Information Sciences 614 (2022) 71–86

Contents lists available at ScienceDirect

Information Sciences

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

OCSTN: One-class time-series classification approach using a signal transformation network into a goal signal

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]

[a] *Faculty of Science, University of Hradec Kralove, Hradec Kralove, Czech Republic*
[b] *Faculty of Information Technology, HUTECH University, Ho Chi Minh City, Viet Nam*
[c] *Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia*
[d] *Regional Research Center, Iwate Prefectural University, Takizawa, Japan*

ARTICLE INFO

Article history:
Received 11 March 2022
Received in revised form 3 September 2022
Accepted 5 September 2022
Available online 26 September 2022

Keywords:
Time-series classification
One-class classification
One-class time-series classification
Signal transformation
Anomaly detection

ABSTRACT

One-class classification (OCC) is a classification task where the training data have only one class. The goal is to classify input data into one seen class or other unseen classes. This paper proposes an OCC approach using a signal transformation network (OCSTN), which aims to process univariate time-series data. The main contribution is developing a signal transformation network (STN) that aims to transform input signals into one signal, namely the goal signal. Moreover, the model error of the STN is a distance metric between the goal signal and the model output. The STN model learns from one-class signals. Therefore, model error for one class is small relative to other classes. Accordingly, OCSTN could discriminate between seen and unseen classes using the model errors. The proposed OCSTN is evaluated using two ballistocardiography (BCG) datasets. The OCSTN achieves fair results in both AUC scores and processing speed. OCSTN has a weak point in training diverse signals. In addition, the entropy and smoothness of the goal signal are highly related to the AUC score.

1. Introduction

Time series data are a set of observations obtained in chronological order. Each observation records a phenomenon at a specific moment [1]. Such data are represented as a sequence or signal and are used in various areas, such as biomedical signals [2], financial records [3], weather readings [4], and energy usage [5]. Time-series classification (TSC) is a significant challenge that has attracted attention, and researchers have proposed various TSC algorithms. These methods are roughly

* 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), h.fujita@hutech.edu.vn, hfujita-799@acm.org (H. Fujita), damian.busovsky@uhk.cz (D. Bušovský), richard.cimler@uhk.cz (R. Cimler).

[1] ORCID: 0000-0002-7599-4404.
[2] ORCID: 0000-0003-2197-8553.
[3] ORCID: 0000-0001-6721-8678.
[4] ORCID: 0000-0001-5256-210X.
[5] ORCID: 0000-0001-8853-8224.
[6] ORCID: 0000-0001-6712-9894.

https://doi.org/10.1016/j.ins.2022.09.027
0020-0255/© 2022 Elsevier Inc. All rights reserved.

classified into four groups: distance-based [6], feature-based [7], deep learning (DL)-based [8], and ensemble approaches [9]. Nevertheless, TSC is still a challenging issue. Since these methods are supervised learning, classification results are affected due to the problems of training data, such as data imbalance [10], noisy labels [11], and outliers [12]. Moreover, the model cannot classify data into unseen classes not included in the training data. In addition, collecting enough data is not possible because some data are rare, dangerous to collect, and nonexistent.

One-class classification (OCC) [13–24] is a promising solution to tackling these problems. OCC is a classification task where only training data have only one class; the classification model needs to learn from one class and classify data into one seen class or other unseen classes. The main advantage of OCC is its function to detect unseen samples. This function could improve binary or multiclass classification by approximating these classification problems as the set of OCC problems. Such a method can tackle the data imbalance problems because data imbalance does not exist in one class [13]. On the other hand, OCC algorithms can detect outliers or noise by training only normal samples.

For this purpose, the researchers proposed various OCC algorithms. Early studies were shallow methods [14–17]. These methods are suitable for feature data, but there are limitations in the feature extraction process from image or time-series data [18]. Recently, DL methods have been proposed to tackle image data. These methods are roughly classified into three groups: feature extraction [18], fake unseen [19], and self-supervised approaches [20]. Nevertheless, OCC for time-series data is still a challenging issue, and only a few studies exist for such purposes [21,22]. Therefore, considering one-class time-series classification is a significant challenge.

One general solution is to apply the OCC algorithms in features/images into time-series data. For this idea, this study applies the OCITN algorithm [22] to time series data. OCITN is a self-supervised approach that uses an image transformation network (ITN) as a pretext task. ITN aims to transform all images into one image, namely, the goal image. In addition, the model error for ITN is the distance metric between the goal image and the output of ITN. Such a network learns using only one-class images. Therefore, the model errors for seen images are considered small relative to images belonging to unseen classes. Accordingly, OCITN could discriminate between seen and unseen classes using the model error.

Besides, OCITN has three merits to extend into time-series data. First, OCITN shows promising results in both the AUC score and processing speed [23]. Second, the OCITN is simple (ITN is implemented with only convolutional layers [23]) and extending into time-series is not difficult. Third, OCITN has an interesting aspect: goal image is highly related to the AUC score (High entropy and smooth images appear to be better AUC [24]). Considering such an aspect in time-series data is fascinating.

In summary, the motivations in this study are as follows:

Supervised time-series classification has limitations due to the problems in the training dataset.

OCC is a promising solution to address these limitations. However, only few OCC methods exist for time-series data.

We have a strong interest in extending the OCITN algorithm to time-series data.

Accordingly, this paper proposes a one-class time-series classification approach that uses a signal transformation network (OCSTN). This strategy is an extension of the OCITN algorithm [23] to univariate time-series data. The main difference is the data structure: while an image is a two-dimensional matrix, the signal is a one-dimensional sequence. Such a difference is worthy of investigation. The proposed solution uses a signal transformation network (STN), which aims to transform all signals into one signal, namely goal signal. In addition, the model error of the STN is the distance metric between the goal signal and the output of the STN. Such a network is trained using only one-class signals; the model error for seen signals should be small relative to unseen signals. Therefore, OCSTN could discriminate between seen and unseen signals using the model error.

The contributions in this paper are as follows:

This paper proposes a novel one-class time series classification algorithm, OCSTN, which discriminates between seen and unseen classes using the model error of the signal transformation network (STN).

The main originality is to bring the transformation subtask to time-series data. Such a study has significant meaning because images and time series have substantial differences in terms of data structure. In addition, the proposal includes two novel concepts, the STN and goal signal. Furthermore, this paper is the first article that uses [49] for the experiment.

OCSTN is evaluated using two ballistocardiography (BCG) datasets [49,50]. The experiments are made with the authentication and the change detection tasks. OCSTN shows fair AUC score and processing speed for the authentication task. On the other hand, OCSTN has a weak point in training diverse signals, such as sliding windows extracted for the change detection task.

New discussion is provided for STN structure and goal signal. According to the experiment results, linear is the best activation function in terms of AUC. In addition, the smoothness of the goal signal is related to the AUC score.

We have created a minor modification of the method from Blazquez-Garcia et al. [22] for computing the AUC score. This contribution allows for making a comparison.

The organization of this paper is as follows. Section 2 describes related work. Section 3 presents the proposed OCSTN framework. Section 4 and Section 5 provide the experimental results and discussion, respectively. Finally, Section 6 gives the conclusions and future work.

2. Related work

2.1. Time-series classification

TSC is a significant challenge and has attracted increasing attention. TSC consists of two stages: preprocessing and classification. The first stage transforms the raw signal into a suitable form for classification. Several techniques have been proposed, such as wavelet transform [25], adaptive thresholding [26], wavelet-based atomic function [27], Cartan Curvatures [2,28], and Euclidean arc length [29].

On the other hand, the classification stage can be classified into four types: distance-based, feature-based approaches, DL-based, and ensemble approaches. Distance-based methods classify data into the same class according to the most similar signals [6]. These methods are known as 1-nearest neighbors or K-nearest neighbors. The researchers apply various distance metrics [21] to compute similar samples.

In contrast, the feature-based method has two stages: feature extraction and classification. Several feature extraction methods have been proposed for time-series data, such as time-series forest [7], dissimilarity-based representation [21], bag of features [30], and symbolic aggregate approximation (SAX) [31]. These extracted features will be inputs for the classification algorithm.

In addition, the DL-based approach extends the feature-based method in an end-to-end framework. Several types of architectures are applied to feature extraction and classification of the signal [8]. For feature extraction, Conv1D captures local sequence patterns [32]. In addition, RNN and LSTM capture temporal information [33,34]. Moreover, residual networks and attention improve the classification accuracy [36]. Furthermore, model optimization algorithms are proposed [37].

In addition, the ensemble approach combines multiple existing techniques and involves fusions between multiple features and classifiers. Bai et al. combined the mean and trend features for TSC [9]. Hussain et al. applied a one-vs-all ensemble that is combination of classifiers [35]. These studies aim to create real applications.

Nevertheless, TSC have limitations because of supervised learning, where the classification model cannot predict unseen data not included in the training data [38]. Moreover, the classification result is affected by several dataset problems, such as data imbalance [10], label noise [11], and outliers [12]. This study considers OCC to tackle these problems.

2.2. One-class classification

OCC is a classification problem where training data have only one seen class. The goal of OCC is to classify data into one seen class or other unseen classes. OCC is applicable to several tasks, such as anomaly detection [39], novelty detection [40]. In addition, combining multiple OCC models could add the function of detecting unseen samples to binary and multiclass classification.

Various OCC algorithms have been proposed in machine learning and DL. Early studies were shallow methods, such as the one-class support vector machine [14], local outlier factor [15], isolation forest [16], and one-class nearest neighbors [17]. These methods are effective for feature data. However, feature extraction is needed to process other data types, such as images and time series.

Recently, DL-based OCC methods have been proposed for image data. These methods are roughly classified into three groups: feature extraction methods [18], fake unseen methods [19], and self-supervised methods [20].

The first strategy extracts the features from images. These features are the input for the shallow OCC algorithms. An autoencoder is a common technique for feature extraction processes. Moreover, Ruff et al. considered a convolutional neural network as the mapping function in feature vector space and computed the SVDD in such a vector space [18].

In contrast, the fake unseen method creates fake unseen samples and trains a binary classification model between the seen class and the fake unseen class. Several techniques like GAN [41] are applied to generate fake unseen samples. In addition, outlier exposure is the technique to import other datasets as unseen data [19].

Besides, the self-supervised approach uses the pretext task, the subtask, to support the downstream process (OCC). This study trains the supervised model for the subtask of using seen data. Since the model is trained from only the seen class, the model error for the seen data is small relative to the unseen data. Therefore, using model error can classify seen and unseen data. Several tasks are proposed as pretext tasks, such as autoencoder [42], and classification of geometric transformation [20].

Nevertheless, OCC for time-series data is still a challenging issue, and only few studies exist. Mauceri et al. proposed a dissimilarity-based feature representation and used those features as a one-class 1-nearest neighbor classifier input [21]. A self-supervised approach is also applied for time-series data. Baldacci et al. proposed a forecasting subtask for detecting anomalous points [43]. Moreover, Blázquez-García et al. provided a classification subtask of signal multiplication for water leak detection [22]. Since there are few existing studies, more alternative methods are needed. This study aims to extend the OCITN algorithm to time-series data.

2.3. One-class image transformation network

OCITN is an OCC algorithm for image data [23]. Such an algorithm is a self-supervised approach using image transformation into one image, namely, the goal image. ITN is the transformation function between input images into a goal image. In addition, the model error for ITN, namely, the construction error (CE), is the distance metric between the goal image and the output of ITN. Such a network trains using only one-class images. Therefore, CE for the seen image is considered small relative to the unseen image. Accordingly, using CE could discriminate between seen and unseen classes [23].

For example, Fig. 1 shows the OCITN framework [24]. The seen and unseen images are presented as apple and banana images, respectively. Moreover, the goal image is presented as Lenna. The ITN model trains image transformation from apples into Lenna. Therefore, ITN could transform apples into Lenna. The core assumption is that ITN cannot transform bananas into Lenna because bananas are not used to train the ITN model. Accordingly, OCITN could discriminate between seen apples and unseen bananas using image transformation into a goal image.

OCITN is implemented using the following Equation (1).

$$OCITN(X) = \begin{cases} Seen\ \ (dist(ITN(X), I) < \lambda) \\ Unseen\ \ (dist(ITN(X), I) \geq \lambda) \end{cases} where\ \ (ITN:\ \ X_{seen} \rightarrow I) \tag{1}$$

where X and I are input images and a goal image, respectively. Then, ITN aims to transform seen images X_{seen} into goal image I. In addition, $dist\ (ITN(X), I)$ is the construction error, the distance metric between a goal image and ITN output. Finally, λ is the threshold value to discriminate between seen and unseen data [24].

The concept of the OCITN algorithm is explored in two articles. The article [23] proposed the OCITN algorithm. The paper [24] reported the experiment results using 236 goal objects and the discussion that high entropy and smoothness of goal images provide better AUC scores.

This study aims to extend such a transformation strategy to time-series data. The main difference is the structure between the image and signal. While an image is a two-dimensional matrix, the signal is a one-dimensional sequence. The main challenges are how to define a goal signal and how to transform signals.

3. One-class classification using signal transformation

This section proposes the OCC approach using a signal transformation network (OCSTN). Such a method is an extension of the OCITN algorithm [23] into univariate time-series data. The main novelty is using the signal transformation into a goal signal.

Fig. 2 shows the OCSTN framework, consisting of training and testing stages. In the training stage, training data include signals for one seen class. Then, the signal transformation between these signals into one defined signal, namely, the goal signal, is considered. The signal transformation network (STN) is trained for such purposes. Since the model training is made using seen data, model errors for seen signals are small relative to unseen signals. Therefore, using model error could discriminate between seen classes and unseen classes.

The following paragraphs provide mathematical descriptions. These equations are defined by replacing the previous Eqs. (1) in Section 2.3. Signal data and goal signals are defined as sequences X and I, as shown in Equations (2) and (3).

$$\text{Input signal}\ \ X = [X_1,\ \ X_2,\ \ \cdots,\ \ X_L] \tag{2}$$

$$\text{Goal signal}\ \ I = [I_1,\ \ I_2,\ \ \cdots,\ \ I_L] \tag{3}$$

where L is the length of the signals.

STN aims to transform seen signals into a goal signal, as shown in Formula (4).

$$STN:\ \ X_{seen} \rightarrow I \tag{4}$$

where X_{seen} represents the signals included in the training data. The STN is a neural network, and its input and output are the signals and a goal signal, respectively. The transformation process is a black box function between the input and output. Moreover, the STN is evaluated with construction error E, which is the distance metric between the goal signal and the output of the STN. Such error is computed for each input X, as shown in Eq. (5).

$$E(X) = dist(STN(X), I) = \frac{1}{L}\sum_{l=1}^{L}|STN(X)_l - I_l| \tag{5}$$

In the training stage, training data are split into Xtrain and Xvalid, as shown in Equations (6) and (7).

$$Xtrain = [Xtr_1, Xtr_2, \cdots, Xtr_m, \cdots, Xtr_M] \tag{6}$$

$$Xvalid = [Xva_1, Xva_2, \cdots, Xva_n, \cdots, Xva_N] \tag{7}$$

where M and N are the numbers of the data. Xtrain is used to update the weight of the STN. In contrast, Xvalid is used to validate STN. The STN is trained to minimize the construction error E for Xvalid, as shown in Formula (8).

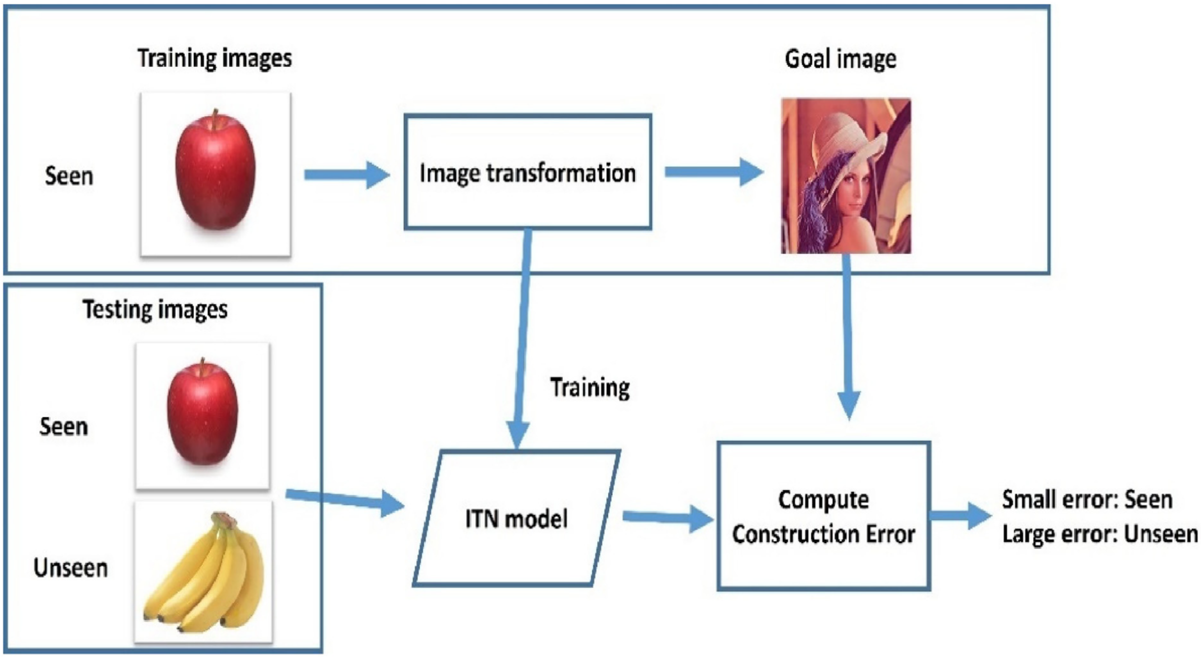

Fig. 1. One-class image transformation network [24].

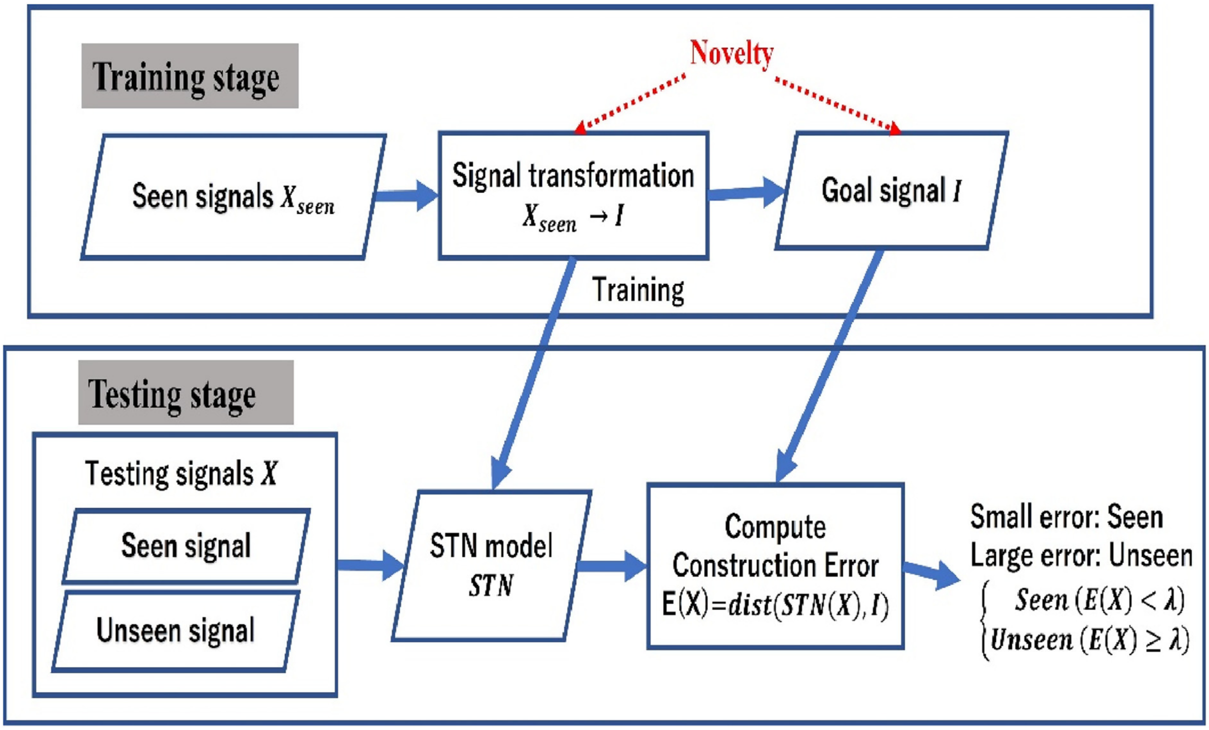
Training stage
Novelty
Seen signals X_{seen}
Signal transformation $X_{seen} \rightarrow I$
Goal signal I
Training
Testing stage
Testing signals X
Seen signal
Unseen signal
STN model STN
Compute Construction Error $E(X)=dist(STN(X), I)$
Small error: Seen
Large error: Unseen
$\begin{cases} Seen\ (E(X) < \lambda) \\ Unseen\ (E(X) \geq \lambda) \end{cases}$

Fig. 2. OCSTN framework.

$$min \frac{1}{N} \sum_{n=1}^{N} E(Xva_n) \tag{8}$$

In the testing stage, the STN transforms the testing signal X into the goal signal. Then, the model error is computed as the distance metric between the STN output and a goal signal.

Finally, the seen and unseen classes are discriminated by using a threshold value for the model error. In summary, OCSTN is represented as Eq. (9).

$$OCSTN(X) = \begin{cases} Seen & (E(X) < \lambda) \\ Unseen & (E(X) \geq \lambda) \end{cases} \tag{9}$$

where λ is the threshold value to discriminate between the seen and unseen signals.

4. Experiments

This section provides the experimental results. Section 4.1 explains the dataset, Section 4.2 describes the evaluation metric, and Section 4.3 provides the experimental results. Finally, Section 4.4 shows a comparison with other one-class time-series classification algorithms.

4.1. Data

This study uses ballistocardiography (BCG) signals for training datasets. BCG is a measurement of the recoil forces of the body in reaction to cardiac ejection of blood into the vasculature [44].

This study considers two tasks for the OCSTN experiments. The first task is authentication. The data are BCG signals collected from 20 people. These signals are annotated with the ID of the signal holders [49]. The objective is to classify signals into a correct person or not. Signals from one person are included in the training data, and other signals are regarded as unseen. Such a setting corresponds to OCC. Table 1 provides the data balance for each class. Moreover, the length of the signal is 660.

The dataset is uploaded at [49]. Raw data are collected from four sensors and then preprocessed by the Cartan curvature [2] and Euclidean arc length [29] to compute two signals; these signals are concatenated to obtain the model input signals.

In contrast, the second task is to detect the critical changes in the signal using a breathing dataset [50]. These data are collected from tested individuals who change their behavior according to the measuring schedule [2]. The details are provided in Section 5.3.

4.2. Evaluation metric

The area under the ROC curve (AUC) is used as an evaluation metric. The ROC curve is a graph that plots the performance in all thresholds, as shown in Fig. 3.

The x-axis and y-axis are the False-Positive Rate (FPR) and True Positive Rate (TPR), respectively. These values are computed using Equations (10) – (11) and the confusion matrix shown in Table 2. Positive and negative correspond to seen and unseen classes, respectively.

$$\mathrm{TPR} = \frac{TP}{TP + FN} \tag{10}$$

$$\mathrm{FPR} = \frac{FP}{FP + TN} \tag{11}$$

4.3. Experimental results

The experimental data are split into training and testing data according to 80 %: 20 %. Then, 70 % of the training data are used to update the weights of the STN. Moreover, another 30 % of the training data are used to validate the STN. Such a split is done with five random seeds, and the average and standard deviations are reported.

Table 3 shows the STN structure, which consists of seven Conv1D layers. For each layer, the convolution size (Conv size) is 25, and the activation function is linear. Moreover, zero-padding is applied to maintain the signal size during the convolution. In addition, the batch size and the epoch number are 16 and 100, respectively. This setting is obtained by parameter tuning.

In addition, the goal signal is a parameter of the OCSTN algorithm. This study considers seven goal signals. Fig. 4 shows the six goal signals used in the experiment.

These signals are selected randomly and are created using the following equations. Let $I_n[T]$ be the T-th value in the n-th goal signal I_n; then, all values are computed as shown in Equations (12) – (17).

$$I_1[T] = \frac{T}{L} \qquad 12)$$

Table 1
Data balance for the dataset.

Person ID	Number of signals
0	2082
1	2116
2	1990
3	1739
4	2180
5	2226
6	863
7	1600
8	2202
9	2063
10	2031
11	1540
12	1641
13	1491
14	2247
15	2108
16	1562
17	2142
18	1410
19	1862

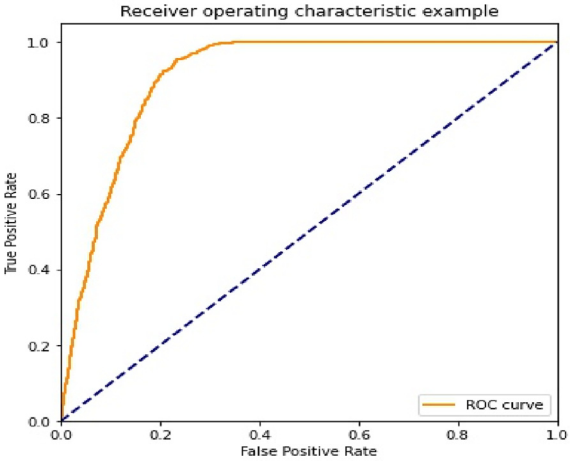

Fig. 3. Area under the ROC curve.

$$I_2[T] = \frac{T^2}{L^2} \tag{13}$$

$$I_3[T] = 0.5 \tag{14}$$

$$I_4[T] = 1 - \frac{\left(\frac{L}{2} - T\right)^2}{\left(\frac{L}{2}\right)^2} \tag{15}$$

Table 2
Confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Table 3
Structure of the Signal Transformation Network.

Layer	Output Shape	Conv size	Param #	Activation	Padding
1. Conv1D	(None, 660, 16)	25	416	linear	zero-padding
2. Conv1D	(None, 660, 16)	25	6416	linear	zero-padding
3. Conv1D	(None, 660, 16)	25	6416	linear	zero-padding
4. Conv1D	(None, 660, 16)	25	6416	linear	zero-padding
5. Conv1D	(None, 660, 16)	25	6416	linear	zero-padding
6. Conv1D	(None, 660, 16)	25	6416	linear	zero-padding
7. Conv1D	(None, 660, 1)	25	6416	linear	zero-padding

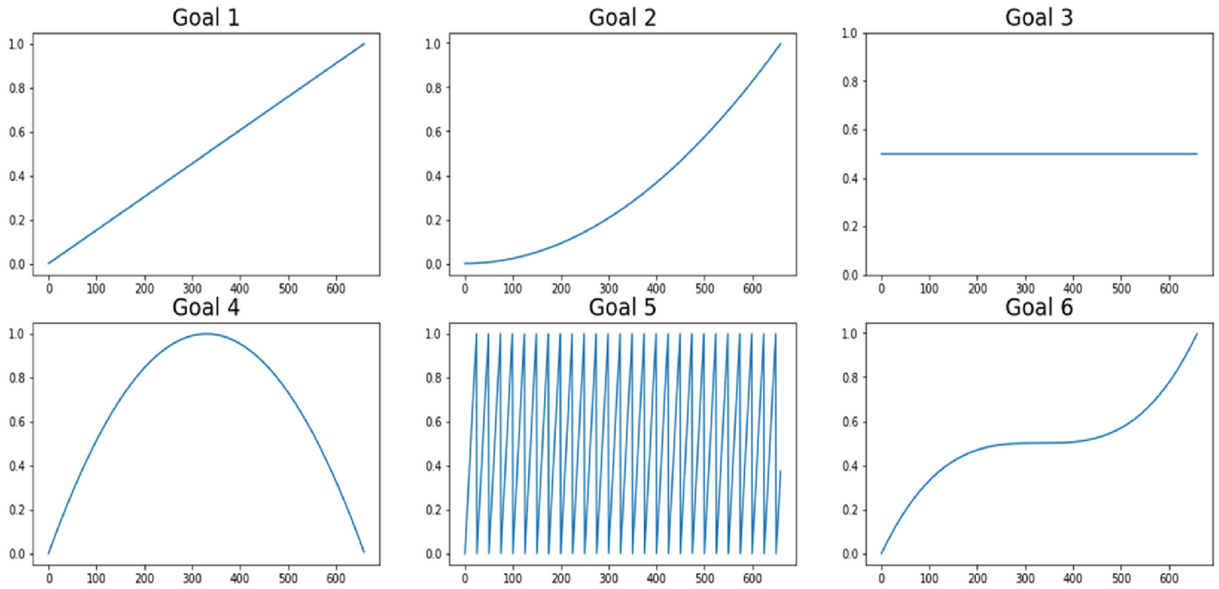

Fig. 4. Goal signals used in the experiment.

$$I_5[T] = \frac{(Tmod25)}{24} \tag{16}$$

$$I_6[T] = \frac{((\frac{L}{2} - T))^3}{2(\frac{L}{2})^3} + 0.5 \tag{17}$$

where L is the length of the signal. This experiment is performed where L = 660.

In addition, Goal 7 is computed as the average of the training signals. Therefore, Goal 7 changes according to the training signals. Please note that the goal signals are parameters, and other signals could be applicable as goal signals.

Table 4 provides the experimental results using these seven goal signals. Each cell shows an AUC score that corresponds to the pair of seen users and goal signals. In addition, the reported scores are averages and standard deviations of 25 experiments.

Due to the randomness of the neural network, the AUC score has a large variance. To avoid a variance, using the summation of multiple model errors is a promising solution. Such a strategy can avoid the lowest score. Table 5 provides AUC scores computed with a summation of five model errors. The reported score is the average of five experiments, which use 25 STN models trained for Table 4.

Overall, the summation of model errors provides a higher AUC score than using a single model error. In both settings, goal 7 shows the best AUC score. The source code is provided at https://github.com/ToshiHayashi/OCSTN.

4.4. Comparison

The proposed OCSTN is compared with existing one-class time-series classification algorithms in terms of the AUC score and processing time. Two comparative methods are selected as follows:

Mauceri et al. proposed a dissimilarity-based representation with a one-class 1-nearest neighbor classifier [21], which was implemented using the source code provided by Mauceri et al. [45]. In their original paper, 12 distance metrics and

Table 4
AUC scores using single model error.

Seen user	Goal 1	Goal 2	Goal 3	Goal 4	Goal 5	Goal 6	Goal 7
0	86.0 ± 5.5	88.6 ± 2.1	67.8 ± 22.3	81.6 ± 5.8	43.4 ± 0.8	81.6 ± 7.1	88.5 ± 3.3
1	75.6 ± 5.2	77.2 ± 5.5	69.3 ± 11.5	77.3 ± 4.9	58.9 ± 0.4	70.3 ± 11.7	81.0 ± 2.7
2	68.2 ± 8.4	69.8 ± 4.9	63.5 ± 7.4	68.7 ± 5.4	56.4 ± 0.3	65.4 ± 6.7	69.7 ± 6.9
3	83.9 ± 3.2	84.1 ± 3.2	73.1 ± 9.1	81.9 ± 5.0	48.6 ± 1.6	76.8 ± 12.8	83.8 ± 6.3
4	84.3 ± 9.7	83.9 ± 6.9	81.0 ± 6.9	79.0 ± 8.5	66.5 ± 0.6	71.0 ± 18.8	84.7 ± 8.4
5	80.6 ± 5.9	83.4 ± 4.8	64.2 ± 18.2	80.6 ± 5.8	43.6 ± 0.6	78.0 ± 5.8	80.9 ± 9.3
6	70.6 ± 9.1	70.7 ± 8.8	66.6 ± 4.2	68.1 ± 8.2	48.0 ± 0.6	63.7 ± 11.0	68.5 ± 10.8
7	65.6 ± 5.1	62.5 ± 7.0	56.0 ± 13.4	63.6 ± 5.7	58.4 ± 0.3	53.8 ± 14.9	64.1 ± 7.3
8	76.2 ± 3.6	77.3 ± 3.1	70.8 ± 5.7	76.2 ± 2.7	56.6 ± 0.3	66.5 ± 9.8	78.0 ± 6.4
9	92.4 ± 4.4	90.9 ± 5.8	84.0 ± 11.1	90.9 ± 3.9	54.5 ± 1.7	88.9 ± 8.1	92.4 ± 4.9
10	98.4 ± 1.8	98.0 ± 2.5	99.1 ± 1.3	99.8 ± 0.2	100. ± 0.0	89.1 ± 17.6	98.9 ± 3.9
11	75.3 ± 10.6	74.1 ± 8.2	73.0 ± 5.4	77.6 ± 5.5	52.3 ± 0.8	63.5 ± 14.1	76.1 ± 10.4
12	70.9 ± 4.6	73.3 ± 4.1	69.3 ± 5.9	74.2 ± 4.0	53.8 ± 0.3	63.9 ± 4.4	73.0 ± 5.6
13	84.8 ± 3.7	85.8 ± 2.6	75.6 ± 11.0	84.4 ± 2.6	47.8 ± 0.9	77.0 ± 9.0	86.7 ± 2.7
14	79.9 ± 4.4	80.7 ± 4.0	70.0 ± 9.3	79.9 ± 5.3	57.0 ± 0.4	69.5 ± 11.0	82.2 ± 5.8
15	62.2 ± 1.4	61.1 ± 2.0	54.8 ± 3.4	59.5 ± 3.4	51.0 ± 0.3	56.4 ± 3.4	63.4 ± 1.7
16	90.0 ± 3.3	91.1 ± 4.6	84.8 ± 7.1	91.9 ± 3.5	61.0 ± 2.6	88.7 ± 5.8	93.8 ± 2.5
17	69.8 ± 4.1	71.5 ± 2.0	61.4 ± 8.5	68.6 ± 3.8	46.8 ± 0.6	65.9 ± 7.5	72.5 ± 1.6
18	74.0 ± 3.3	75.7 ± 2.5	73.7 ± 3.1	76.5 ± 2.6	53.0 ± 0.8	66.2 ± 8.3	72.2 ± 3.1
19	73.3 ± 9.0	75.5 ± 7.9	70.2 ± 9.8	73.5 ± 5.2	52.9 ± 0.8	63.6 ± 13.0	77.2 ± 6.7
Average	78.1	78.7	71.4	77.6	55.5	70.9	**79.4**

Table 5
AUC scores using the summation of five models' errors.

Seen user	Goal 1	Goal 2	Goal 3	Goal 4	Goal 5	Goal 6	Goal 7
0	89.6 ± 1.4	91.0 ± 0.7	86.1 ± 1.3	85.7 ± 4.1	43.3 ± 2.0	88.0 ± 2.2	90.6 ± 0.8
1	79.1 ± 3.0	80.1 ± 1.8	77.4 ± 4.7	80.3 ± 1.0	58.9 ± 1.4	76.5 ± 7.3	83.8 ± 0.7
2	71.9 ± 4.6	75.2 ± 2.7	67.0 ± 5.6	76.7 ± 2.4	56.5 ± 1.6	76.0 ± 2.1	75.9 ± 4.3
3	86.6 ± 0.8	86.8 ± 0.7	84.4 ± 2.1	85.6 ± 0.8	48.6 ± 1.3	84.2 ± 2.3	86.6 ± 0.4
4	88.9 ± 2.8	88.2 ± 2.4	87.4 ± 2.0	86.6 ± 4.6	66.6 ± 0.7	82.9 ± 6.1	89.1 ± 4.6
5	86.1 ± 2.4	87.2 ± 0.9	78.6 ± 4.3	85.6 ± 0.8	43.6 ± 1.2	85.6 ± 2.9	87.3 ± 1.2
6	76.7 ± 2.3	76.6 ± 1.2	72.1 ± 3.5	72.5 ± 3.6	48.0 ± 2.3	71.7 ± 8.4	77.2 ± 1.0
7	67.5 ± 5.8	63.9 ± 2.4	57.5 ± 12.0	67.4 ± 2.0	58.5 ± 0.8	59.1 ± 8.9	66.6 ± 5.0
8	79.3 ± 3.5	80.4 ± 0.7	76.4 ± 4.2	78.7 ± 2.3	56.6 ± 1.2	74.0 ± 5.7	80.8 ± 4.5
9	95.1 ± 1.0	94.9 ± 0.8	93.1 ± 1.7	93.0 ± 1.4	54.6 ± 2.1	94.2 ± 1.0	95.1 ± 0.5
10	100. ± 0.0	99.9 ± 0.1	100. ± 0.0	100. ± 0.0	100. ± 0.0	98.4 ± 2.7	99.9 ± 0.1
11	80.3 ± 3.2	83.0 ± 2.6	80.5 ± 1.8	81.5 ± 2.3	52.3 ± 1.9	77.3 ± 7.0	82.9 ± 2.7
12	77.7 ± 2.2	78.3 ± 1.2	74.4 ± 2.2	78.5 ± 1.6	53.8 ± 1.3	72.5 ± 8.5	75.3 ± 2.8
13	88.7 ± 0.7	88.5 ± 1.0	84.3 ± 2.0	87.7 ± 0.6	47.8 ± 1.4	85.2 ± 2.5	88.9 ± 0.9
14	84.2 ± 1.1	85.9 ± 0.5	80.2 ± 3.5	83.2 ± 2.9	57.0 ± 1.2	79.5 ± 3.7	85.8 ± 1.4
15	63.9 ± 0.6	62.7 ± 1.1	59.6 ± 2.3	61.7 ± 3.5	51.0 ± 0.8	62.1 ± 3.1	64.4 ± 0.4
16	91.5 ± 1.5	93.1 ± 1.7	89.7 ± 5.2	94.1 ± 0.9	61.2 ± 1.5	92.8 ± 2.6	94.9 ± 0.7
17	73.5 ± 1.4	73.9 ± 1.0	69.6 ± 1.3	71.7 ± 1.9	46.7 ± 1.4	73.3 ± 1.1	73.7 ± 0.7
18	77.3 ± 0.8	77.5 ± 1.2	76.7 ± 2.6	78.0 ± 2.3	53.1 ± 2.3	72.2 ± 5.1	73.4 ± 2.0
19	77.5 ± 2.9	79.2 ± 3.5	75.4 ± 3.8	76.9 ± 1.0	52.9 ± 1.4	69.3 ± 9.6	79.3 ± 4.3
Average	81.7	**82.3**	78.5	81.3	55.5	78.7	**82.6**

8 prototypes are used for the experiment [21]. In this comparison, the k-means-based prototype and 9 distance metrics are applied because trying all combinations is extensive, and some dissimilarity metrics require much computation time. Applied dissimilarity metrics are Kullback Leibler divergence (KL), Autocorrelation (AC), Chebyshev (CS), City block (CB), Cosine (Cos), Euclidean (Euc), Gaussian (Gauss), Sigmoid (Sig), and Wasserstein (WS).

Blazquez-Garcia et al. proposed a self-supervised approach using the classification of signal multiplication [22]. Such a method is implemented by following the paper [22], but a small change is applied to compute AUC (because the original paper did not use the AUC score [22]). According to the paper, the self-labeled dataset is created by considering four self-labels; self-labeled signals are made as 1, 1.7, 2.4, and 3.1 times the original signal [22]. Then, the classification model is trained from the self-labeled dataset. The RISE classifier in the sktime library [46] is applied. Finally, the classification model is used to compute the AUC score. The previous study did not use AUC. Therefore, we decided how to compute it by model accuracy and probability related to correct prediction. In particular, the model accuracy is computed using the "predict" function [46]. However, since the number of self-labels is only four, the model accuracy for each data point has only five patterns (100 %, 75 %, 50 %, 25 %, and 0 %). Such accuracy cannot discriminate between seen and unseen data. To address such an issue, the probability corresponding to correct prediction is computed using the "predict_proba" function [46], which can provide more detailed values. In the following tables, these metrics are written as Acc and Prob Acc, respectively.

The source codes are uploaded at https://github.com/ToshiHayashi/OCSTN.

Table 6 provides a comparison result. For OCSTN, goal 7 is used as the goal signal. OCSTN shows the second-best AUC. The method of Blazquez-Garcia et al. with Prob Acc is the best in terms of the AUC.

In addition, the processing speed is compared. While OCSTN uses GPU, comparative methods' implementations do not use it. For a fair comparison, the processing time is calculated in a non-GPU environment (Intel® Core™ i9-9900 K CPU @ 3.60 GHz, RAM 64 GB). Table 7 reports the processing time where training signals are taken from person ID 0. The training time and testing times are calculated separately. The numbers of training and testing signals are approximately 1600 and 7400, respectively.

The method from Mauceri et al. is the fastest, and OCSTN is the second fastest. In addition, the approach from Blazquez-Garcia et al. is slow in the testing stage because their system needs to create self-labeled data for all testing data. Such a process is time-consuming and is not applicable in real-time applications. In contrast, the OCSTN requires long training and a short testing time. Such an aspect is suitable for real-time processing. In addition, the training speed could be increased with a GPU environment.

This result is related to the previous study for image data [23]. The classification subtask has high AUC score but is slow. In contrast, generative subtasks like transformation are fast, but the AUC score is not the best. These aspects are trade-off practices. OCSTN could be one of the alternatives in this trade-off.

5. Discussion

5.1. Goal signal

In the experiment, the goal signal is related to the AUC score. Goal 5 shows the lowest AUC score, and the difference is significant compared to other signals. The reason is that the signal is not smooth, and the STN cannot output such a signal. This result is like previous experiments with images. Smooth goal images show higher AUC scores [24].

Signal entropy [47] is computed as shown in Equation (18).

$$H = -\sum_{v} p_v log_2(p_v) \tag{18}$$

where V is the number of value levels, and p is the probability associated with value level V. Such computation is done in the same way as image entropy [48]. However, defining the value level is a challenge in time-series. Roughly speaking, the highest entropy signals do not have the same values in the whole signal. Therefore, goal signals 1, 2, and 6 have high entropy. In contrast, goal 3 is the lowest entropy because all values in the signal are the same.

In addition, smoothness is related to the average signal derivative defined by Equation (19).

$$D(I) = \frac{\sum_{i}^{L-1} |I_i - I_{i+1}|}{L-1} \tag{19}$$

where I is the signal, and L is the length of the signal defined in Equation (3). A smaller derivative value is smoother.

Table 8 provides the signal entropy, average derivative, and average AUC scores for goal signals (Goal 7 is out of comparison because the signal entropy and derivatives change according to the seen class.).

Goal 3 provides a low AUC score because its entropy is 0. In contrast, Goal 5 gives a low AUC score because it is not as smooth as its high average derivative. On the other hand, Goal 6 shows a low AUC despite high entropy and smoothness. The center of signal 6 has quite similar values. However, these values are not equal, and assigned to different value levels. Such computation increases the entropy value.

In addition, Table 9 provides the signal entropy and average derivative of Goal 7 for all seen classes.

OCSTN shows a high AUC score where the average derivative is small, such as class 10 or 11. In contrast, AUC is low where the average derivative is large, such as class 7 or 8.

Overall, the goal signals with high entropy and low derivatives provide better AUC. In future work, OCSTN should have experimented with other goal signals. In particular, the optimization of the goal signal should be considered. Compared to the goal image, the goal signal can easily be defined as a function with fewer parameters. Such an aspect is suitable for optimization.

5.2. Structure of the signal transformation network and parameters

The STN is built with a convolution-only structure in the same way as the method for image data [23]. Apart from image data, Conv2D is replaced by Conv1D because the univariate signal is a sequence. In addition, the size and shape of the convolution have differences because the shape of the signal is more elongated than that of the image. Furthermore, zero-padding for the signal is applied to only the first and the last of the sequence. Therefore, the effect of zero-padding should be smaller than that of the image.

Table 6
Comparison results with other OCC methods (signal length is 660).

Seen user	Mauceri et al. [21], Dissimilarity + One-class 1-Nearest Neighbors, kmeans-based prototype									Blazquez-Garcia et al. [22]		**OCSTN (Goal 7)**	
	KL	AC	CS	CB	Cos	Euc	Gauss	Sig	WS	Acc	Prob Acc	1 STN	5 STN
0	86.7 ± 0.5	69.3 ± 0.1	47.7 ± 5.0	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
1	77. 4 ± 0.8	67.8 ± 2.4	75.3 ± 0.7	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
2	68.5 ± 0.7	63.1 ± 0.8	57.8 ± 2.0	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
3	77.5 ± 0.6	60.1 ± 2.1	22.8 ± 3.0	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
4	87.0 ± 0.6	87.8 ± 0.2	83.5 ± 0.4	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
5	79.3 ± 0.5	64.9 ± 1.0	54.7 ± 4.6	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
6	74.3 ± 1.3	78.9 ± 1.2	64.9 ± 3.7	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
7	74.2 ± 0.7	52.3 ± 1.7	67.9 ± 3.1	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
8	77.5 ± 3.9	67.8 ± 1.8	73.2 ± 0.3	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
9	89.8 ± 0/3	69.9 ± 0.6	41.9 ± 7.2	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
10	100 ± 0.0	100 ± 0.0	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
11	79.1 ± 0.9	66.5 ± 0.9	57.1 ± 3.1	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
12	74.3 ± 1.2	65.7 ± 1.5	64.5 ± 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
13	87.9 ± 0.5	66.5 ± 1.1	43.4 ± 2.5	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
14	72.4 ± 1.0	72.2 ± 2.8	70.2 ± 2.8	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
15	63.0 ± 0.4	55.7 ± 0.9	62.9 ± 0.9	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
16	84.4 ± 1.0	62.3 ± 1.2	31.7 ± 5.9	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
17	68.1 ± 0.9	65.9 ± 1.6	63.3 ± 1.0	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
18	72.3 ± 1.0	83.6 ± 1.5	81.8 ± 0.3	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
19	81.4 ± 1.5	57.6 ± 0.8	63.0 ± 1.0	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
Average	78.7	68.9	61.3	71.8	71.0	73.8	75.6	76.2	77.4	74.6	**88.6**	79.4	82.6

Table 7
Processing time without GPU.

Method	Training time (second)	Testing time (second)
Mauceri [21]	1.39	0.93
Blazquez-Garcia [22]	840	1440
OCSTN (1 STN)	276	3.46
OCSTN (5 STNs)	1178	16.8

Table 8
Signal Entropy and Average Derivative for Six Goal Signals.

Goal Signal	Signal Entropy	Average Derivative	AUC (1 STN)	AUC (5 STN)
Goal 1	9.366	0.001513	78.1	81.7
Goal 2	9.366	0.001511	78.7	82.3
Goal 3	0	0	71.4	78.5
Goal 4	8.369	0.003021	77.6	81.3
Goal 5	4.644	0.079356	55.5	55.5
Goal 6	9.366	0.001508	70.9	78.7

Table 9
Signal Entropy and Average Derivative for Goal 7.

Seen class	Signal Entropy	Average Derivative	AUC (1 STN)	AUC (5 STN)
1	9.366	0.002704	88.5 ± 3.3	90.6 ± 0.8
2	9.366	0.003052	81.0 ± 2.7	83.8 ± 0.7
3	9.366	0.002985	69.7 ± 6.9	75.9 ± 4.3
4	9.366	0.002661	83.8 ± 6.3	86.6 ± 0.4
5	9.366	0.002864	84.7 ± 8.4	89.1 ± 4.6
6	9.366	0.002623	80.9 ± 9.3	87.3 ± 1.2
7	9.366	0.003796	68.5 ± 10.8	77.2 ± 1.0
8	9.366	0.003357	64.1 ± 7.3	66.6 ± 5.0
9	9.366	0.002824	78.0 ± 6.4	80.8 ± 4.5
10	9.366	0.002385	92.4 ± 4.9	95.1 ± 0.5
11	5.198	0.001681	98.9 ± 3.9	99.9 ± 0.1
12	9.366	0.003149	76.1 ± 10.4	82.9 ± 2.7
13	9.366	0.003300	73.0 ± 5.6	75.3 ± 2.8
14	9.366	0.003178	86.7 ± 2.7	88.9 ± 0.9
15	9.366	0.002937	82.2 ± 5.8	85.8 ± 1.4
16	9.366	0.003007	63.4 ± 1.7	64.4 ± 0.4
17	9.366	0.002576	93.8 ± 2.5	94.9 ± 0.7
18	9.366	0.003113	72.5 ± 1.6	73.7 ± 0.7
19	9.366	0.003037	72.2 ± 3.1	73.4 ± 2.0
20	9.366	0.003139	77.2 ± 6.7	79.3 ± 4.3

Table 10
Comparison of activation functions.

Activation	Goal 1	Goal 2	Goal 3	Goal 4	Goal 5	Goal 6	Goal 7
linear	**78.1**	**78.7**	**71.4**	**77.6**	55.5	70.9	**79.6**
relu	66.3	66.8	55.3	67.3	64.3	67.2	58.9
sigmoid	64.4	65.3	52.2	61.5	59.3	67.2	58.4
softmax	50.0	50.0	50.0	50.0	50.0	50.0	50.0
softplus	70.4	67.2	52.4	70.3	65.2	72.9	66.1
softsign	70.0	68.8	51.2	62.0	64.8	68.6	71.0
tanh	71.7	72.8	55.7	68.3	65.2	68.6	72.8
selu	75.2	76.7	59.9	73.1	**68.6**	73.2	66.8
elu	73.0	74.0	65.1	70.3	65.8	**75.3**	74.0

In addition, the activation function is changed from relu to linear because linear shows the best AUC score, as shown in Table 10. In this comparison, the model uses only one type of activation function. The other structure is the same as that in Table 3. The reported score is the average of AUCs from 20 seen classes (with 1 STN).

Overall, linear shows the best AUC scores. However, it is not the best activation function for Goal 5 and Goal 6. There is a concern that if all activation functions are linear, the network collapses into a linear model. Goals 5 and 6 are not easy to construct by such a model. However, this problem does not affect most goal signals used in the experiment because the OCSTN needs only the model error of the STN, and such an error can be computed even in the network collapse.

Overall, the signal transformation process is different from the image transformation. Further study is necessary to consider what occurs inside the network.

5.3. Applying OCSTN to detect critical changes in the signal

In this section, OCSTN is applied to detect critical changes in the signal using the breathing dataset [50]. These data are collected from twenty tested individuals who change their behavior according to the measuring schedule, including events such as holding breath and changing body position [2], as shown in Table 11.

Where critical changes will exist during the time around the events. Accordingly, the OCSTN is applied to detect the signals around events.

For this purpose, training signals are created by sliding windows (size = 660) from 0 to 50 s for twenty people. These signals do not include events; there are no critical changes. On the other hand, testing signals are extracted by sliding windows from the whole signal.

Fig. 5 shows the construction errors for all signals of user 17. Red lines indicate the time when the events start.

This error is not clear in detecting the critical part. The problem is that signals extracted by sliding windows have diversity. In such a case, STN outputs from these signals have diversity and make up the variance of the construction errors.

To address such a problem, the hypothesis is that the neighboring parts of the time-series data are similar. In particular, the signals should be seen if their neighbors are seen. Accordingly, the local minimum construction error (LMCE) is defined as the sequence computed by applying sliding windows (size = 1000) to the construction error sequence and taking the minimum values from all windows. Fig. 6 shows the LMCE.

The OCSTN can detect changes in body position using LMCE. However, the OCSTN does not recognize breath holdings as significant changes. Perhaps BCG during breath holding include a window similar to normal windows. Since OCSTN is OCC and trains only the normal windows, the LMCE decreases where such a neighbor exists. In addition, the window size should be an important aspect of detection. Breathing is a process with a period of approximately 3–5 s. However, the window includes only 0.66 s, and it cannot cover the changes related to breathing. Further study is needed to address this issue.

5.4. How to apply the transformation subtask to other data types

This paper proposes an OCSTN for time-series data. In addition, OCITN was proposed for image data [23]. Therefore, the transformation subtask had tackled to images and time series. One of the following challenges is to apply the transformation strategy to further data types, such as multivariate-time-series data, 3D data, feature data, and text data.

OCITN can tackle multivariate time series because such data is a matrix. In addition, applying the Conv3D layer could create a transformation network for 3D data. On the other hand, OCSTN could solve feature data. However, such data do not have the order of the dimensions, and the convolutional transformation result will change corresponding to the feature order. In addition, the categorical feature will be a problem for transformation. Furthermore, applying a transformation subtask to text data has a challenge. The process could be transforming all words or all sentences into one goal. These ideas are future work.

6. Conclusions and future work

This study considers a one-class time-series classification problem and proposes the OCSTN algorithm for univariate signals. OCSTN considers the STN model to transform input signals into a goal signal. Such a model is trained using signals belongs to a seen class, and the model error is used to discriminate between seen and unseen classes. The proposed OCSTN shows a fair AUC score and reasonable processing time. Therefore, OCSTN could be considered an alternative solution for one-class time-series classification. The most interesting outcome is the relation between AUC scores and goal signals. OCSTN is suitable where goal signals are smooth and have high entropy.

Finally, future works are given as follows. Improving the AUC score is a significant issue. The problems are the structure of the STN, the appropriate goal signal, and how to process the short signal. Moreover, applying OCSTN to real applications should be considered. Deciding the threshold value is a requirement. Furthermore, applying a transformation strategy to other data types is an exciting challenge to detect unseen data from every data type.

CRediT authorship contribution statement

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

Table 11
Measuring schedule [2].

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

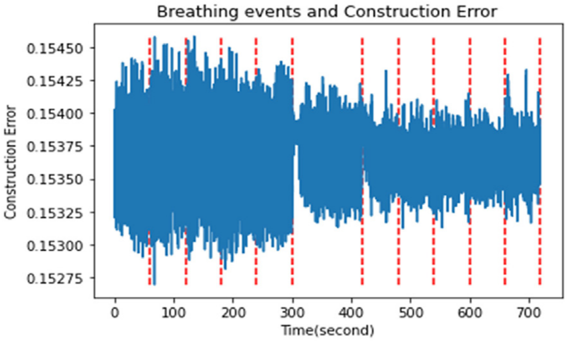

Fig. 5. Detecting the critical part using the construction error; the red line shows the time that the events start.

Data availability

The data and the link of the source program are mentioned in the paper

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.

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,

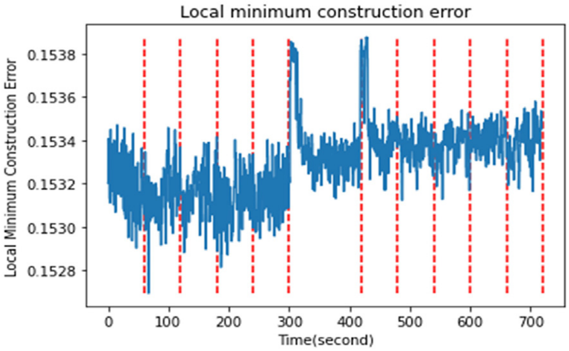

Fig. 6. Detecting the critical part using the local minimum construction error; the red lines show the time that the events start.

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 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) #JP20K11955.The dataset is available at https://data.mendeley.com/datasets/8yzmk4dd7p/1.The source code is available at https://github.com/ToshiHayashi/OCSTN.

References

[1] Wen Xin Cheng, P.N. Suganthan, Rakesh Katuwal, Time series classification using diversified Ensemble Deep Random Vector Functional Link and Resnet features, Applied Soft Computing, Volume 112, 2021, 107826,

[2] D. Cimr, F. Studnicka, H. Fujita, H. Tomaskova, R. Cimler, J. Kuhnova, J. Slegr, Computer aided detection of breathing disorder from ballistocardiography signal using convolutional neural network, Inf. Sci. 541 (2020) 207–217.

[3] H.u. Weilong, Y.-W. Si, S. Fong, R.Y.K. Lau, A formal approach to candlestick pattern classification in financial time series, Appl. Soft Comput. 84 (2019) 105700.

[4] X. Wang, Y. Sun, D. Luo, J. Peng, Comparative study of machine learning approaches for predicting short-term photovoltaic power output based on weather type classification, Energy 240 (2022) 122733.

[5] José de Jesús Rubio, Marco Antonio Islas, Genaro Ochoa, David Ricardo Cruz, Enrique Garcia, Jaime Pacheco, Convergent newton method and neural network for the electric energy usage prediction, Inf. Sci. 585 (2022) 89–112, https://doi.org/10.1016/j.ins.2021.11.038.

[6] Y.-H. Lee, C.-P. Wei, T.-H. Cheng, C.-T. Yang, Nearest-neighbor-based approach to time-series classification, Decis. Support Syst. 53 (1) (2012) 207–217.

[7] H. Deng, G. Runger, E. Tuv, M. Vladimir, A time series forest for classification and feature extraction, Inf. Sci. 239 (2013) 142–153.

[8] L. Chen, D. Chen, F. Yang, J. Sun, A deep multi-task representation learning method for time series classification and retrieval, Inf. Sci. 555 (2021) 17–32.

[9] B. Bai, G. Li, S. Wang, W. Zongda, W. Yan, Time series classification based on multi-feature dictionary representation and ensemble learning, Expert Syst. Appl. 169 (2021) 114162.

[10] T. Mao, L. Zhou, Y. Zhang, et al, Classification algorithm for class imbalanced data based on optimized Mahalanobis-Taguchi system, Appl. Intell. (2022), https://doi.org/10.1007/s10489-021-02929-8.

[11] B. Ma, W. Cai, Y. Han, and G. Yu, "A Novel Probability Confidence CNN Model and Its Application in Mechanical Fault Diagnosis," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021, Art no. 3517111.

[12] G. Zhong, Y. Xiao, B. Liu, et al, Pinball loss support vector data description for outlier detection, Appl. Intell. (2022), https://doi.org/10.1007/s10489-022-03237-5.

[13] T. Hayashi, H. Fujita, One-class ensemble classifier for data imbalance problems, Appl. Intell. (2021), https://doi.org/10.1007/s10489-021-02671-1.

[14] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the Support of a High Dimensional Distribution, Neural Comput. 13 (7) (2001) 1443–1471.

[15] M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, In ACM Sigmod Record, 2000.

[16] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, Eighth IEEE International Conference on Data Mining (2008).

[17] S. Khan, A. Ahmad, Relationship between Variants of One-Class Nearest Neighbors and Creating Their Accurate Ensembles, IEEE Trans. Knowl. Data Eng. 30 (09) (2018) 1796–1809.

[18] L. Ruff, et al. Deep one-class classification. In: Proceedings of the 35th International Conference on Machine Learning (PMLR), vol. 80, pp. 4393–4402 (2018).

[19] D. Hendrycks, M. Mazeika, T.G. Dietterich, Deep anomaly detection with outlier exposure, ICLR (2019).

[20] Golan, Izhak, El-Yaniv, Ran. Deep anomaly detection using geometric transformations. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018. p. 9781-9791.

[21] S. Mauceri, J. Sweeney, J. McDermott, Dissimilarity-based representations for one-class classification on time series, Pattern Recogn. 100 (2020) 107122.

[22] A. Blázquez-García, A. Conde, U. Mori, J.A. Lozano, Water leak detection using self-supervised time series classification, Inf. Sci. 574 (2021) 528–541.

[23] T. Hayashi, H. Fujita, A. Hernandez-Matamoros, Less complexity one-class classification approach using construction error of convolutional image transformation network, Inf. Sci. 560 (2021) 217–234.

[24] T. Hayashi, H. Fujita, Experiment of OCITN: Considering Appropriate Goal Images and Metric for One-Class Image Transformation Network, Frontiers in Artificial Intelligence and Applications, Volume 337: New Trends in Intelligent Software Methodologies, Tools and Techniques (2021) 459–472.

[25] M.J. Shensa, The discrete wavelet transform: wedding the a trous and Mallat algorithms, IEEE Trans. Signal Process. 40 (10) (1992) 2464–2482.

[26] I. Sadek, E. Seet, J. Biswas, B. Abdulrazak, M. Mokhtari, Nonintrusive vital signs monitoring for sleep apnea patients: a preliminary study, IEEE Access 6 (2018) 2506–2514.

[27] A. Hernandez-Matamoros, H. Fujita, E. Escamilla-Hernandez, H. Perez-Meana, M. Nakano-Miyatake, Recognition of ECG signals using wavelet based on atomic functions, Biocybernetics and Biomedical Engineering 40 (2) (2020) 803–814.

[28] D. Cimr, F. Studnička, Automatic detection of breathing disorder from ballistocardiography signals, Knowl.-Based Syst. 188 (2020) 104973.

[29] D. Cimr, F. Studnicka, H. Fujita, R. Cimler, J. Slegr, Application of mechanical trigger for unobtrusive detection of respiratory disorders from body recoil micro-movements, Comput. Methods Programs Biomed. 207 (2021), https://doi.org/10.1016/j.cmpb.2021.106149 106149.

[30] M.G. Baydogan, G. Runger, E. Tuv, "A Bag-of-Features Framework to Classify Time Series," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 11, pp. 2796-2802,

[31] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery (DMKD '03). Association for Computing Machinery, New York, NY, USA, 2–11.

[32] R. Sánchez-Reolid, F. López, M.T. de la Rosa, López,, Antonio Fernández-Caballero, One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity, Biomedical Signal Processing and Control, Volume 71, Part B 103203 (2022).

[33] M. Hüsken, P. Stagge, Recurrent neural networks for time series classification, Neurocomputing 50 (2003) 223–235.

[34] Sebamai Parija, Ranjeeta Bisoi, P.K. Dash, Mrutyunjaya Sahani, Deep long short term memory based minimum variance kernel random vector functional link network for epileptic EEG signal classification, Engineering Applications of Artificial Intelligence, Volume 105, 2021, 104426,

[35] Syed Fawad Hussain, Saeed Mian Qaisar, Epileptic seizure classification using level-crossing EEG sampling and ensemble of sub-problems classifier, Expert Syst. Appl. 191 (2022) 116356.

[36] H. Zhu, J. Zhang, H. Cui, K. Wang, Q. Tang, TCRAN: Multivariate time series classification using residual channel attention networks with time correction, Appl. Soft Comput. 114 (2022) 108117.

[37] J.D.J. Rubio, "Stability Analysis of the Modified Levenberg–Marquardt Algorithm for the Artificial Neural Network Training," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 8, pp. 3510-3524, Aug. 2021, doi: 10.1109/TNNLS.2020.3015200.

[38] R. Zhang, X. Bai, L. Pan, et al, Zero-small sample classification method with model structure self-optimization and its application in capability evaluation, Appl. Intell. (2021), https://doi.org/10.1007/s10489-021-02686-8.

[39] Waleed Hilal, S. Andrew Gadsden, John Yawney, Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances, Expert. Syst. Appl. 193 (2022) 116429.

[40] Y. Zhang, B. Zhou, X. Ding, J. Ouyang, X. Cai, J. Gao, X. Yuan, Adversarially learned one-class novelty detection with confidence estimation, Inf. Sci. 552 (2021) 48–64.

[41] C.H. Yang, Yue Lang, Guanghui Yue, Yuan He, One-class classification using generative adversarial networks, IEEE, Access 7 (2019) 37970–37979.

[42] J. Fan, Q. Zhang, J. Zhu, M. Zhang, Z. Yang, H. Cao, Robust deep auto-encoding Gaussian process regression for unsupervised anomaly detection, Neurocomputing 376 (2020) 180–190.

[43] L. Baldacci, M. Golfarelli, D. Lombardi, F. Sami, Natural gas consumption forecasting for anomaly detection, Expert Syst. Appl. 62 (2016) 190–201.

[44] O.T. Inan et al, Ballistocardiography and Seismocardiography: A Review of Recent Advances, IEEE J. Biomed. Health. Inf. 19 (4) (July 2015) 1414–1427, https://doi.org/10.1109/JBHI.2014.2361732.

[45] https://github.com/spaghettix/DB_OC_TSC, Accessed 11 Feb 2022.

[46] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, F. Király (Eds.), SkTime: A Unified InterfAce for MAchine LeArning With Time Series, 2019.

[47] R. Craig Herndon, Determining signal entropy in uncertainty space, Measurement, Volume 178, 2021, 109336,

[48] D.Y. Tsai, Y. Lee, E. Matsuyama, Information entropy measure for evaluation of image quality, J. Digit. Imaging 21 (2008) 338–347.

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

[50] F. Studnicka, Ballistocardiography with breathing disorderes, V3 (2020), https://doi.org/10.17632/9fmfn6kfn7.3.