feature-based methods substitute the preprocessing stage with feature extraction. Various feature extraction methods have been proposed, such as dissimilarity-based representation (Mauceri et al., 2020), bag of features (Baydogan et al., 2013), symbolic aggregate approximation (SAX) (Lin et al., 2003), and symmetric-based feature extraction (Barua et al., 2023). These extracted features are then used as the input for classification algorithms.

In addition, the DL-based approach extends the feature-based method into an end-to-end framework. Several types of architectures have been developed, such as Conv1D (one-dimensional convolution) to capture local sequence patterns (Sánchez-Reolid et al., 2022) and RNN (recurrent neural network) and LSTM (long short-term memory) to capture temporal information (Hüsken and Stagge, 2003; Parija et al., 2021). Moreover, residual networks and attention have been applied to improve classification accuracy (Zhu et al., 2021).

On the other hand, ensemble approaches involve fusions between multiple features (Cheng et al., 2021; Bai et al., 2021), classifiers (Hussain and Mian Qaisar, 2022), or other existing techniques. These studies achieve high accuracy and have been implemented in applications such as biometric signals (Hussain and Mian Qaisar, 2022; Singer et al., 2021), sound classification (Zhang et al., 2021), and power quality disturbance analysis (Wang et al., 2022).

Since the classification stage involves supervised learning, it experiences challenges due to problems in the training dataset, such as data imbalance, outliers, and unseen classes.

2.2. One-class classification

OCC is a machine learning task where the training data have only one seen class. The goal of OCC is to classify the data into the seen class or other unseen classes. Several OCC algorithms have been proposed. The common framework includes the score functions and threshold values. The main challenge is to define a score function, which can be a likelihood for a seen class (seen score) or a distance metric from a seen class (unseen score). Moreover, OCC is evaluated by the area under the curve (AUC) score. Such a metric is computed by considering all possible thresholds between seen and unseen classes. The AUC score corresponds to the accuracy where the best threshold value is selected (how to select

Table 1
OCC algorithms.

Strategy	Data types		
	Feature	Image	Time-series
Shallow methods	OCSVM (Schölkopf et al., 2001) LOF (Breunig et al., 2000) IF (Liu et al., 2008) OC1-NN (Khan et al., 2008) ALP (Lenz et al., 2021)	Combined with feature extraction.	
Feature extraction	**None**	AE (Cao et al., 2019) Ruff et al. (2018)	Mauceri et al. (2020) Chang et al. (2021) **OCLCB (Proposal)**
Fake unseen	Aguilar et al., 2021; Kang, 2022	GAN (Yang et al., 2019) OE (Hendrycks et al. 2018)	Zhu et al. (2022)
Self-supervision	AE (Hawkins et al., 2002) FSP (Hayashi and Fujita, 2022)	Combined with feature extraction. AE (Hawkins et al., 2002) GEO (Golan and El-Yaniv, 2018) OCITN (Hayashi et al., 2021)	Blázquez-García et al. (2021) OCSTN (Hayashi et al., 2022)

such a value is not considered.

Table 1 summarizes the OCC methods in three data types (feature, image, and time-series) and four strategies (shallow method, feature extraction, fake unseen, and self-supervision).

Shallow methods are traditional algorithms for feature data that were proposed in the 2000 s. The one-class support vector machine (OCSVM) (Schölkopf et al., 2001) considers the mapping function in the feature vector space and computes the hyperplane between the mapped data and the origin in the vector space. In contrast, the local outlier factor (LOF) (Breunig et al., 2000) considers local density for original and neighbor samples, where outliers are supposed to have different local density from those neighbors. In addition, isolation forest (Liu et al., 2008) creates a random tree structure and classifies isolated samples in the tree into unseen classes. Moreover, one-class 1-nearest neighbor (Khan and Ahmad, 2018) computes unseen scores by a distance metric from the neighbor samples; prototypes, the representative samples, are used to reduce the computation cost. These methods are still competitive on feature data. Recently, Lenz et al. (2021) proposed average localized proximity (ALP) to improve the localization process of LOF. ALP shows the best AUC score for feature data.

The feature extraction strategy obtains feature vectors from images or time series. Extracted feature vectors are then classified by shallow OCC methods or other OCC algorithms designed for feature data. Autoencoder (AE) (Cao et al., 2019) is a common method for feature extraction. In addition, Ruff et al. (2018) combined DL-based feature extraction with support vector data description (SVDD) in the same way as in OCSVM.

The fake unseen strategy creates pseudo samples for unseen classes and applies supervised classification between a real seen class and fake unseen classes. This strategy was proposed for image data, where methods such as Generative Adversarial Net (GAN) are applied to generate fake unseen data (Yang et al., 2019). In addition, outlier exposure (OE) (Hendrycks et al., 2018) imports other datasets as fake unseen classes. The fake unseen strategy is reimported to feature data; the articles (Aguilar et al., 2021; Kang, 2022) treat the subpart of the training data as an unseen class. However, a fake sample is created without information about the actual unseen classes. This causes errors due to the difference between the fake unseen and actual unseen samples.

The self-supervised approach trains the model for subtasks and classifies between seen and unseen classes by model error. The model learns from only data belonging to a seen class. Therefore, model errors for seen data are smaller compared to the data belonging to unseen classes. AE is a common subtask, where reconstruction error is a model error (Hawkins et al., 2002). This architecture supplies a fast-processing speed but has limitations in the AUC score. Golan and El-Yaniv (2018) proposed classification subtasks into 72 geometric transformations (GEO), which shows the best AUC score but has limitations in processing speed. The one-class image transformation network (OCITN) (Hayashi et al., 2021) considers the image transformation subtask to the one defined output image, namely, the goal image. Such a method shows a fair AUC score and fast processing speed. In addition, a self-supervised strategy is imported for feature data. Hayashi and Fujita (2022) proposed feature slide prediction (FSP) subtasks, where self-labeled data are created by feature slides.

In addition, this study considers the OCTSC algorithms described in the next section.

2.3. One-class time-series classification

OCTSC algorithms are categorized into feature extraction (Mauceri et al., 2020; Chang et al., 2021), fake unseen (Zhu et al., 2022), and self-supervision methods (Blázquez-García et al., 2021). This categorization is the same as for image data.

For feature extraction, Mauceri et al. (2020) combined dissimilarity-based feature representation with prototypes. This representation is then combined with a one-class 1-nearest neighbor classifier. Their experiments are performed with 12 distance metrics and 8 prototypes. While the method is simple and fast, the accuracy is not higher than that of other methods (Hayashi et al., 2022).

Chang et al. (2021) achieved better performance than Mauceri et al. (2020) in semiconductor manufacturing by incorporating expert knowledge into the signal segmentation process. However, the use of expert knowledge is a potential weakness, as it may not be easily generalized to other domains.

Zhu et al. (2022) introduced the fake unseen approach for time-series data by applying adversarial training, where adversarial samples are considered an unseen class. However, fake unseen samples might be different from actual ones because these fakes are not created from real unseen classes.

Blázquez-García et al. (2021) proposed the signal multiplication subtask for water leak detection, which achieved the best AUC score in the comparison (Hayashi et al., 2022). Nevertheless, the method has disadvantages in terms of processing speed, which is critical for processing a large dataset, such as sliding windows. It should be noted that the experiment did not finish even after a week of computation.

Hayashi et al. (2022) proposed a one-class signal transformation network (OCSTN). The algorithm considers the signal transformation network (STN) to transform seen signals into a goal signal. The unseen score is a distance metric between a goal signal and model output. The core hypothesis of this algorithm is that the model error for the seen class is smaller compared to the unseen classes because the STN model trains on data from only the seen class. However, the method is related to the STN parameters and goal signal, and the result may degrade when training data have diversity, such as sliding windows.

One common weak point in existing OCTSC algorithms is their inability to determine the appropriate signal length for analysis. Discovering the best length is challenging because existing OCTSC methods require model retraining and parameter tuning for signals of different lengths. To address this weakness, the proposed OCLCB applies a clustering algorithm to sliding windows of the signal. This eliminates the need for model retraining for processing different lengths of signals. More specifically, LCB is computed as the feature vector in the local area of a signal, and OCLCB is categorized into feature extraction methods in

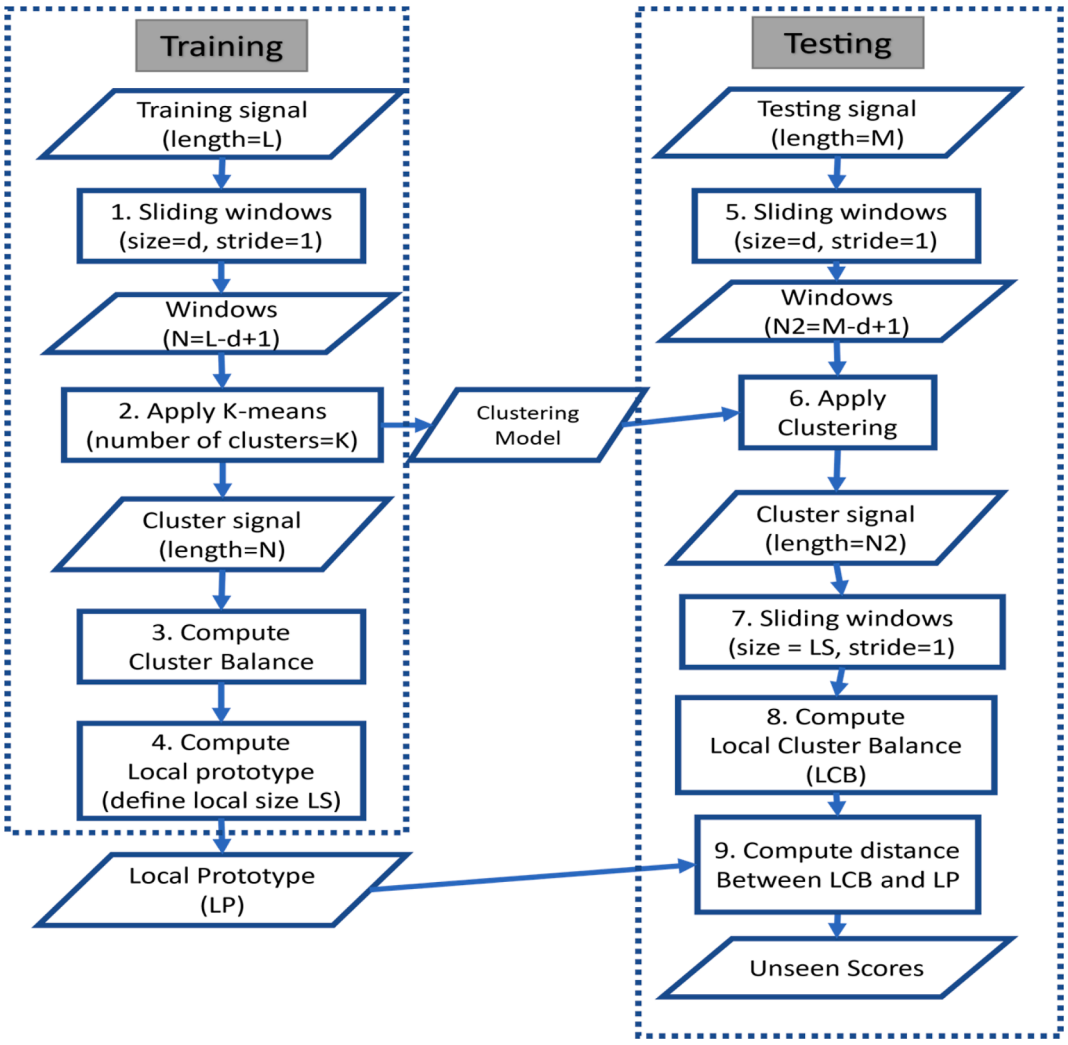


Fig. 1. OCLCB framework.

Table 1.

2.4. Clustering algorithms and cluster balance

Clustering is an unsupervised learning technique that does not rely on annotation. The goal of clustering is to divide data into groups (clusters) based on a hypothesis. Several clustering algorithms have been proposed, including K-means, Gaussian Mixture Model, and DBSCAN (Pedregosa et al., 2011). These algorithms can be categorized into two groups based on whether or not they create models. The proposed OCLCB requires a clustering model because the same clustering must be applied to both training and testing signals.

The idea is inspired by the class imbalance problem. Apart from class imbalance, cluster imbalance is not an important problem, and only a few studies care about such an issue when applying clustering results as pseudolabels for self-supervised learning. Cluster imbalances can be addressed in the same ways as class imbalances (Li et al., 2022). Moreover, there is a specific clustering task called balanced clustering (Dai et al., 2022).

This study considers cluster balance for feature extraction. Here, cluster imbalance is not considered a problem because the goal is to compute the distance between seen and unseen classes.

2.5. The proposed method

This study presents a novel OCTSC algorithm, namely, OCLCB. The proposed method applies a clustering algorithm to sliding windows extracted from the training signal and computes the cluster balance from the clustering result. The cluster balance corresponds to the balance of shapes in the signal. The hypothesis lies in the dissimilarity of the cluster balance for seen and unseen classes. Consequently, the unseen score could be computed as a distance metric between cluster balances for training and testing data. Here, the distance should be computed between the same length of signals. For this purpose, local cluster balance (LCB) is defined as the cluster balance in the local area of the signals. Fig. 1 shows the OCLCB framework, which includes training and testing stages.

The goal of the training stage is to create a clustering model and a local prototype (LP) for the training signal. These outputs are used in the testing stage, while the training signal is the whole univariate time-series signal, which needs to be split into subsignals for analysis. For this purpose, d-dimensional sliding windows are extracted. In addition, K-means is applied to sliding windows, and the clustering result is used to calculate cluster balance. Finally, the LP is computed as a representation of the training data, where local size LS corresponds to the signal length for analysis.

The goal of the testing stage is to compute unseen scores for the local area of the signal. First, d-dimensional sliding windows are extracted,

and the cluster signal is obtained by applying the clustering model to sliding windows. Then, another sliding window is applied to cluster the signal, and the LCB is computed; LCB corresponds to the feature vector in the local area of the signal. Finally, unseen scores are computed by the distance metric between LCBs and LP.

2.6. Training stage

The training data are a univariate signal Xtr, which is given as:

$$Xtr = [Xtr_1, Xtr_2, \cdots, Xtr_T, \cdots, Xtr_L], \tag{1}$$

where Xtr_T represents a value at time T, and L is the length of Xtr and is supposed to be an arbitrarily large value. Therefore, Xtr needs to be split into subsignals for analysis.

The straightforward approach is to apply arbitrary length (defined as local size LS) sliding windows to Xtr, which requires a large amount of memory space for larger window sizes. Thus, this study instead extracts d-dimensional sliding windows Wtr from Xtr as:

$$\begin{aligned} Wtr &= [Wtr_1, \cdots, Wtr_T, \cdots, Wtr_{L-d+1}] \\ &= [(Xtr_1, \cdots Xtr_d), \cdots, (Xtr_T, \cdots Xtr_{d+T-1}), \cdots, (Xtr_{L-d+1}, \cdots Xtr_L)], \end{aligned} \tag{2}$$

where d is the window size, which is smaller than LS (d>=2); therefore, the memory space can be smaller, and the number of sliding windows is L-d + 1. Then, clustering is considered for sliding windows. Let f be a clustering model that aims to assign sliding window Wtr_T into an arbitrary cluster C_i:

$$f : Wtr_T \rightarrow C_i, \text{ where } C_i \in C$$

$$C = \{C_1, C_2, \cdots, C_i, \cdots, C_K\} \tag{3}$$

where C is the set of clusters and K indicates the number of clusters. This study applies k-means (MacQueen, 1967) to obtain the clustering model, as shown in **Algorithm**.

Algorithm. Obtaining the K-means model from sliding windows

Input: d-dimensional sliding windows Wtr, number of clusters K, $\varepsilon > 0$
Output: clustering model f, centroids μ
1. Prepare k centroids:
$\mu = \{\mu_1, \mu_2, \cdots, \mu_K\}$, where μ is d-dimensional data.
$\mu_{old} = \mu$ //μ_{old} is used in step 4 to check the convergence.
2. Assign data to the cluster that has the nearest centroid.
f: $Wtr_T \rightarrow C_i$.
$i = argmin(d(Wtr_T, \mu_j)), forj = 1 to K$.
3. Update centroids that are the average of data in clusters.
$\mu_j = \frac{(\sum \text{Wtr})}{|C_j|}, where Wtr \in C_j, forj = 1, \cdots, K.$
4. Repeat steps 2 and 3 until convergence.
//Convergence is guaranteed based on the centroids' change.
If $\sum_{j=1}^{K} d(\mu_j, \mu_{oldj}) \leq \varepsilon$:
Convergence.
else:
$\mu_{old} = .\mu$
Return to step 2.
5. Take model f and the centroids μ.

In the previous algorithm, ε is threshold value to determine the convergence.

The clustering signal Ctr is obtained by applying a clustering model:

$$\begin{aligned} Ctr &= [Ctr_1, Ctr_2, \cdots, Ctr_T, \cdots, Ctr_{L-d+1}] \\ &= [f(Wtr_1), f(Wtr_2), \cdots, f(Wtr_T), \cdots, f(Wtr_{L-d+1})]. \end{aligned} \tag{4}$$

Cluster Balance CBtr is a k-dimensional vector that is calculated from the clustering result by:

$$CBtr = (|Ctr_T = C_1|, |Ctr_T = C_2|, \cdots, |Ctr_T = C_K|). \tag{5}$$

Cluster balance in a local area is called LCB, which is not computed in the training stage to reduce memory space and processing time. Instead, LP is computed as a representative vector for LCBs in training data as follows:

$$LP = \text{LS}^* \frac{CBtr}{L - d + 1}, \tag{6}$$

where LS is defined as the local size (the size of the local area in the signal).

2.7. Testing stage

The testing signal Xte is given as follows:

$$Xte = [Xte_1, Xte_2, \cdots, Xte_T, \cdots, Xte_M], \tag{7}$$

where M is the length of the testing signal. The proposed method computes the feature vector of subsignals in Xte by applying Equations (8)-(10) instead of splitting Xte into subsignals by sliding windows (with lengths of LS), which can be memory intensive.

Sliding windows Wte are extracted from Xte as:

$$\begin{aligned} Wte &= [Wte_1, \cdots, Wte_T, \cdots, Wte_{M-d+1}] \\ &= [(Xte_1, \cdots Xtr_d), \cdots, (Xte_T, \cdots Xte_{d+T-1}), \cdots, (Xte_{M-d+1}, \cdots Xtr_M)], \end{aligned} \tag{8}$$

where the window size is d, which is the same as the training data. The cluster signal Cte is computed by applying the clustering model f to Wte by the following equation:

$$\begin{aligned} Cte &= [Cte_1, Cte_2, \cdots, Cte_T, \cdots, Cte_{M-d+1}] \\ &= [f(Wte_1), f(Wte_2), \cdots, f(Wte_T), \cdots, f(Wte_{M-d+1})]. \end{aligned} \tag{9}$$

Then, local clusters are sliding windows from Cte as given:

$$\begin{aligned} LCte &= [LCte_1, LCte_2, \cdots, LCte_{M-LS+1}] \\ &= [(Cte_1, \cdots Cte_{LS}), \cdots, (Cte_n, \cdots Cte_{n+LS-1}), \cdots, (Cte_{M-LS+1}, \cdots Vte_M)]. \end{aligned} \tag{10}$$

LCB is the cluster balance for the local area that is computed in each sliding window in *LCte*:

$$\begin{aligned} LCBs &= [LCB_1, LCB_2, \cdots, LCB_n, \cdots, LCB_{M-LS+1}] LCB_n \\ &= (|C_1 in LCte_n|, |C_2 in LCte_n|, \cdots, |C_K in LCte_n|) \end{aligned} \tag{11}$$

where LCB_n is a K-dimensional feature vector to represent the local area $LCte_n$.

Equations (10) and (11) require considerable memory and processing time, so the cluster balance is only computed for the first window to reduce processing time. The balances for the other windows are calculated based on the previous window by decreasing the count of the first cluster and increasing the count of the last cluster by one.

Finally, the Unseen Score is computed as a distance metric between *LCB* and *LP*:

$$\text{Unseen Score} = \text{dist}(LCB_n, LP) = \sum_{i=1}^{K} |LCB_{ni} - LP_i|, \tag{12}$$

where i is the dimension of LCB_n and *LP*.

3. Experiment

3.1. Experimental setting

The main advantage of OCLCB is that it does not require model retraining for different signal lengths, which is evident without an experiment. However, the main question is whether the performance of OCLCB is acceptable in terms of accuracy and processing speed. For this purpose, OCLCB is evaluated on two tasks, authentication in subsection 4.2 and change detection in subsection 4.3. Additionally, the source

Table 2
Comparison of AUC scores for the authentication dataset.

Seen user	Mauceri et al. (2020), Dissimilarity + One-class 1-Nearest Neighbors, kmeans-based prototype							Blázquez-García et al. (2021)		OCSTN (Hayashi et al., 2022)		**OCLCB**
	KL	CB	Cos	Euc	Gauss	Sig	WS	Acc	Prob Acc	1 STN	5 STN	
1	86.7 ± 0.5	68.9 ± 21.5	68.6 ± 17.6	74.0 ± 17.9	77.4 ± 17.3	79.2 ± 16.3	80.9 ± 15.6	84.8 ± 0.3	91.4 ± 0.6	88.5 ± 3.3	90.6 ± 0.8	**91.7 ± 0.6**
2	77. 4 ± 0.8	75.3 ± 1.0	73.1 ± 3.2	73.6 ± 3.1	73.9 ± 3.0	74.5 ± 3.0	75.4 ± 3.6	72.1 ± 0.3	**84.4 ± 0.7**	81.0 ± 2.7	83.8 ± 0.7	80.0 ± 0.5
3	68.5 ± 0.7	62.0 ± 4.4	59.5 ± 5.1	61.3 ± 5.4	62.4 ± 5.4	63.5 ± 5.5	64.9 ± 6.1	67.4 ± 0.2	**81.8 ± 0.3**	69.7 ± 6.9	75.9 ± 4.3	72.0 ± 0.5
4	77.5 ± 0.6	55.2 ± 32.5	60.7 ± 27.9	67.4 ± 26.8	71.5 ± 25.3	73.8 ± 23.7	75.7 ± 22.4	82.4 ± 0.4	**90.4 ± 0.6**	83.8 ± 6.3	86.6 ± 0.4	85.0 ± 0.6
5	87.0 ± 0.6	88.4 ± 4.9	88.8 ± 4.0	89.9 ± 4.0	90.8 ± 4.0	90.2 ± 3.8	90.7 ± 3.7	74.9 ± 0.3	**92.6 ± 0.2**	84.7 ± 8.4	89.1 ± 4.6	88.1 ± 0.5
6	79.3 ± 0.5	69.1 ± 14.8	69.6 ± 12.1	73.3 ± 12.3	75.7 ± 12.0	76.6 ± 11.2	77.6 ± 10.6	74.0 ± 1.2	**92.1 ± 0.3**	80.9 ± 9.3	87.3 ± 1.2	85.4 ± 0.3
7	74.3 ± 1.3	74.5 ± 10.0	75.3 ± 8.2	77.6 ± 8.1	78.9 ± 7.8	77.8 ± 7.6	78.9 ± 7.5	61.2 ± 0.7	**87.6 ± 1.7**	68.5 ± 10.8	77.2 ± 1.0	81.1 ± 1.5
8	74.2 ± 0.7	75.0 ± 7.4	76.1 ± 6.3	78.0 ± 6.4	79.1 ± 6.1	78.1 ± 6.0	**78.7 ± 5.8**	69.3 ± 0.2	77.3 ± 0.8	64.1 ± 7.3	66.6 ± 5.0	64.3 ± 2.1
9	77.5 ± 3.9	73.4 ± 0.6	70.1 ± 4.7	70.8 ± 4.3	71.3 ± 4.0	72.3 ± 4.3	73.5 ± 4.9	71.5 ± 0.3	**86.7 ± 0.4**	78.0 ± 6.4	80.8 ± 4.5	83.1 ± 0.5
10	89.8 ± 0/ 3	67.8 ± 26.4	64.1 ± 22.2	71.4 ± 23.0	75.9 ± 22.5	78.7 ± 21.5	81.1 ± 20.7	90.0 ± 0.3	95.1 ± 0.1	92.4 ± 4.9	95.1 ± 0.5	**95.6 ± 0.1**
11	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0	96.2 ± 0.2	**100 ± 0.0**	98.9 ± 3.9	99.9 ± 0.1	99.2 ± 0.2
12	79.1 ± 0.9	70.0 ± 13.1	68.3 ± 11.0	71.7 ± 11.2	74.0 ± 11.0	74.8 ± 10.2	75.9 ± 9.8	73.1 ± 1.3	**90.7 ± 0.7**	76.1 ± 10.4	82.9 ± 2.7	82.5 ± 0.9
13	74.3 ± 1.2	66.6 ± 2.3	64.1 ± 4.0	65.3 ± 4.1	66.1 ± 4.0	67.1 ± 4.3	68.7 ± 5.7	68.6 ± 0.2	**85.9 ± 0.4**	73.0 ± 5.6	75.3 ± 2.8	79.9 ± 0.3
14	87.9 ± 0.5	66.4 ± 23.0	68.4 ± 19.0	73.8 ± 18.9	76.9 ± 18.0	78.8 ± 17.0	80.3 ± 16.2	84.5 ± 0.4	**92.7 ± 0.4**	86.7 ± 2.7	88.9 ± 0.9	89.1 ± 0.5
15	72.4 ± 1.0	73.2 ± 3.6	72.3 ± 3.2	73.5 ± 3.6	74.5 ± 3.8	75.2 ± 3.7	76.4 ± 4.6	69.7 ± 0.6	**90.3 ± 0.6**	82.2 ± 5.8	85.8 ± 1.4	84.7 ± 0.8
16	63.0 ± 0.4	67.0 ± 4.3	64.2 ± 5.4	65.9 ± 5.6	66.9 ± 5.4	66.0 ± 5.4	66.6 ± 5.2	58.9 ± 0.8	**80.7 ± 0.8**	63.4 ± 1.7	64.4 ± 0.4	62.4 ± 1.0
17	84.4 ± 1.0	61.9 ± 30.5	69.3 ± 27.0	74.9 ± 25.3	78.5 ± 23.8	80.7 ± 22.3	82.4 ± 21.0	87.5 ± 0.2	**95.7 ± 0.4**	93.8 ± 2.5	94.9 ± 0.7	90.4 ± 0.4
18	68.1 ± 0.9	69.3 ± 6.0	63.8 ± 9.1	66.8 ± 9.5	68.6 ± 9.2	68.8 ± 8.4	69.9 ± 8.2	66.6 ± 0.2	**80.8 ± 0.8**	72.5 ± 1.6	73.7 ± 0.7	73.9 ± 0.7
19	72.3 ± 1.0	78.4 ± 3.5	75.4 ± 5.0	75.3 ± 4.4	75.3 ± 4.0	74.9 ± 3.8	75.5 ± 3.8	70.7 ± 0.5	**91.8 ± 0.4**	72.2 ± 3.1	73.4 ± 2.0	83.0 ± 0.8
20	81.4 ± 1.5	73.9 ± 11.0	68.0 ± 12.2	71.4 ± 12.0	73.5 ± 11.6	73.8 ± 10.6	75.6 ± 10.8	68.7 ± 0.4	**83.1 ± 0.7**	77.2 ± 6.7	79.3 ± 4.3	82.3 ± 0.9
Average	78.7	71.8	71.0	73.8	75.6	76.2	77.4	74.6	**88.6**	79.4	82.6	82.7

code is uploaded at https://github.com/ToshiHayashi/OCLCB.

Experiments were conducted on one synthetic dataset and two real-world datasets. The synthetic dataset was created by combining sine waves, while the two real-world datasets included ballistocardiography (BCG) signals. BCG signals measure the recoil forces of the body in reaction to the cardiac ejection of blood into the vasculature (Cimr et al., 2020. Cimr et al., 2021).

3.2. Authentication task

The goal of the authentication task is to classify signals as either the target person or not. The input signals are taken from the target person, which corresponds to the OCC. The experiment uses a **sleeping dataset** (Studnicka, 2022a), which has BCG signals collected from twenty people and annotated with the ID of signal holders.

The proposed OCLCB is compared with existing OCTSC algorithms in terms of the AUC scores and processing time, namely, Mauceri et al. (2020), Blázquez-García et al. (2021), and OCSTN (Hayashi et al., 2022). Preprocessing is performed by following Hayashi et al. (2022), where the original signals are split into subsignals with a length of 660 (without sliding windows). The data are then randomly split into training and testing sets at a ratio of 80:20 by five different random seeds. The source code of the comparative methods is uploaded at https://github.com/ToshiHayashi/OCSTN.

Mauceri et al. (2020) combined dissimilarity-based feature extraction and a one-class 1-nearest neighbor classifier. In their original paper, they conducted experiments with 12 distance metrics and 8 prototypes. This comparison includes a k-means-based prototype and seven distance metrics, Kullback Leibler Divergence (KL), City block (CB), Cosine (Cos), Euclidean (Euc), Gaussian, Sigmoid (Sig), and Wasserstein (WS), because trying all combinations is extensive, and some dissimilarity metrics require much computation time.

Blázquez-García et al. (2021) proposed a self-supervised approach using a classification subtask for signal multiplication, which uses four self-labels. Self-labeled signals are created as 1, 1.7, 2.4, and 3.1 times the original signal. The subtask is evaluated by two score functions: accuracy (Acc) and probability corresponding to correct prediction (prob acc).

OCSTN (Hayashi et al., 2022**)** considered a signal transformation subtask into a goal signal defined as the average in the training signals. The results are obtained by two unseen scores: the model errors from a single STN (1 STN) and the summation of five errors (5 STN).

The parameters are set with **OCLCB** as the proposed method, the window size for clustering as 2, and the number of clusters as 25. It is important to note that the experimental process differs from Fig. 1 to adjust OCLCB, as local size is not considered because the signal is already split into subsignals, and cluster balances are computed for each subsignal. Then, the prototype is calculated as the average of the cluster balances in the training signals.

Table 2 compares the AUC score for the authentication dataset. Each

Table 3
Comparison of processing time.

Method	Training time (second)	Testing time (second)
Mauceri et al. (2020)	1.39	0.93
Blázquez-García et al. (2021)	840	1440
OCSTN (1 STN) (Hayashi et al., 2022)	276	3.46
OCSTN (5 STNs) (Hayashi et al., 2022)	1178	16.8
OCLCB	16.3	62.0

cell shows the AUC scores corresponding to OCC algorithms and seen users who are the target person for authentication. OCLCB shows the best AUC scores for two seen classes and the second-best on average. Therefore, OCLCB can be considered an alternative solution for OCTSC. The method proposed by Blázquez-García et al. achieves the highest average AUC score.

In addition, Table 3 compares the processing time in a non-GPU environment using Intel® Core™ i9-9900 K CPU @ 3.60 GHz, and RAM 64 GB. Training and testing times are calculated separately, and the number of training and testing signals is the same as Hayashi et al. (2022), consisting of 1600 and 7400 signals, respectively. OCLCB achieves a fast training time among the evaluated methods, which is advantageous for parameter tuning. However, its testing time is increased

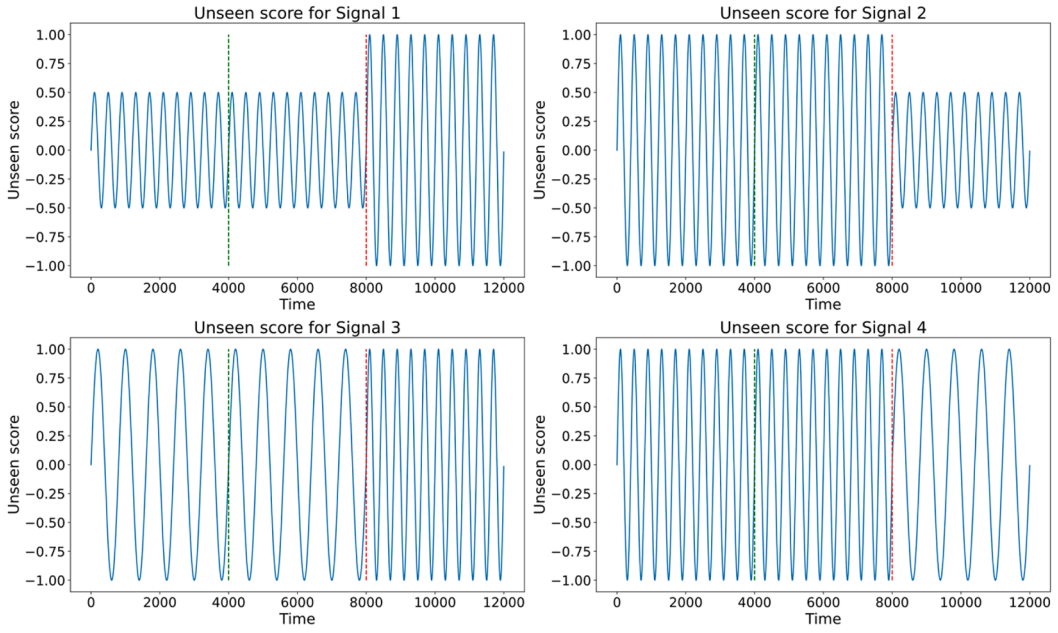


Fig. 2. Synthetic signals for the experiment.

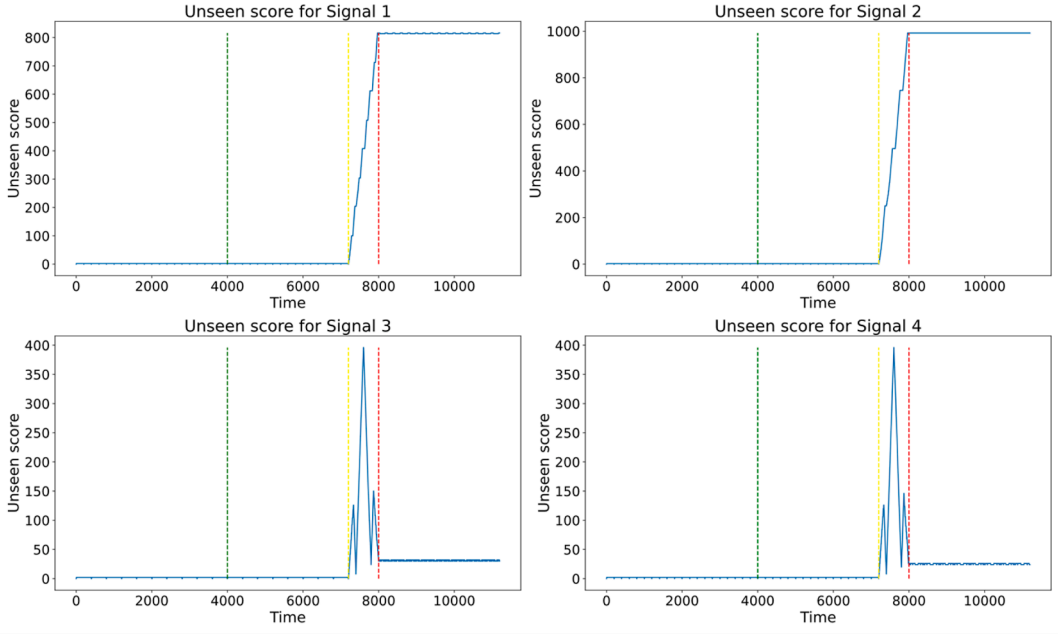


Fig. 3. Unseen scores obtained by OCLCB.

Table 4
Comparison of AUC scores for synthetic signals.

Algorithms	Signal 1	Signal 2	Signal 3	Signal 4
Mauceri et al. (2020)	**99.94 ± 0.00**	**99.89 ± 0.00**	**99.89 ± 0.00**	99.89 ± 0.00
Blázquez-García et al. (2021)	99.69 ± 0.24	99.49 ± 0.39	99.76 ± 0.12	99.38 ± 0.68
OCSTN (Hayashi et al., 2022)	77.81 ± 9.08	33.54 ± 4.69	59.70 ± 9.63	51.35 ± 3.64
OCLCB	99.89 ± 0.04	99.86 ± 0.03	99.80 ± 0.05	**99.90 ± 0.04**

Table 5
Measuring schedule.

Time (s)	Event
0	start of measuring on back
60	breath-holding during inhalation (30 s)
120	breath-holding during inhalation (30 s)
180	breath-holding during exhalation (30 s)
240	breath-holding during exhalation (30 s)
300	underlay of legs for position change
420	turning on the side.
480	breath-holding during inhalation (30 s)
540	breath-holding during inhalation (30 s)
600	breath-holding during exhalation (30 s)
660	breath-holding during exhalation (30 s)
720	end of measuring

due to the need to extract sliding windows and apply a clustering algorithm.

According to Table 2 and Table 3, the AUC score and processing time are trade-off practices. The method from Blázquez-García et al. has the highest AUC score but the lowest processing speed, while Mauceri et al. has the fastest processing time but the lowest AUC. Moreover, OCSTN and OCLCB provide fair results in the trade-off of AUC score and processing speed.

Since the comparison uses a signal length of 660, the main advantage of OCLCB—not requiring model retraining for different signal lengths—is not utilized. Nevertheless, OCLCB shows fair performance in terms of AUC scores and processing speed. These aspects support the idea that OCLCB can perform well with minimal drawbacks.

3.3. Change detection task

Change detection is evaluated using two datasets: a synthetic dataset and a breathing dataset. **A synthetic dataset** is created by connecting two different sine waves. Fig. 2 shows four synthetic signals. Signals 1 and 2 change amplitudes (0.5 and 1.0), while signals 3 and 4 change the period (400 and 800). All synthetic signals have 12,000 values, and changes exist in a connected point (the index is 8000). Training data are shown before the green line (the first 4000 values).

Fig. 3 visualizes the unseen scores obtained by OCLCB, where signals are preprocessed into subsignals by sliding windows (local size = 800). The yellow line indicates the start of the change, while the red line shows the end of the change. The sliding windows between these lines include the parts for both before and after the change. OCLCB detected the synthetic changes for all signals. The amplitude changes (Signals 1 and 2) provided the highest unseen scores after the change, while the cycle changes (Signals 3 and 4) offered the highest unseen scores around the change. It is important to note that the results might differ when the signal has different shapes or change ratios.

Table 4 provides the AUC comparison with other OCTSC algorithms. The signals are preprocessed into subsignals by sliding windows (window size = 800) to apply these algorithms. The compared methods include Mauceri et al. (KL), Blázquez-García et al. (Prob Acc), OCSTN (1 STN), and OCLCB.

OCLCB shows a competitive AUC score for detecting the change in synthetic signals. The method from Mauceri et al. achieved the best AUC, while Blázquez-García et al. showed a high AUC as well. It is important to note that Table 2 provides more substantial evidence than Table 4 for AUC comparison because the dataset is an actual signal. On the other hand, Table 4 suggests a fundamental problem for OCSTN, which is that it requires parameter tuning to improve the AUC. Such a process requires much retraining of the models.

The breathing dataset (Studnicka, 2022b) is a real-world dataset that includes BCG signals collected from tested individuals who varied their breathing behavior according to the measuring schedule (Cimr et al., 2020) (see Table 5).

The training data consist of the beginning of the signal (the first 50 s) because these parts do not include the breathing event. OCLCB is compared for change detection with the methods from Mauceri et al. (KL), Blázquez-García et al., and OCSTN.

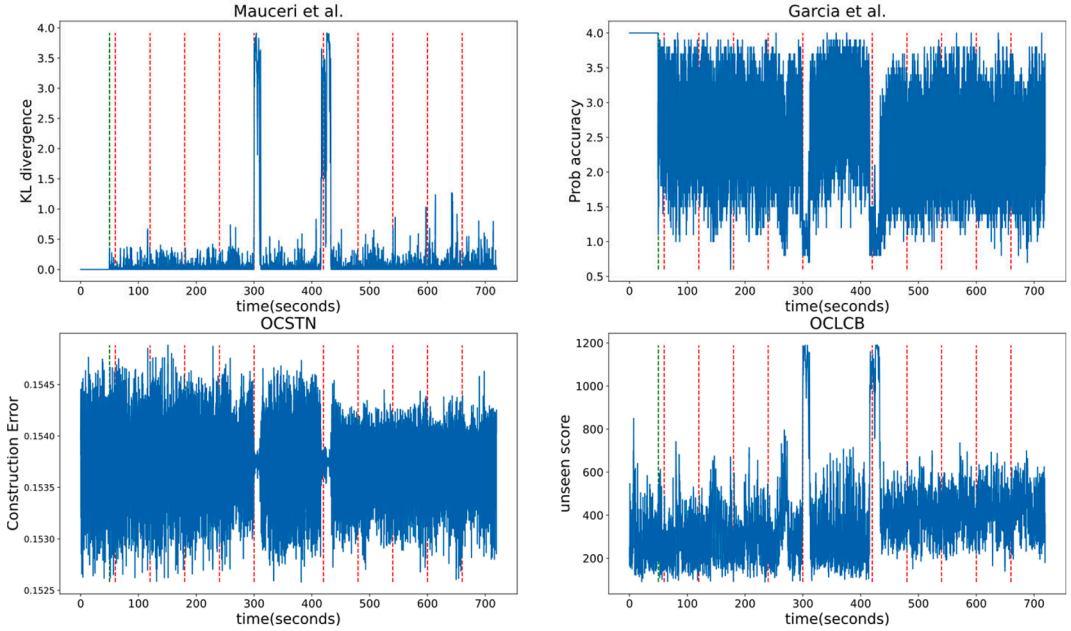


Fig. 4. Comparison of change detection for the breathing dataset (signal length = 660).

Table 6
Experimental parameters.

Parameter	Values
Window size for clustering	2, 3, 4, 5, 6, 7, 8, 9, 10
Number of clusters	2, 3, 5, 10, 15, 20, 25, 50
Local size	50, 100, 500, 660, 1000, 2000, 5000, 10,000

The evaluation is made with visual analysis based on the following hypotheses:

- Unseen score increases (seen score decreases) around breathing events that are marked by red lines.
- The unseen score increases (seen score decreases) after the change in body position.

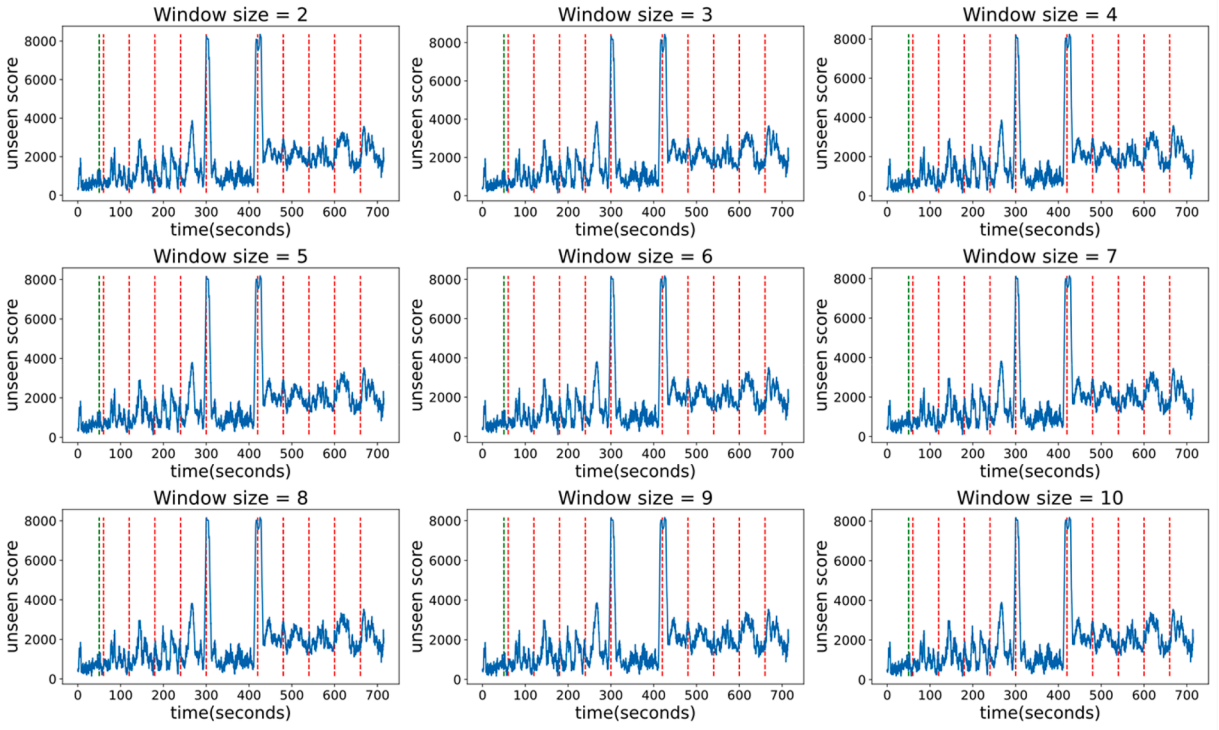


Fig. 5. Comparison of window sizes (breathing dataset).

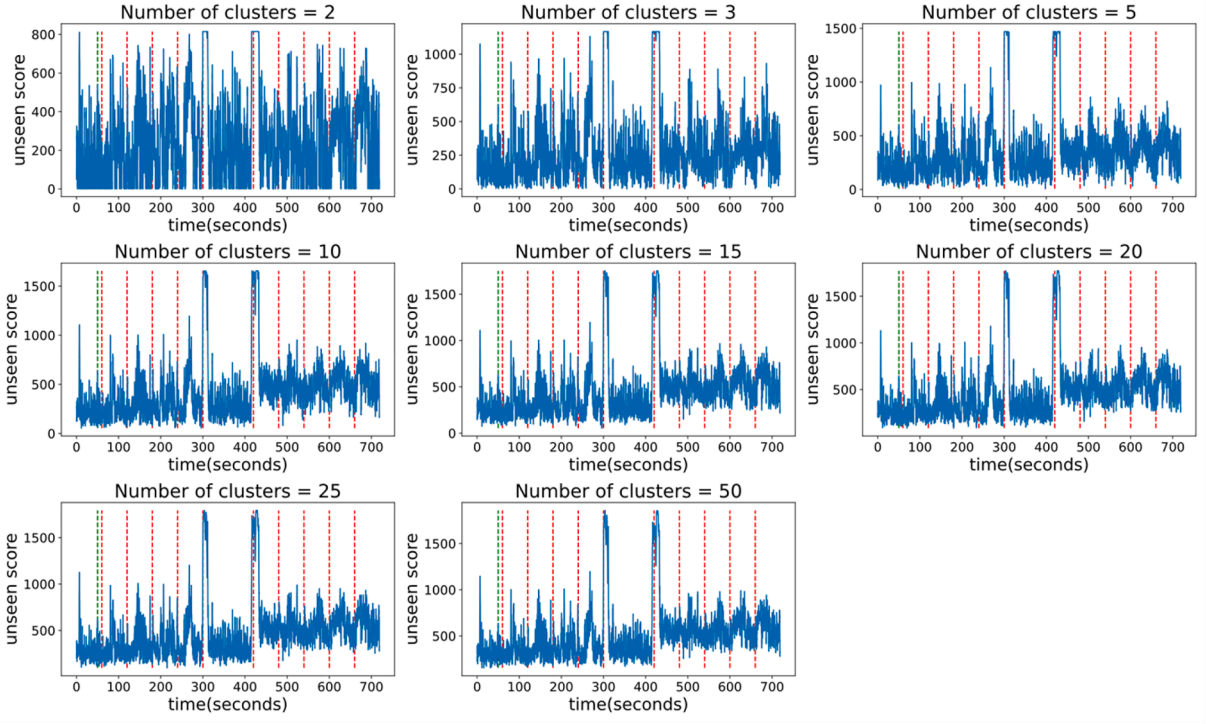


Fig. 6. Comparison of the number of clusters (breathing dataset).

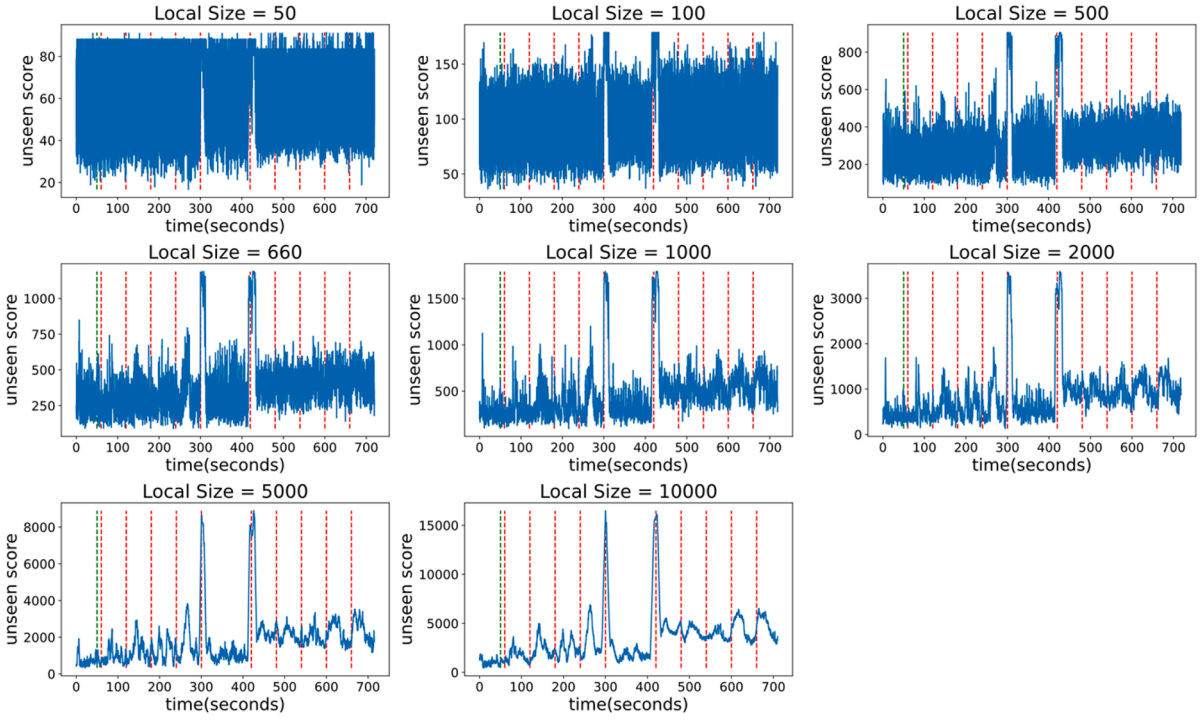


Fig. 7. Comparison of local sizes (breathing dataset).

Fig. 4 visualizes seen scores (for Blázquez-García et al.) and unseen scores (for the rest of the methods) for four OCSTN algorithms. Mauceri et al., Blázquez-García et al., and OCLCB increased unseen scores (or decreased seen scores) around body position changes. However, OCSTN did not show high unseen scores around the change and exhibited low variance, which can be problematic in determining the threshold value. On the other hand, the methods from Mauceri et al. and Blázquez-García et al. overfit to training parts, while OCLCB did not exhibit this issue because the model is trained from sliding windows.

However, the visual analysis does not show clear differences in the unseen scores for changes in breathing behavior, possibly due to the small window size used. Further experiments should be performed with different signal lengths. Nevertheless, other OCTSC algorithms require model retraining and parameter tuning for different lengths of signals, while OCLCB does not have this issue because the model is created from few-dimensional sliding windows.

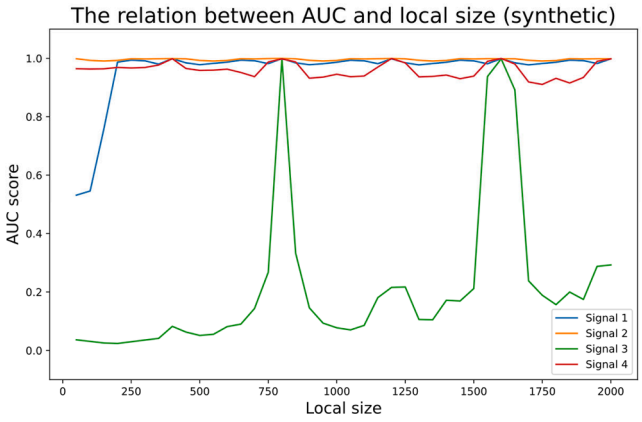


Fig. 8. Comparison of local sizes (synthetic dataset).

4. Discussion

4.1. Parameter analysis

OCLCB has three parameters: window size for clustering, number of clusters, and local size. This study performed the grid search for the parameters provided in Table 6.

Fig. 5 compares the window size for clustering. In the comparison, all results appeared to be the same; window size does not change the results. Therefore, the size should be two because a smaller window size gives faster processing speed.

On the other hand, Fig. 6 compares different numbers of clusters. Small values (2, 3, and 5) degrade the performance. In these cases, the number of clusters is smaller than the number of shape types in the signal, and clusters cannot cover all shape types.

Fig. 7 compares the local size parameter, which has the most significant differences in these three parameters. A larger local size is better because it reduces the noise in unseen scores. On the other hand, a larger size increases testing time in the actual application because the algorithm needs to collect extensive information. This is a trade-off that should be decided based on the purpose of the application. For example, a long testing time is acceptable for analyzing changes with no urgency, while a short testing time is needed to quickly detect an emergency.

Fig. 8 illustrates the relationship between AUC and local size using synthetic signals. The AUC is computed for every 50 local sizes between 50 and 2000. It is important to note that the local size should be larger than the cycle of the trained signal, where signal 3 starts with a period of 800, and the other signals start with 400. This ensures that the clusters cover all shapes in the cycle.

OCLCB has difficulty detecting Signal 3, which includes a period change from a large cycle to a small cycle. The local size needs to be equal to the period or its multiples. Otherwise, the AUC scores are degraded. In the actual signal, the period is unknown, but the larger local sizes have a greater chance of being multiples of the period.

According to Fig. 5-Fig. 8, the sliding window size should be 2, and the number of clusters should be more than 10. Moreover, the local size should be a multiple of the period; increasing the local size is a simple

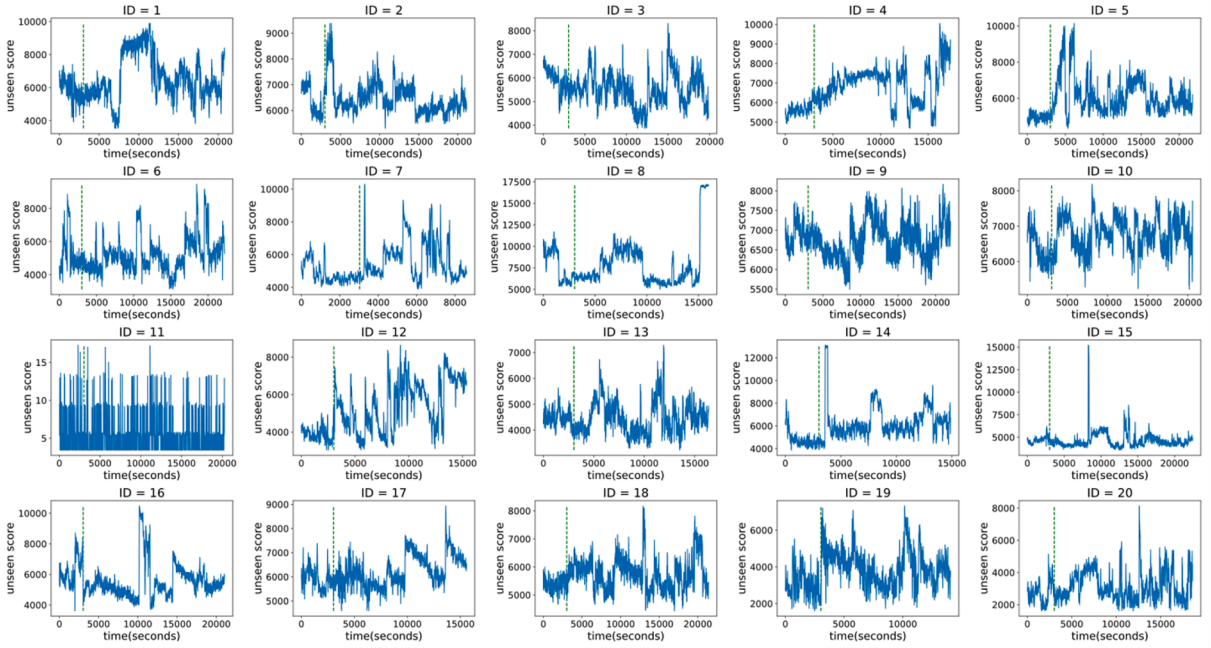


Fig. 9. For change detection from sleeping people, the x-axis and y-axis are the time (sec) and an unseen score, respectively.

way to achieve such requirements for real data.

4.2. Change detection in authentication dataset

This section applies OCLCB to change detection for the authentication dataset. Evaluation is not possible because no information is given about the changes, but the result will support the authentication task. Data are sliding windows extracted from the signal. The training data consist of the parts before the green line that have one million values at 330 Hz. Therefore, training data are values of approximately 3000 s.

Fig. 9 visualizes the unseen scores. Changes are detected in 19 signals, while ID 11 shows consistent unseen scores with a maximum value of approximately 20, which can provide a high AUC score in OCTSC.

The authentication dataset is collected from sleeping individuals. Therefore, the algorithms detected the changes during sleeping. Two possible explanations for the changes are considered:

- Changes are related to the cycle of sleep.
- Individuals change their body positions during sleep.

If the algorithm were able to detect sleep cycles, it would be an interesting finding for further study. However, these changes would degrade the AUC score on the authentication task. This limitation highlights the need for consistent signals in the authentication task, which could be addressed in future work.

4.3. Advantages and limitations

The main advantage of OCLCB is its robustness to changes in signal length, while other OCC methods need model retraining to classify different signal lengths, which reduces the parameter tuning cost. Additionally, OCLCB has the advantage of processing large-length sliding windows, as the method can represent a large window as a K-dimensional vector. In contrast, the approach from Blázquez-García et al. takes time to process large data as the algorithm creates self-labeled data for each window. OCSTN has problems processing sliding windows, and the method from Mauceri et al. has memory issues when sliding windows have LS-length.

On the other hand, OCLCB has limitations in detecting signal cycle changes from a large value to a small value. In these cases, OCLCB must use local sizes equal to the period or its multiples. However, estimating the period is not easy in real world data. Another limitation is deciding the threshold value because the OCC algorithm cannot access unseen classes during the training stage. The threshold can be selected, such as "the value larger than N% of unseen scores in training data". However, confidence is only given with access to actual unseen classes.

5. Conclusion and future work

This paper proposes an OCLCB algorithm that extracts feature vectors by cluster balance of sliding windows. The unseen score is computed by the distance metric between the training and testing data. The main advantage of the proposed OCLCB algorithm is that the model does not require retraining to process signals of different lengths, which supports parameter tuning.

Further study is needed to improve the AUC score of the model. The main challenge is how to determine the appropriate local size. Moreover, exploring alternative distance metrics and clustering algorithms, as well as increasing the number of prototypes, could improve the AUC. Furthermore, applying LCB to other types of data could avoid the need for model retraining. Another important challenge is creating more synthetic signals to address specific problem settings, such as dataset corruption, and to help identify the limitations of existing TSC and OCTSC algorithms.

CRediT authorship contribution statement

Toshitaka Hayashi: Conceptualization, Methodology, Writing – original draft, Investigation, Software, Visualization. **Dalibor Cimr:** Writing – original draft, Investigation, Data curation. **Filip Studnička:** Investigation, Data curation. **Hamido Fujita:** Conceptualization, Writing – review & editing, Supervision. **Damián Bušovský:** Investigation, Data curation. **Richard Cimler:** Project administration, Funding acquisition. **Ali Selamat:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used in the article is explained in the paper.

Acknowledgments

This publication was based on the information (data) provided by Program 4 within the research project designated (named) “Healthy Aging in Industrial Environment HAIE” with registration number CZ.02.1.01/0.0/0.0/16_019/0000798, which was funded by the European Union and provided by the Ministry of Education, Youth and Sports of the Czech Republic. The mentioned information (data) was obtained (gathered) by the Department of Human Movement Studies, The Human Motion Diagnostic Center, University of Ostrava, Ostrava, Czech Republic.”

The presented research was financially supported by the Specific research projects 2103/2020 at the Faculty of Science, University of Hradec Králové.

This study is supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research) # JP2 3 K 1 1 2 3 5.

The dataset is available at (Studnicka, 2022a. Studnicka, 2022b).

The source code is available at https://github.com/ToshiHayashi/OCLCB.

References

Abanda, A., Mori, U., & Lozano, J. A. (2018). A review on distance based time series classification. *Data Mining and Knowledge Discovery, 33*(2), 378–412. https://doi.org/10.1007/s10618-018-0596-4

Aguilar, D. L., Loyola-González, O., Medina-Pérez, M. A., Canete-Sifuentes, L., & Choo, K. R. (2021). PBC4occ: A novel contrast pattern-based classifier for one-class classification. *Future Generation Computer Systems, 125*, 71–90. https://doi.org/10.1016/j.future.2021.06.046

Alimohammadi, H., & Chen, S. (2021). Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis. *Expert Systems with Applications, 191*, Article 116371. https://doi.org/10.1016/j.eswa.2021.116371

Bai, B., Li, G., Wang, S., Wu, Z., & Yan, W. (2021). Time series classification based on multi-feature dictionary representation and ensemble learning. *Expert Systems with Applications, 169*, Article 114162. https://doi.org/10.1016/j.eswa.2020.114162

Barua, P. D., Keles, T., Dogan, S., Baygin, M., Tuncer, T., Demir, C. F., … Acharya, U. R. (2023). Automated EEG sentence classification using novel dynamic-sized binary pattern and multilevel discrete wavelet transform techniques with TSEEG database. *Biomedical Signal Processing and Control, 79*, Article 104055. https://doi.org/10.1016/j.bspc.2022.104055

Baydogan, M. G., Runger, G. C., & Tuv, E. (2013). A Bag-of-Features Framework to Classify Time Series. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(11), 2796–2802. https://doi.org/10.1109/tpami.2013.72

Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021). Water leak detection using self-supervised time series classification. *Information Sciences, 574*, 528–541. https://doi.org/10.1016/j.ins.2021.06.015

Breunig, M., Kriegel, H., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *Sigmod Record, 29*(2), 93–104. https://doi.org/10.1145/335191.335388

Cao, V. L., Nicolau, M., & McDermott, J. (2019). Learning Neural Representations for Network Anomaly Detection. *IEEE Transactions on Cybernetics, 49*(8), 3074–3087. https://doi.org/10.1109/tcyb.2018.2838668

Chang, K., Yoo, Y., & Baek, J. (2021). Anomaly Detection Using Signal Segmentation and One-Class Classification in Diffusion Process of Semiconductor Manufacturing. *Sensors, 21*(11), 3880. https://doi.org/10.3390/s21113880

Cheng, W., Suganthan, P. N., & Katuwal, R. (2021). Time series classification using diversified Ensemble Deep Random Vector Functional Link and Resnet features. *Applied Soft Computing, 112*, Article 107826. https://doi.org/10.1016/j.asoc.2021.107826

Cimr, D., Studnicka, F., Fujita, H., Tomaskova, H., Cimler, R., Kuhnova, J., & Slegr, J. (2020). Computer aided detection of breathing disorder from ballistocardiography signal using convolutional neural network. *Information Sciences, 541*, 207–217. https://doi.org/10.1016/j.ins.2020.05.051

Cimr, D., Studnicka, F., Fujita, H., Cimler, R., & Slegr, J. (2021). Application of mechanical trigger for unobtrusive detection of respiratory disorders from body recoil micro-movements. *Computer Methods and Programs in Biomedicine, 207*, Article 106149. https://doi.org/10.1016/j.cmpb.2021.106149

Dai, X., Wang, J., & Zhang, W. (2022). Balanced clustering based on collaborative neurodynamic optimization. *Knowledge Based Systems, 250*, Article 109026. https://doi.org/10.1016/j.knosys.2022.109026

D'Urso, P., De Giovanni, L., Massari, R., D'Ecclesia, R. L., & Maharaj, E. A. (2020). Cepstral-based clustering of financial time series. *Expert Systems with Applications, 161*, Article 113705. https://doi.org/10.1016/j.eswa.2020.113705

Golan, I., & El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In *Advances in neural information processing systems* (p. 31).

Hawkins, S., He, H., Williams, G. R., & Baxter, R. A. (2002). In *Outlier Detection Using Replicator Neural Networks* (pp. 170–180). Springer Science+Business Media. https://doi.org/10.1007/3-540-46145-0_17.

Hayashi, T., Fujita, H., & Hernandez-Matamoros, A. (2021). Less complexity one-class classification approach using construction error of convolutional image transformation network. *Information Sciences, 560*, 217–234. https://doi.org/10.1016/j.ins.2021.01.069

Hayashi, T., & Fujita, H. (2022). OCFSP: Self-supervised one-class classification approach using feature-slide prediction subtask for feature data. *Soft Computing, 26*(19), 10127–10149. https://doi.org/10.1007/s00500-022-07414-z

Hayashi, T., Cimr, D., Studnička, F., Fujita, H., Bušovský, D., & Cimler, R. (2022). OCSTN: One-class time-series classification approach using a signal transformation network into a goal signal. *Information Sciences, 614*, 71–86. https://doi.org/10.1016/j.ins.2022.09.027

Hendrycks, D., Mazeika, M., & Dietterich, T. G. (2018). Deep Anomaly Detection with Outlier Exposure. *In International Conference on Learning.* Representations.

Hernandez-Matamoros, A., Fujita, H., & Perez-Meana, H. (2020). A novel approach to create synthetic biomedical signals using BiRNN. *Information Sciences, 541*, 218–241. https://doi.org/10.1016/j.ins.2020.06.019

Hüsken, M., & Stagge, P. (2003). Recurrent neural networks for time series classification. *Neurocomputing, 50*, 223–235. https://doi.org/10.1016/s0925-2312(01)00706-8

Hussain, S. F., & Mian Qaisar, S. (2022). Epileptic seizure classification using level-crossing EEG sampling and ensemble of sub-problems classifier. *Expert Systems with Applications, 191*, Article 116356. https://doi.org/10.1016/j.eswa.2021.116356

Kang, S. (2022). Using binary classifiers for one-class classification. *Expert Systems with Applications, 187*, Article 115920. https://doi.org/10.1016/j.eswa.2021.115920

Karmy, J. P., López, J., & Maldonado, S. (2021). Simultaneous model construction and noise reduction for hierarchical time series via Support Vector Regression. *Knowledge-Based Systems, 232*, Article 107492. https://doi.org/10.1016/j.knosys.2021.107492

Khan, S. S., & Ahmad, A. (2018). Relationship between Variants of One-Class Nearest Neighbors and Creating Their Accurate Ensembles. *IEEE Transactions on Knowledge and Data Engineering, 30*(9), 1796–1809. https://doi.org/10.1109/tkde.2018.2806975

Lee, Y., Wei, C., Cheng, T., & Yang, C. (2012). Nearest-neighbor-based approach to time-series classification. *Decision Support Systems, 53*(1), 207–217. https://doi.org/10.1016/j.dss.2011.12.014

Lenz, O. U., Peralta, D., & Cornelis, C. (2021). Average Localised Proximity: A new data descriptor with good default one-class classification performance. *Pattern Recognition, 118*, Article 107991. https://doi.org/10.1016/j.patcog.2021.107991

Li, G., & Jung, J. J. (2022). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion, 91*, 93–102. https://doi.org/10.1016/j.inffus.2022.10.008

Li, M., Sun, H., Lin, C., Li, C., & Guo, J. (2022). The devil in the tail: Cluster consolidation plus cluster adaptive balancing loss for unsupervised person re-identification. *Pattern Recognition, 129*, Article 108763. https://doi.org/10.1016/j.patcog.2022.108763

Lin, J., Keogh, E. J., Lonardi, S., & Chiu, B. Y. (2003). A symbolic representation of time series, with implications for streaming algorithms. *Workshop on Research Issues on Data Mining and Knowledge Discovery*. https://doi.org/10.1145/882082.882086

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. In *2008 eighth IEEE international conference on data mining* (pp. 413–422). doi: 10.1109/icdm.2008.17.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam, & J. Neyman (Eds.), *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297). California: University of California Press.

Mauceri, S., Sweeney, J. F., & McDermott, J. (2020). Dissimilarity-based representations for one-class classification on time series. *Pattern Recognition, 100*, Article 107122. https://doi.org/10.1016/j.patcog.2019.107122

Merdjanovska, E., & Rashkovska, A. (2022). Comprehensive survey of computational ECG analysis: Databases, methods and applications. *Expert Systems with Applications, 203*, Article 117206. https://doi.org/10.1016/j.eswa.2022.117206

Parija, S., Bisoi, R., Dash, P. K., & Sahani, M. (2021). Deep long short term memory based minimum variance kernel random vector functional link network for epileptic EEG signal classification. *Engineering Applications of Artificial Intelligence, 105*, Article 104426. https://doi.org/10.1016/j.engappai.2021.104426

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *the Journal of machine Learning research, 12*, 2825–2830.

Rjoob, K., Bond, R., Finlay, D. D., McGilligan, V., Leslie, S. J., Rababah, A., … Macfarlane, P. W. (2022). Machine learning and the electrocardiogram over two decades: Time series and meta-analysis of the algorithms, evaluation metrics and applications. *Artificial Intelligence in Medicine, 132*, Article 102381. https://doi.org/10.1016/j.artmed.2022.102381

Ruff, L., Vandermeulen, R. A., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. In *International Conference on Machine Learning* (pp. 4393–4402). http://proceedings.mlr.press/v80/ruff18a/ruff18a.pdf.

Sadek, I., Seet, E., Biswas, J., Abdulrazak, B., & Mokhtari, M. (2018). Nonintrusive vital signs monitoring for sleep apnea patients: A preliminary study. *IEEE Access, 6*, 2506–2514. https://doi.org/10.1109/access.2017.2783939

Sánchez-Reolid, R., López de la Rosa, F., López, M. T., & Fernández-Caballero, A. (2022). One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity. *Biomedical Signal Processing and Control, 71*, Article 103203. https://doi.org/10.1016/j.bspc.2021.103203

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13* (7), 1443–1471. https://doi.org/10.1162/089976601750264965

Singer, G., Ratnovsky, A., & Naftali, S. (2021). Classification of severity of trachea stenosis from EEG signals using ordinal decision-tree based algorithms and ensemble-based ordinal and non-ordinal algorithms. *Expert Systems with Applications, 173*, Article 114707. https://doi.org/10.1016/j.eswa.2021.114707

Studnicka, F. (2022a). Ballistocardiography sleep dataset. *Mendeley Data, V2*. https://doi.org/10.17632/8yzmk4dd7p.2

Studnicka, F. (2022b). Ballistocardiography with breathing disorderes. *Mendeley Data, V3*. https://doi.org/10.17632/9fmfn6kfn7.3

Wang, J., Zhang, D., & Zhou, Y. (2022). Ensemble deep learning for automated classification of power quality disturbances signals. *Electric Power Systems Research, 213*, Article 108695. https://doi.org/10.1016/j.epsr.2022.108695

Wang, Y., Wu, D., Yu, Q., Zhu, D., & Meng, F. (2021). A weather signal detection algorithm based on EVD in elevation for airborne weather radar. *Digital Signal Processing, 116*, Article 103118. https://doi.org/10.1016/j.dsp.2021.103118

Yang, Y., Hou, C., Lang, Y., Yue, G., & He, Y. (2019). One-Class Classification Using Generative Adversarial Networks. *IEEE Access, 7*, 37970–37979. https://doi.org/10.1109/access.2019.2905933

Zhang, L., Lim, C. P., Yu, Y., & Jiang, M. (2021). Sound classification using evolving ensemble models and Particle Swarm Optimization. *Applied Soft Computing, 116*, Article 108322. https://doi.org/10.1016/j.asoc.2021.108322

Zhu, H., Zhang, J., Cui, H., Wang, K., & Tang, Q. (2021). TCRAN: Multivariate time series classification using residual channel attention networks with time correction. *Applied Soft Computing, 114*, Article 108117. https://doi.org/10.1016/j.asoc.2021.108117

Zhu, H., Liu, S., & Jiang, F. (2022). Adversarial training of LSTM-ED based anomaly detection for complex time-series in cyber-physical-social systems. *Pattern Recognition Letters, 164*, 132–139. https://doi.org/10.1016/j.patrec.2022.10.017