

ITC 2/53 Information Technology and Control Vol. 53 / No. 2 / 2024 pp. 470-491 DOI 10.5755/j01.itc.53.2.35797	Small Sample Time Series Classification Based on Data Augmentation and Semi-supervised Learning	
	Received 2023/12/05	Accepted after revision 2024/03/20
	HOW TO CITE: Liu, J.-J., Yao, J.-P., Wang, Z., Huang, L. (2024) Small Sample Time Series Classification Based on Data Augmentation and Semi-supervised Learning. <i>Information Technology and Control</i> , 53(2), 470-491. https://doi.org/10.5755/j01.itc.53.2.35797	

Small Sample Time Series Classification Based on Data Augmentation and Semi-supervised Learning

Jing-Jing Liu

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
Key Laboratory of Agricultural Information Acquisition Technology (Beijing), Ministry of Agriculture, Beijing 100083, China

Jie-Peng Yao

Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

Zhuo Wang

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
Key Laboratory of Modern Precision Agricultural System Integration (Beijing), Ministry of Education, Beijing 100083, China

Zhong-Yi Wang

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
Key Laboratory of Modern Precision Agriculture System Integration Research, Ministry of Education, Beijing 100083, China
Key Laboratory of Agricultural Information Acquisition Technology, Ministry of Agriculture, Beijing 100083, China

Lan Huang

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
Key Laboratory of Agricultural Information Acquisition Technology (Beijing), Ministry of Agriculture, Beijing 100083, China

Corresponding author: hlan@cau.edu.cn

Realistic scenarios produce labeled data and unlabeled data, however, there are significant challenges in labeling time series data. It is imperative to effectively integrate the relationship between labeled and unlabeled data within semi-supervised classification model. This paper presents a novel semi-supervised classification method, namely Data Augmentation-Fast Shapelet Semi-Supervised Classification, which employs a data augmentation module to enhance the diversity of data and improve the generalization ability of the model, as well as a feature fusion module to enhance the semi-supervised network. A conditional generative adversarial network is used to synthesize excellent labeled time series samples to enhance the homogeneous data in the sample space, the fast shapelets method is used to quickly extract the important shape feature vectors in the time series, self-supervised and supervised learning are combined to fully learn the unlabeled and labeled data of the time series dataset. The joint loss function combines the loss functions of the two networks to optimize multiple objectives. Reinforcement learning is used to determine the weight coefficients of the joint loss function, at the same time, the reward function is modified to bias the supervisory loss, which improves the classification performance of the model under limited labeled data, and the model can also better achieve the semi-supervised classification task. The proposed method is validated on the UCR benchmark dataset, Electrocardiogram dataset, and Electroencephalogram dataset, the results show that the semi-supervised classification method can perform a more accurate semi-supervised classification of the time series, with an accuracy better than the comparison methods. Meanwhile, we use the plant electrical signal dataset obtained from actual measurements for testing, the visualization analysis can clearly show the model role in the semi-supervised classification task, and the experimental results fully demonstrate the effectiveness and applicability of the proposed method.

KEYWORDS: Small sample time series, Data augmentation, Fast Shapelets, Self-supervised learning, Semi-supervised classification.

1. Introduction

The reality is that time series data exists in several industries, such as in the medical industry with electrocardiographic (ECG) data [30] and electroencephalogram (EEG) data [3]. In the agricultural and biological industries, plant electrical signals data are observed [20], and Internet of Things data are studied [32]. In the industrial sector, there are bearing signal data [26], as well as aerospace signal data [27]. Unauthorized broadcast data identification [48], radio data classification [46, 47] in the communications industry. However, these time series data share common drawbacks: industry specificity limits the availability of large amounts of data; the process of collecting, organizing, and manually labeling large amounts of time series training data is time-consuming and costly, and thus the time series dataset consists of a large amount of unlabeled data with a small amount of labeled data. Classifiers constructed using a small amount of labeled data with a large amount of unlabeled data are called semi-supervised classification models, and using a small amount of labeled time series data for semi-supervised classification tasks is a great challenge in machine learning research [24].

In recent years, deep networks have achieved satisfactory results in semi-supervised classification of

time series data [44, 45]. In particular, deep networks have achieved satisfactory results in semi-supervised classification of time series data [36]. However, semi-supervised classification models for time series also have two obvious shortcomings: first, it is well known that deep neural networks require a large amount of training data, but with the current small amount of time series data and the lack of labeled data, deep networks are prone to overfitting and poor robustness [21]. Spline interpolation and Piecewise Cubic Hermite Interpolating Polynomial interpolation methods, Empirical Mode Decomposition are common time series data enhancement techniques [1, 2]. Inadequately trained semi-supervised classification models are unable to correctly represent the distribution of time series data, which leads to low classification accuracy of semi-supervised models [13]. Second, how semi-supervised classification models can effectively integrate labeled data with unlabeled data to improve model performance [5].

The semi-supervised classification task for small-sample time-series data mainly stems from the high cost of data acquisition in practical applications, the lack of data labels, the pursuit of model generalization ability,

as well as the specificity of time-series data. To solve the problems of semi-supervised classification models for time series data, this paper proposes a new semi-supervised time series classification method Data Augmentation-Fast Shapelet Semi-Supervised Classification (-DA-FSSSC), which firstly uses conditional generative adversarial networks to expand the number of labeled time series, then extracts the important shape features of the time series, finally assists the supervised learning task by self-supervised learning to improve the performance of supervised learning. The main contributions of this study are summarized below:

- 1 Aiming at addressing the problem of unlabeled part of the data in small sample time series data, based on the idea of module integration, this paper can realize the semi-supervised classification of time series under the condition of small samples by combining the data augmentation module, the feature extraction fusion module, and the semi-supervised classification module.
- 2 The feature extraction module can extract important shape features of the time series. The semi-supervised classification module combines the self-supervised learning network for unlabeled data with the supervised learning network for labeled data, and the reinforcement learning can effectively learn the hyper-parameters of the joint loss function, this process can effectively integrate the labeled data with the unlabeled data. A large amount of time series data after data enhancement can effectively train the semi-supervised deep neural network.
- 3 For the method we conducted a series of data experiments, validated on the UCR benchmark dataset, ECG dataset, and EEG dataset, and tested the statistical method for the experimental results on the UCR benchmark dataset, which proved that the classification performance of the new method proposed is superior to the comparative methods. Tests are carried out on real acquired plant electrical signal datasets, and the method is visualized and analyzed to show the effectiveness of the model proposed in this paper in real-life datasets.

2. Related Works

Currently, researchers have proposed some semi-supervised classification models for time series data classification. The semi-supervised time series feature

learning model proposed by Wang, et al. incorporates labeled and unlabeled time series data into an integrated model that efficiently learns through least square minimization, spectral analysis, scaled pseudo labels, and feature similarity regularization terms [35]. Jawed et al. provide a powerful alternative to supervised signals for feature learning by utilizing unlabeled training data through a prediction task, optimizing the multi-tasking learning approach and model prediction as a secondary task along with the primary task of classification, with a model that has better performance [14]. Xi et al. proposed that the past-anchor-future strategy can extract higher-quality semantic context from unlabeled time series data, and that self-supervised temporal relation learning can effectively assist supervised models [38]. Rezaei et al. pre-trained the model on a large unlabeled dataset by inputting the time series features of the sampled packets, and then the learned weights were transferred to a small labeled dataset, which has the same accuracy as a fully supervised method with a large labeled dataset [31]. Goschenhofer et al. show significant performance gains in deep semi-supervised learning models by discussing the transferability of state-of-the-art deep semi-supervised models from image to time-series classification, combined with the use of appropriate model backbone architectures and customized data enhancement strategies [11]. Xi et al. used the lower bound of DTW, LB_Keogh to construct pairwise distance matrices and construct a graph neural network, which is a new graph construction module, and the experimental results showed that it accelerated the network training without decreasing the classification accuracy [39]. Fan et al. proposed a simple and effective semi-supervised time series classification architecture approach, for labeled time series, SemiTime performs supervised classification directly under the supervision of annotated class labels; for unlabeled time series, segments of past-future pairs were sampled from the time series, SemiTime predicts the temporal relationships between these segments in a self-supervised manner, and experimental results showed that SemiTime outperforms state-of-the-art techniques[9]. Wei et al. proposed a multi-task learning scheme for semi-supervised time series classification (MTFC) with time-frequency mining. Unsupervised tasks are used to capture the time-frequency information of the time series, and the multi-task learning framework is used

to learn the common features of the labeled and unlabeled data, and the model can effectively improve the performance of semi-supervised classification [36]. Eldele et al. proposed a novel framework for learning time series representations, which learns representations from unlabeled data through contrast learning, using their views to learn robust temporal relations in the proposed temporal contrast module through weak and strong augmentation specific to the time series, in addition to learning discriminative representations through the contextual contrast module, which has shown to be highly efficient in a few labeled data and in migration learning scenarios [8].

Meanwhile, semi-supervised classification of time series has been applied in practical time series classification scenarios. Wu et al. proposed a semi-supervised fault diagnosis model with an unsupervised autoencoder modified using mean square error, employing labels of the data and utilizing a softmax classifier to directly diagnose the health condition based on the coded features of the autoencoder, which was validated on the electric motor bearings dataset and the industrial hydraulic turbine dataset, and the results showed that the method obtains high diagnostic accuracy [37]. Liu et al. present a framework dedicated to classification (AMC) radio modulation, which achieves higher performance with less labeled data by carefully utilizing unlabeled signal data and a self-supervised comparative learning pre-training step [22]. Han et al. proposed an end-to-end semi-supervised learning framework with two deep neural networks with different backbones and achieved high classification accuracies in a motion picture EEG dataset using contrast learning and adversarial training strategies end-to-end semi-supervised learning framework [12]. Semi-supervised learning allows labeled and unlabeled data to be used efficiently and improves the efficiency of data usage, but the algorithms tend to be more complex and require more tuning and optimization.

There is an even more extreme situation in the real scenario: time series datasets have a limited number of samples, so researchers have proposed some methods for semi-supervised classification methods under the condition of small sample time series data. There is less data available in bearing fault diagnosis, to address this challenge, Yongtao et al. carried out fault feature extraction through variational modal decomposition (VMD) and sample entropy, pre-trained the model

using the feature matrix of unlabeled samples, and utilized the feature matrix and labels of labeled samples to fine-tune the model, and finally achieved fast and accurate bearing fault diagnosis [42]. Ma et al. proposed a Consistent Regularized Auto-Encoder (CRAE) framework based on encoder-decoder networks, which firstly uses data augmentation strategies to process individual process samples into sample matrices, and extracts the local and global spatio-temporal features from the sample matrices by using local encoders and global encoders, and the introduced Consistent Regularization (CR) method pushes the decision boundaries to the low-density region, which makes the distinction between different categories more obvious and improves the accuracy of the model classification [25]. Lao et al. proposed a semi-supervised weighted prototype network (SSWPN), a dual-scale neural network (DSNN) that enhances the ability to extract data features and express different scales, and used a new semi-supervised weighted prototype updating strategy, and the experimental results showed significant advantages in real-world scenarios with scarce data [18]. Zhou et al. proposed the use of Deep Convolutional Generative Adversarial Networks (DCGAN), which overcome the limitations of training data and achieve highly accurate gear diagnosis in the presence of scarce labeled data [49].

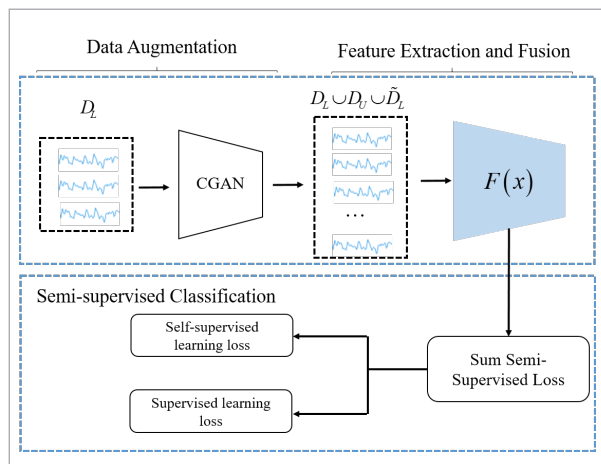
Semi-supervised classification models for small-sample time-series data offer significant advantages in terms of improving data utilization and saving annotation costs, but also face challenges in model design. In practical applications, careful consideration needs to be given to how to balance the advantages and disadvantages of the models, as well as how to design and adapt the models to semi-supervised classification tasks and data characteristics. Thus, we choose to generate additional labeled data, and further explore a semi-supervised classification model more suitable for small sample time series data by exploring the intrinsic connection between labeled and unlabeled data.

3. Methods

In this section, we elaborate on the semi-supervised time series classification method proposed. The semi-supervised classification method consists of three modules: data augmentation module, feature extraction fusion module, and semi-supervised clas-

sification module, where the semi-supervised classification module consists of a self-supervised network suitable for unlabeled data and a supervised network suitable for labeled data. The data augmentation module generates additional synthetic samples to augment the original time series data, followed by extracting the discriminant subsequence from the time series data, self-predictive regression utilizes the historical observations of the time series data to predict its future values, the encoder processes the input sequences through a series of LSTM layers and ultimately outputs a compact representation representing the future values, and the classification task uses multiple LSTM layers and convolutional layers at the same time to construct the feature vectors that ultimately classify the labeled data. The semi-supervised network is capable of extracting global and local features of the time series at multiple scales, thus this semi-supervised classification model can fully learn two different sets of data and improve the performance of the model. The time series data $S = L \cup U = \{(x_{m,j}, y_{m,j})\}_{m=1, j=1}^{M,N} \cup \{(x_{m,j})\}_{m=1, j=1}^{M,N}$ is divided into two groups, one is labeled data D_L and the other is unlabeled data D_U . The sample size of unlabeled time series data is larger and the sample size of labeled data is smaller. These two parts of the dataset are trained using different networks, the loss networks of the two networks are combined and trained by minimizing the joint loss function to train the semi-supervised network to successfully classify the time series data. The overall architecture of the proposed method is displayed in Figure 1.

Figure 1
Flowchart of semi-supervised classification method



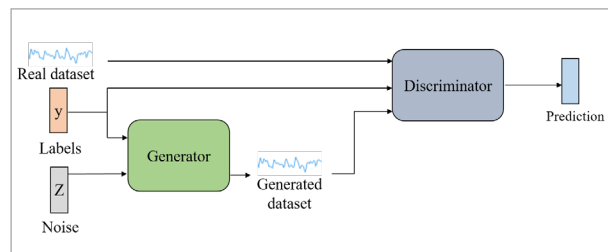
3.1. Data Augmentation Module

Conditional generative adversarial networks (CGAN) add additional auxiliary training information to generative adversarial networks [28]. To better train the network, the conditional generative adversarial network is used to generate multiple types of time series data \tilde{D}_L based on the labeled data D_L in the time series dataset S . We use the category labels of the data as additional information y is combined with a random noise vector Z , which goes from the input layer to the generator and the discriminator. The generator extracts random vectors z from a priori random distributions $p(z)$ and splices the random vectors with extra information y to produce joint hidden representations, which are used by the generator to generate a variety of pseudo-time series data. The discriminator inputs are real-time series data x , or the generator generates fake time series data samples $G(z|y)$ with auxiliary information y . The discriminator can determine whether the time series data samples are real or generated based on the input time series data samples and label condition information. The expression for the objective function of the conditional generative adversarial network is:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}(x)} [\log D(x|y)] + E_{z \sim p_z(z)} [\log (1 - D(G(z|y)))] \quad (1)$$

where G denotes the generative model and D denotes the discriminative model, $p_z(z)$ denotes the prior input noise and y denotes class labels. Conditional generative adversarial networks make full use of the existing labeling information to learn and generate high-quality and diverse time series data more efficiently. Figure 2 illustrates the basic structure of data enhancement using CGAN.

Figure 2
Basic structure of CGAN



3.2. Feature Extraction and Fusion Module

Fast shapelets use a certain method to find out the important points in a time series, then use the sub-segments of the time series that contain one or more important points as a candidate set of shapelets, and then find the best shapelet from these candidate sets, which improves the efficiency of the shapelet searching and saves the running time [29]. Data in the same category in a time series dataset S is a subcategory of the dataset, so there will be a different number of subcategories in different datasets. The distance between the time series is expressed using the distance function $dist(S, D)$, the expression of the distance function for the series of two times is:

$$dist(S, D) = \sqrt{\sum_{i=1}^m (S_i - D_i)^2}, \quad (2)$$

where m denotes the number of time series. The distance between the time series and the subsequence is expressed using the distance function $sdist(S, D)$, and the expression of the distance function between the time series and the subsequence is:

$$sdist(S, T) = \min_{1 \leq p \leq q} dist(T, S_p^k), \quad (3)$$

where the length of the time series T is m , k denotes the length of the subsequence, $q = m - k + 1$. A shapelet is essentially a segment of a time series that maximizes the representation of a class as a subsequence. Using subclass segmentation enables strategic sampling from the training dataset, ensuring that the sampled sequence captures the core of the entire time series. Shapelet Local Farthest Deviation Points (LFDPs) are identified in the sampled time series, and by identifying subsequences between two non-adjacent LFDPs, we can obtain shapelet candidate features with high discriminative power [15]. Therefore, the expression for the transformed feature $S_{transformed}$ of the time series is:

$$S_{transformed} = \{sdist(T_1, S), sdist(T_2, S), \dots, sdist(T_n, S)\}, \quad (4)$$

where n denotes the number of shapelets. The transformed fused feature vectors are then learned using a semi-supervised classification model and the semi-supervised classification model classifies the feature vectors.

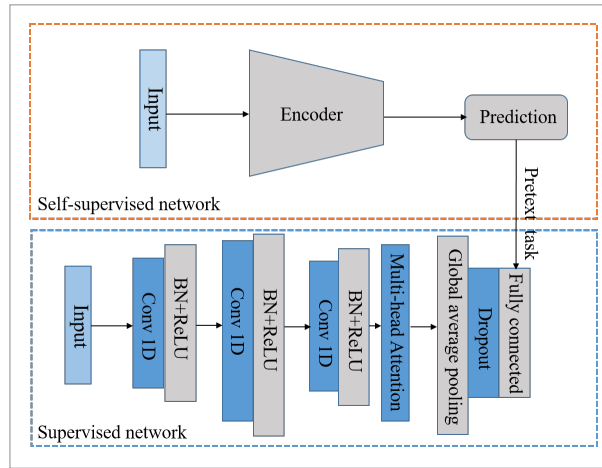
3.3. Semi-supervised Classification Module

In this section, we describe our proposed semi-supervised classification module in detail. As shown in Figure 3, the semi-supervised classification module consists of two networks including a self-supervised network and a supervised network. For unlabeled time series datasets $x_i \in D_u$, a self-supervised learning method is used for training and the self-prediction regression task is performed on unlabeled time series datasets. Self-prediction as an auxiliary task can help models learn rich hidden state representations from unlabeled but structured time series data [14]. Self-supervised networks are encoder architectures based on Long Short-Term Memory (LSTM) networks that are capable of learning effective feature representations from input time-series data. The encoder network consists of multiple LSTM layers, each of which is an LSTM module, and each LSTM module is set up with two LSTM layers to enhance the learning capability of the model. The hidden layer activation function and the output layer activation function are used to introduce nonlinear transformations between the hidden layers and at the final output, respectively, to increase the expressive power of the model. The loss function expression for self-supervised network training is:

$$L_{ss} = \frac{1}{|D_u|} \sum_{i=1}^{|D_u|} (y_i - \hat{y}_i)^2, \quad (5)$$

where D_u denotes the number of unlabeled samples of the time series, y_i denotes the true value of the data series data, \hat{y}_i denotes the predicted value of the network. For labeled time series datasets $x_i \in D_l$, a supervised learning approach is used for training to perform classification tasks on time series datasets. Supervised networks are classification networks that combine deep convolutional networks and self-attention mechanisms to capture both local features and long-range dependencies of time series data. The stacking of multiple convolutional layers enables the network to learn complex feature representations of time series, the self-attention mechanism to capture long-range dependencies of time series data. The global average pooling layer and the dropout layer reduce the number of parameters, thus reducing the risk of over-network fitting and improving the computational efficiency of the network. The computational efficiency of the model is optimized using the Dynamic Quantization meth-

Figure 3
Architecture of Semi-supervised classification module



od, which accelerates computation and reduces energy consumption. The fully connected layer categorizes the data features. The loss function expression for supervised network training is:

$$L_s = -\frac{1}{|D_L|} \sum_{i=1}^{|D_L|} y_i * \log(p_i), \quad (6)$$

where D_L denotes the number of time series labeled samples, y denotes the true label of the dataset, p_i denotes the result of the probability value predicted by the classification model.

Semi-supervised classification models include both self-supervised and supervised networks, the self-supervised task is the auxiliary task and the supervised task is the main task, so the objective function is the weighted sum of the two loss functions. The self-supervised task is the auxiliary task and the supervised task is divided into the main task. By optimizing the joint loss function, the method can effectively train all the task modules and improve the generalization ability of the semi-supervised classification model. The loss function expression for the semi-supervised classification module is:

$$L = \lambda L_s + (1 - \lambda) L_{ss}, \quad (7)$$

where λ is a hyper-parameter. The correct λ neither biases the supervised network weights nor ignores the learning task of the self-supervised network [16]. Reinforcement learning algorithms in which individual intelligentsia are used for parameter updating,

dynamically updating the weights and significantly improving the algorithm convergence efficiency and performance [43]. We use reinforcement learning to dynamically adjust the weight coefficients λ in the loss function [23]. The steps of weight updating based on reinforcement learning are as follows:

Step 1: Agent Initialization. When an agent is initialized, a Q-table is created and a state matrix $s = [\alpha_1, \alpha_2]$ and an action matrix $a = [\Delta\alpha_1, \Delta\alpha_2]$ are constructed. α_1 is the weight of the supervisory network, α_2 is the weight of the self-supervisory network, $\Delta\alpha_i$ denotes the change in weight. The state matrix is the current weights of the self-supervised network and the supervised network, and the action matrix is the adjustment of the weights, which represents an increase of 0.1 or a decrease of 0.1 in the value.

Step 2: Construct the reward function R , the loss function L and the evaluation function Q . The reward function indicates that rewards are given based on improvements in accuracy, thus encouraging actions that will improve the performance of the model.

Step 3: Q-table state update. Using the learning rate and discount factor, the Q-value corresponding to the current state and action is adjusted according to the update rules of the Q-learning algorithm. The agent updates the Q-table based on the reward received and the maximum expected Q-value of the next state. The agent all updates its state based on the rewards it receives α , with the aim of finding the optimal balance between supervised and unsupervised loss and achieving the highest accuracy.

$$Q(s, a) \leftarrow Q(s, a) + l \cdot (r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a)), \quad (8)$$

where $Q(s, a)$ denotes the Q value of the action a taken in the current state s , l denotes the learning rate, r denotes the reward R obtained after the execution of the current action a , γ denotes the discount factor, which is the degree of importance attached to the future reward, and $\max_{a'} Q(s', a')$ denotes the maximum Q value of all the possible actions in the next state s' , which denotes the maximum reward expected from the next state.

Step 4: Loop training. Repeat steps 2 and 3 to calculate the accuracy using the model performance on the validation set and update the α value via Agent accordingly.

Step 5: Depending on the action chosen by the Agent and the reward received, the α value is updated to guide the training of the model in the next loop.

Step 6: When both the model and the Agent reach a certain level of optimization, the performance of the model on the test set is used to evaluate the effectiveness of the whole training process. The optimal α value is obtained, and ultimately the optimal classification accuracy is obtained.

After data augmentation, 1000 samples are added to each category in the time series dataset. The feature vectors are obtained, and we perform semi-supervised classification on the time series dataset. During semi-supervised classification model training, the learning rate is set to 0.001, the batch size is set to 128, and the training period is set to 1000. Meanwhile, the model uses two optimizers, Adam and SGD, to take advantage of Adam fast convergence, and then takes advantage of SGD generalization performance by using the Adam optimizer in training and switching the optimizer to SGD when it reaches the specified number of training times. All methods in this manuscript were used on a Windows 10 PC with a 13th Gen Intel(R) Core(TM) i9-13900KF 3.00 GHz CPU, 64.00 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU. The

Python version uses Version 3.8.10, PyTorch version uses version 2.0.1.

4. Results

4.1. UCR Time Series Benchmark Dataset

We conducted a validation process by selecting small datasets from the UCR benchmark dataset, UCR dataset is a well-established dataset for time series analysis. The training and testing of our method were performed using the datasets specified in the UCR time series [6]. The primary criteria for the dataset were that the sample size was less than 1000, the dataset consisted of several different time series lengths, and contained both binary and multi-categorical datasets. Additionally, In order to ensure the reliability of the experimental results and the validity of the comparisons, we employed three different ratios of labeled dataset to unlabeled dataset, namely 1:9, 3:7, and 5:5. The basic information of the UCR time series is presented in Table 1, where instance represents the num-

Table 1

Basic information of the UCR benchmark time series dataset

Datasets	Instance	Length	Class
Beef	30	470	5
BeetleFly	20	512	2
BirdChicken	20	512	2
CBF	900	30	3
ChlorineConcentration	40	166	3
Coffee	28	286	2
CricketX	390	300	12
CricketY	390	300	12
CricketZ	390	300	12
DistalPhalanxOutlineAgeGroup	400	80	3
DistalPhalanxOutlineCorrect	600	80	2
DistalPhalanxTW	400	80	6
ECG200	100	96	2
ECG5000	500	140	5
ECGFiveDays	23	136	2
FaceAll	560	131	14
FreezerRegularTrain	150	301	2
HouseTwenty	40	2000	2
InsectWingbeatSound	220	256	11
Meat	60	448	3

Datasets	Instance	Length	Class
MedicalImages	381	99	10
MiddlePhalanxOutlineAgeGroup	600	80	3
MiddlePhalanxOutlineCorrect	400	80	3
MiddlePhalanxTW	399	80	6
MoteStrain	20	84	2
PhalangesOutlinesCorrect	600	291	2
ProximalPhalanxOutlineAgeGroup	400	80	3
ProximalPhalanxOutlineCorrect	600	80	2
ProximalPhalanxTW	400	80	6
ShapeletSim	20	500	2
SonyAIBORobotSurface1	20	70	2
SonyAIBORobotSurface2	27	65	2
Strawberry	613	235	2
SwedishLeaf	500	128	15
SyntheticControl	300	60	6
ToeSegmentation1	40	277	2
ToeSegmentation2	36	343	2
TwoLeadECG	23	82	2
UMD	36	150	3
Wine	57	234	2

ber of samples in the dataset, all the series data are of equal length, length represents the length of the dataset, and class represents the category of the dataset.

4.1.1. Methods of Comparison

This paper compares several semi-supervised classification methods for time series. Label Propagation propagates the labels of labeled data to unlabeled data, thus realizing the classification of labeled data [33]. The self-training method utilizes the model to make predictions on unlabeled data and uses these self-generated predictions to augment the training of the model, for comparative use, KNN is used as the base classifier in this paper, and good classification accuracies are obtained through multiple iterations [4]. Pseudo Label helps the model to learn better from unlabeled information, and our semi-supervised classification model based on Pseudo Label implemented by the underlying LSTM network [19]. Semi-supervised Time Series Classification (MTL) model utilizes features learned from self-supervised tasks with unlabeled data while drawing on established multi-task learning methods and model predictions as auxiliary tasks to be optimized along with the primary task of classification [14]. A semi-supervised time series classification model for self-supervised learning (SSTSC), which uses self-supervised learning as a secondary task that is co-optimized with the primary Time Series Classification (TSC) task [38]. In this paper, these five methods are fully compared with the DA-FSSSC method to verify the validity of the present method.

4.1.2. Performance Evaluation

For the semi-supervised classification problem, this paper chooses the classical classification accuracy rate as the evaluation index. The accuracy rate refers to the percentage of the total samples that the model predicts correctly.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (9)$$

where TP denotes the number of positive class samples that the model correctly predicts as positive, TN denotes the number of negative class samples that the model correctly predicts as negative, FP denotes the number of negative class samples that the model incorrectly predicts as positive, and FN denotes the number of positive class samples that the model incorrectly predicts as negative.

4.1.3. Analysis of Results

The DA-FSSSC method is implemented in Python using the Pytorch library. Before training, we normalize all datasets. Table 2 demonstrates the results of the classification comparison between the DA-FSSSC method and the comparison method. Classification accuracy is the average of the results from five runs of the model. The best results are shown in black and bold.

Based on the determined small sample dataset, the comparison of methods is carried out, the results are shown in Table 2. Compared with the comparison methods, the proposed method in this paper obtains the highest accuracy on most of the datasets, which indicates that the semi-supervised classification method in this paper performs better, and the advantage outperforms the other methods. The label propagation method for semi-supervised classification of small-sample time-series data can effectively utilize the available labels and is highly scalable, but the method is difficult to capture long-term dependencies in time-series data. Self-training algorithms can utilize unlabeled data to improve model performance in semi-supervised classification tasks using small samples of temporal data. However, the effectiveness of these algorithms is compromised by error propagation and the inherent complexity of temporal data. Pseudo-labeling techniques can be adapted to time-series features and are a simple and versatile semi-supervised technique. The effectiveness of the algorithm depends heavily on the quality of the initial model, and thus requires robust initial model training as well as strategies to limit the impact of erroneous pseudo-labeling. The MTL algorithm significantly outperforms state-of-the-art baseline algorithms in a semi-supervised setting by means of a ConvNet model that jointly performs classification and auxiliary prediction, but the method requires careful tuning of the hyper-parameters, and in the future more methods incorporating consistency regularization will need to be explored to improve performance. SSTSC improves the performance of the classification task by exploiting the semantic context in unlabeled data. DA-FSSSC can effectively expand the dataset by generating additional training samples to improve the diversity of data and alleviate the overfitting problem caused by small-sample data. Meaningful shape features extracted from time-series data can improve

Table2

Performance comparison in UCR benchmark time series dataset

Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio
		90%	70%	50%
Beef	Label Propagation	0.300	0.400	0.667
	Self-Training	0.267	0.300	0.433
	Pseudo Label	0.300	0.333	0.363
	MTL	0.500	0.750	0.830
	SSTSC	0.500	0.550	0.640
	DA-FSSSC	0.600	0.780	0.860
BeetleFly	Label Propagation	0.600	0.610	0.650
	Self-Training	0.500	0.550	0.700
	Pseudo Label	0.400	0.500	0.530
	MTL	0.625	0.775	0.875
	SSTSC	0.500	0.675	0.750
	DA-FSSSC	0.650	0.780	0.880
BirdChicken	Label Propagation	0.500	0.510	0.520
	Self-Training	0.400	0.420	0.510
	Pseudo Label	0.500	0.510	0.525
	MTL	0.750	0.805	0.875
	SSTSC	0.500	0.750	0.875
	DA-FSSSC	0.650	0.755	0.800
CBF	Label Propagation	0.449	0.592	0.599
	Self-Training	0.331	0.683	0.817
	Pseudo Label	0.395	0.575	0.618
	MTL	0.962	0.978	0.995
	SSTSC	0.899	0.925	0.998
	DA-FSSSC	0.893	0.894	0.976
ChlorineConcentration	Label Propagation	0.435	0.465	0.512
	Self-Training	0.462	0.496	0.514
	Pseudo Label	0.532	0.532	0.543
	MTL	0.597	0.702	0.816
	SSTSC	0.527	0.779	0.854
	DA-FSSSC	0.616	0.806	0.897
Coffee	Label Propagation	0.914	0.934	0.964
	Self-Training	0.893	0.929	0.929
	Pseudo Label	0.464	0.464	0.535
	MTL	0.833	0.917	0.927
	SSTSC	0.700	0.979	0.996
	DA-FSSSC	0.793	0.864	0.944
CricketX	Label Propagation	0.277	0.377	0.436
	Self-Training	0.231	0.323	0.408
	Pseudo Label	0.220	0.310	0.433
	MTL	0.663	0.718	0.776
	SSTSC	0.513	0.653	0.674
	DA-FSSSC	0.795	0.828	0.899
CricketY	Label Propagation	0.341	0.449	0.503
	Self-Training	0.395	0.415	0.474
	Pseudo Label	0.297	0.315	0.405
	MTL	0.677	0.737	0.756
	SSTSC	0.449	0.551	0.696
	DA-FSSSC	0.815	0.895	0.941
CricketZ	Label Propagation	0.226	0.362	0.423
	Self-Training	0.233	0.362	0.464
	Pseudo Label	0.205	0.366	0.451
	MTL	0.577	0.737	0.756
	SSTSC	0.474	0.570	0.609
	DA-FSSSC	0.774	0.897	0.907
DistalPhalanxOutlineAgeGroup	Label Propagation	0.691	0.683	0.705
	Self-Training	0.669	0.676	0.676
	Pseudo Label	0.467	0.683	0.683
	MTL	0.641	0.759	0.806
	SSTSC	0.775	0.794	0.803
	DA-FSSSC	0.777	0.791	0.884

Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio	Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio
		90%	70%	50%			90%	70%	50%
DistalPhalanxOutlineCorrect	Label Propagation	0.656	0.710	0.717	FaceAll	Label Propagation	0.432	0.567	0.612
	Self-Training	0.714	0.743	0.746		Self-Training	0.326	0.581	0.653
	Pseudo Label	0.583	0.681	0.731		Pseudo Label	0.437	0.528	0.740
	MTL	0.739	0.773	0.807		MTL	0.740	0.771	0.787
	SSTSC	0.672	0.724	0.770		SSTSC	0.751	0.799	0.865
	DA-FSSSC	0.623	0.656	0.736		DA-FSSSC	0.778	0.800	0.898
DistalPhalanxTW	Label Propagation	0.439	0.612	0.619	FreezerRegularTrain	Label Propagation	0.573	0.724	0.767
	Self-Training	0.612	0.619	0.633		Self-Training	0.589	0.709	0.755
	Pseudo Label	0.302	0.582	0.676		Pseudo Label	0.502	0.757	0.758
	MTL	0.806	0.815	0.825		MTL	0.785	0.890	0.893
	SSTSC	0.651	0.717	0.764		SSTSC	0.861	0.906	0.928
	DA-FSSSC	0.842	0.921	0.906		DA-FSSSC	0.845	0.889	0.936
ECG200	Label Propagation	0.630	0.636	0.655	HouseTwenty	Label Propagation	0.420	0.555	0.639
	Self-Training	0.630	0.647	0.666		Self-Training	0.361	0.672	0.706
	Pseudo Label	0.640	0.654	0.670		Pseudo Label	0.420	0.420	0.579
	MTL	0.800	0.900	0.910		MTL	0.706	0.738	0.769
	SSTSC	0.789	0.795	0.821		SSTSC	0.767	0.808	0.936
	DA-FSSSC	0.730	0.850	0.960		DA-FSSSC	0.743	0.846	0.884
ECG5000	Label Propagation	0.927	0.936	0.938	InsectWingbeatSound	Label Propagation	0.318	0.495	0.583
	Self-Training	0.925	0.926	0.926		Self-Training	0.304	0.422	0.557
	Pseudo Label	0.886	0.923	0.927		Pseudo Label	0.290	0.290	0.324
	MTL	0.940	0.953	0.955		MTL	0.398	0.459	0.536
	SSTSC	0.964	0.975	0.993		SSTSC	0.438	0.527	0.575
	DA-FSSSC	0.844	0.869	0.965		DA-FSSSC	0.715	0.741	0.881
ECGFiveDays	Label Propagation	0.481	0.561	0.534	Meat	Label Propagation	0.633	0.717	0.827
	Self-Training	0.482	0.498	0.642		Self-Training	0.633	0.663	0.850
	Pseudo Label	0.497	0.499	0.502		Pseudo Label	0.333	0.343	0.533
	MTL	0.955	0.966	0.994		MTL	0.567	0.792	0.833
	SSTSC	0.820	0.900	0.995		SSTSC	0.625	0.647	0.700
	DA-FSSSC	0.859	0.882	0.949		DA-FSSSC	0.667	0.851	0.913

Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio
		90%	70%	50%
MedicalImages	Label Propagation	0.489	0.572	0.604
	Self-Training	0.478	0.617	0.661
	Pseudo Label	0.496	0.582	0.623
	MTL	0.699	0.738	0.803
	SSTSC	0.550	0.580	0.630
	DA-FSSSC	0.799	0.842	0.896
MiddlePhalanxOutlineAgeGroup	Label Propagation	0.571	0.597	0.623
	Self-Training	0.487	0.591	0.591
	Pseudo Label	0.388	0.616	0.616
	MTL	0.685	0.694	0.766
	SSTSC	0.645	0.709	0.736
	DA-FSSSC	0.695	0.779	0.818
MiddlePhalanxOutlineCorrect	Label Propagation	0.677	0.746	0.766
	Self-Training	0.674	0.732	0.749
	Pseudo Label	0.570	0.575	0.580
	MTL	0.771	0.782	0.810
	SSTSC	0.746	0.768	0.808
	DA-FSSSC	0.767	0.802	0.881
MiddlePhalanxTW	Label Propagation	0.439	0.612	0.619
	Self-Training	0.552	0.604	0.604
	Pseudo Label	0.272	0.272	0.558
	MTL	0.604	0.622	0.649
	SSTSC	0.509	0.659	0.792
	DA-FSSSC	0.794	0.868	0.909
MoteStrain	Label Propagation	0.831	0.836	0.852
	Self-Training	0.539	0.827	0.828
	Pseudo Label	0.554	0.559	0.585
	MTL	0.833	0.946	0.970
	SSTSC	0.894	0.971	0.987
	DA-FSSSC	0.798	0.804	0.886

Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio
		90%	70%	50%
PhalangesOutlinesCorrect	Label Propagation	0.643	0.731	0.740
	Self-Training	0.785	0.800	0.810
	Pseudo Label	0.613	0.624	0.630
	MTL	0.778	0.820	0.835
	SSTSC	0.764	0.787	0.826
	DA-FSSSC	0.797	0.838	0.922
ProximalPhalanxOutlineAgeGroup	Label Propagation	0.776	0.844	0.878
	Self-Training	0.785	0.800	0.800
	Pseudo Label	0.487	0.487	0.853
	MTL	0.768	0.778	0.876
	SSTSC	0.800	0.849	0.884
	DA-FSSSC	0.820	0.853	0.886
ProximalPhalanxOutlineCorrect	Label Propagation	0.725	0.808	0.825
	Self-Training	0.722	0.804	0.832
	Pseudo Label	0.683	0.683	0.683
	MTL	0.821	0.872	0.919
	SSTSC	0.849	0.893	0.946
	DA-FSSSC	0.832	0.881	0.895
ProximalPhalanxTW	Label Propagation	0.732	0.766	0.771
	Self-Training	0.566	0.717	0.727
	Pseudo Label	0.351	0.351	0.682
	MTL	0.716	0.777	0.793
	SSTSC	0.765	0.782	0.798
	DA-FSSSC	0.824	0.905	0.922
ShapeletSim	Label Propagation	0.478	0.500	0.513
	Self-Training	0.467	0.478	0.500
	Pseudo Label	0.500	0.510	0.516
	MTL	0.600	0.825	0.925
	SSTSC	0.725	0.802	0.950
	DA-FSSSC	0.683	0.733	0.844

Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio	Dataset	Methods	Unlabeled ratio	Unlabeled ratio	Unlabeled ratio
		90%	70%	50%			90%	70%	50%
SonyAIBORobotSurface1	Label Propagation	0.424	0.433	0.433	ToeSegmentation1	Label Propagation	0.500	0.513	0.548
	Self-Training	0.429	0.639	0.684		Self-Training	0.522	0.566	0.605
	Pseudo Label	0.428	0.429	0.440		Pseudo Label	0.473	0.568	0.573
	MTL	0.792	0.784	0.799		MTL	0.696	0.844	0.881
	SSTSC	0.868	0.884	0.900		SSTSC	0.794	0.862	0.880
	DA-FSSSC	0.886	0.895	0.905		DA-FSSSC	0.713	0.865	0.905
SonyAIBORobotSurface2	Label Propagation	0.611	0.612	0.711	ToeSegmentation2	Label Propagation	0.384	0.600	0.615
	Self-Training	0.683	0.778	0.817		Self-Training	0.792	0.815	0.869
	Pseudo Label	0.617	0.617	0.639		Pseudo Label	0.385	0.434	0.554
	MTL	0.706	0.794	0.853		MTL	0.600	0.794	0.853
	SSTSC	0.849	0.874	0.969		SSTSC	0.706	0.758	0.837
	DA-FSSSC	0.752	0.839	0.879		DA-FSSSC	0.685	0.769	0.877
Strawberry	Label Propagation	0.819	0.889	0.895	TwoLeadECG	Label Propagation	0.531	0.571	0.575
	Self-Training	0.870	0.892	0.903		Self-Training	0.500	0.539	0.605
	Pseudo Label	0.552	0.640	0.643		Pseudo Label	0.499	0.509	0.556
	MTL	0.904	0.934	0.959		MTL	0.781	0.874	0.883
	SSTSC	0.863	0.954	0.974		SSTSC	0.787	0.830	0.887
	DA-FSSSC	0.876	0.896	0.922		DA-FSSSC	0.766	0.876	0.943
SwedishLeaf	Label Propagation	0.504	0.608	0.650	UMD	Label Propagation	0.493	0.514	0.556
	Self-Training	0.470	0.634	0.698		Self-Training	0.312	0.493	0.646
	Pseudo Label	0.136	0.552	0.764		Pseudo Label	0.330	0.340	0.500
	MTL	0.836	0.933	0.978		MTL	0.806	0.817	0.844
	SSTSC	0.644	0.769	0.862		SSTSC	0.810	0.820	0.855
	DA-FSSSC	0.923	0.963	0.997		DA-FSSSC	0.826	0.833	0.871
SyntheticControl	Label Propagation	0.630	0.807	0.830	Wine	Label Propagation	0.481	0.648	0.759
	Self-Training	0.630	0.773	0.873		Self-Training	0.500	0.530	0.704
	Pseudo Label	0.640	0.983	0.983		Pseudo Label	0.500	0.511	0.600
	MTL	0.942	0.953	0.983		MTL	0.518	0.652	0.696
	SSTSC	0.948	0.961	0.977		SSTSC	0.523	0.538	0.574
	DA-FSSSC	0.949	0.967	0.998		DA-FSSSC	0.689	0.744	0.826

the accuracy and efficiency of classification, and thus the method can enable the model to maintain a high degree of flexibility and accuracy when dealing with complex time-series data. The comparison results can also show the advantage of the integrated model over a single semi-supervised model in that it can effectively utilize the advantages of each component to improve the classification performance.

4.1.4. Statistical Analysis

The Friedman Test and post-hoc Nemenyi Test can benchmark algorithms based on their rankings on different datasets, comparing the performance of multiple algorithms on multiple datasets [7]. The algorithms were first ranked and the average ranking was calculated, then Friedman test was utilized to determine if there was a significant difference in the rankings of the algorithms. If the Friedman test shows that there was a significant difference, pairwise differences between the algorithms were further investigated using the post-hoc test, which yields a Critical Difference (CD) value. Figure 4 illustrates the results of statistical tests for different proportions of unlabeled datasets, where the more rightward the position of the model on the axes, the higher the classification accuracy of the model.

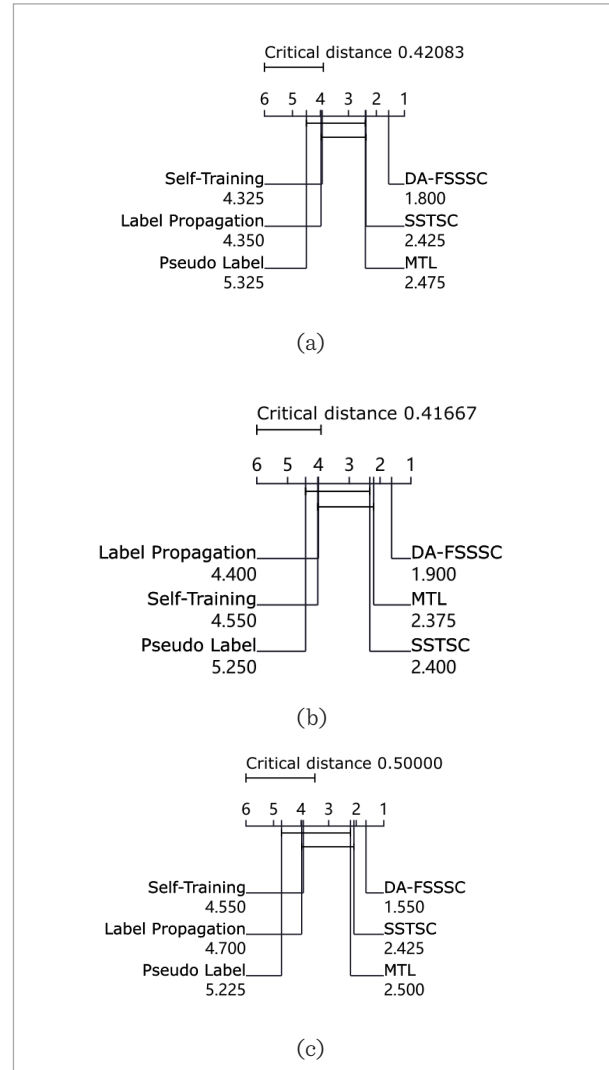
Figure 4 illustrates the results of the statistical analysis, where we rejected the original hypothesis because the p value was much smaller than the significance level, indicating that at least two of the five models were statistically significantly different. The results in Figure 4 are all calculated at a significance level of 0.05. According to Figure 4, it can be clearly seen that the semi-supervised classification method proposed in this paper performs the best among the six methods when the unlabeled percentage of the dataset is 90%, 70%, and 50. The integrated methods MTL, SSTSC, and DA-FSSSC are located on the right hand side of the graphical axes, so it can be demonstrated that the integrated methods outperform the single methods in semi-supervised classification of small sample datasets of time series.

4.2. ECG Dataset

Semi-supervised classification of ECG data has a very important role in the medical industry as classification models can identify previously undetected arrhythmia patterns or subtle changes [17]. Semi-su-

Figure 4

Results of statistical analysis, (a) Rate of unlabeled dataset is 90%, (b) Rate of unlabeled dataset is 70%, (c) Rate of unlabeled dataset is 50%

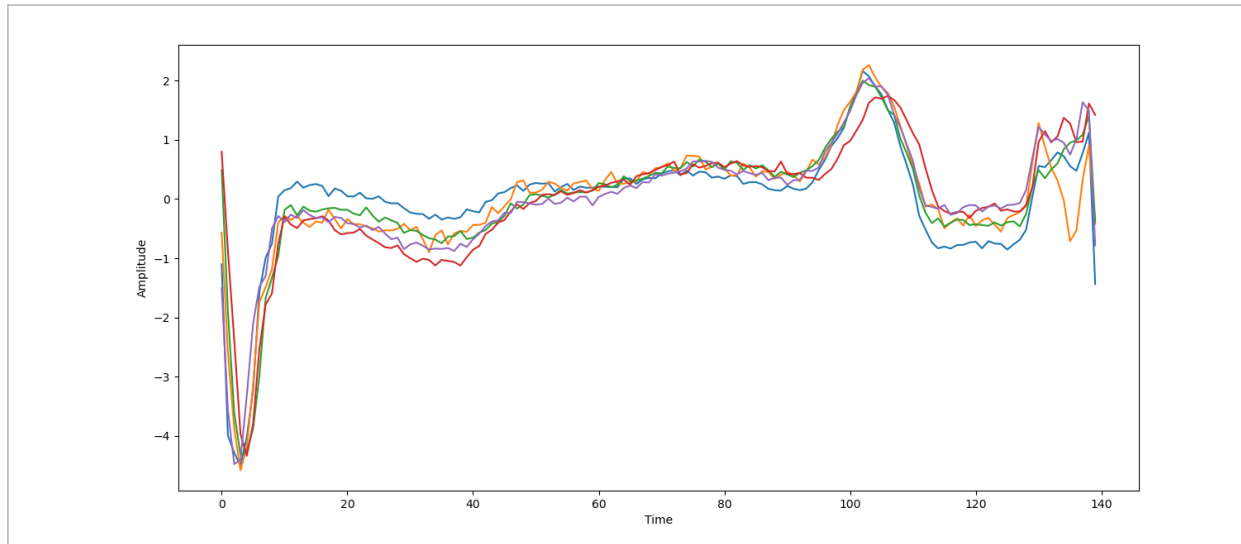


pervised learning can reduce the need for expert labeling of ECG datasets, thereby reducing the costs associated with data collection and labeling. In this paper, we choose 1000 samples for the training set and 200 samples for the test set of ECG data. The goal of semi-supervised classification methods is to be able to recognize different types of ECG data. The ECG dataset is shown in Figure 5.

We test the ECG dataset using five comparative methods and the method proposed in this paper and sim-

Figure 5

Display of ECG data samples

**Table 3**

Classification results in ECG dataset

Unlabeled ratio	Label Propagation	Self-Training	Pseudo Label	MTL	SSTSC	DA-FSSSC
90%	0.790	0.790	0.670	0.755	0.800	0.805
70%	0.800	0.820	0.775	0.797	0.845	0.855
50%	0.890	0.867	0.800	0.880	0.899	0.905

ulate three different unlabeled datasets of 90%, 70%, and 50%. The accuracy test results of the method are shown in Table 3.

According to the comparison results in Table 3, when the percentage of unlabeled datasets is higher than 90%, the accuracy of the semi-supervised classification method proposed in this paper is higher than 0.800 and higher than all the compared methods. When the percentage of unlabeled dataset is 50%, the classification accuracy of the proposed method in this paper is higher than 0.900. When the percentage of unlabeled datasets is 70%, the classification accuracy of the proposed method in this paper is 0.855, and the accuracies of label propagation, self-training, and SSTSC are higher than 0.800. When the percentage of unlabeled dataset is 50%, the classification accuracy of the proposed method in this paper is higher than 0.900. Semi-supervised learning reduces the need for large amounts of labeled data, which reduces the cost

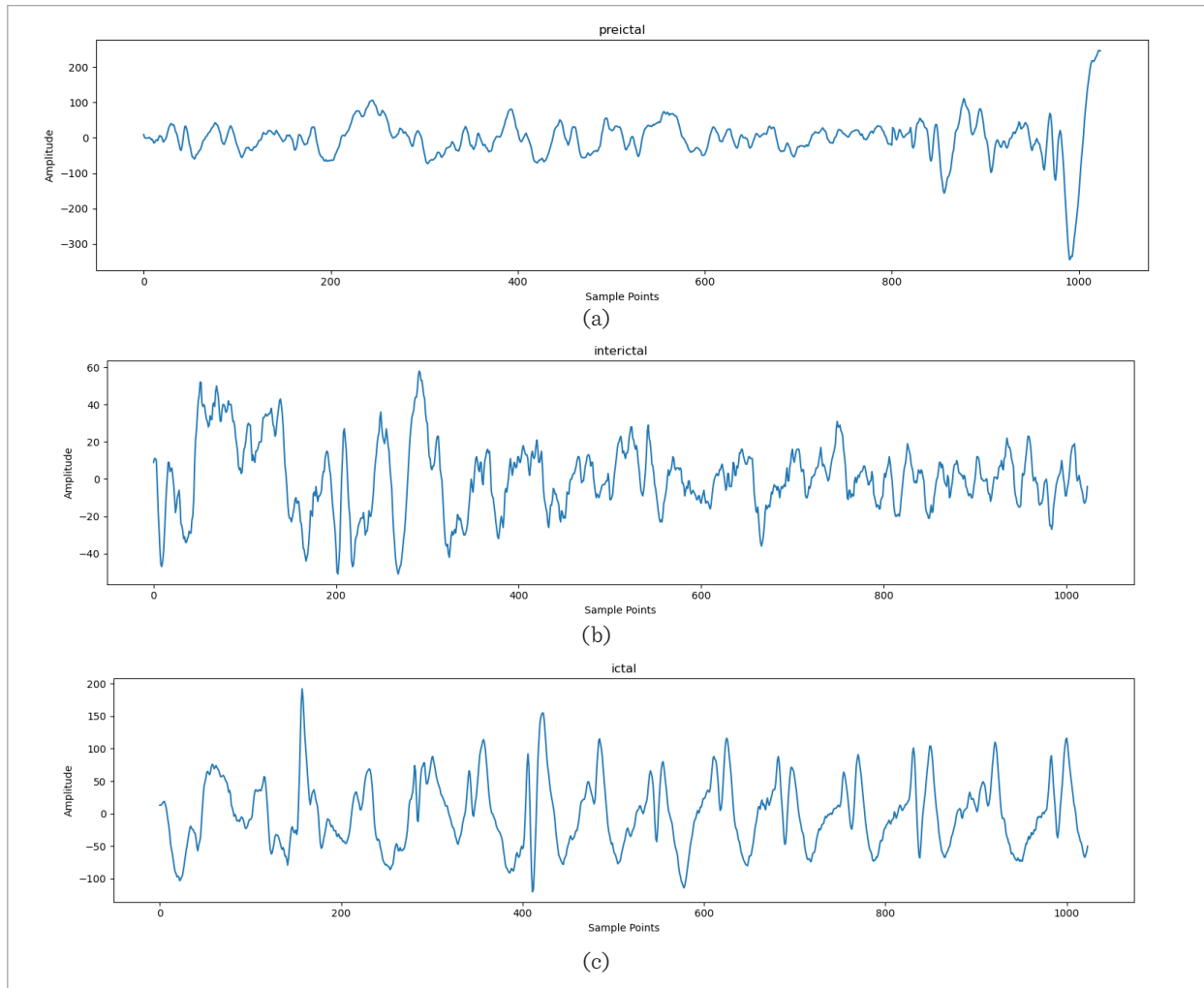
and time of data preparation, and achieving higher accuracy on semi-supervised classification tasks for ECG data is important for early diagnosis and treatment of cardiac diseases.

4.3. EEG Dataset

In reality, it is relatively easy to obtain unlabeled EEG data, but obtaining expert manually labeled data is very difficult and expensive. Unlabeled EEG data may be overlooked or underutilized in many studies, and semi-supervised learning makes these data a valuable resource to help improve their utilization [40]. The dataset used in this paper is exemplary segmented EEG time series recordings of ten epilepsy patients collected from the Neurology and Sleep Center, Hauz Khas, New Delhi [34]. We randomly assigned 80% of the data in each class to the training set and the remaining 20% to the test set. The goal of the semi-supervised classification meth-

Figure 6

Presentation of a sample of EEG data, (a) preictal, (b) interictal, (c) ictal



od is to be able to recognize the seizure phases of epilepsy, which are preictal, interictal, and ictal. The EEG dataset is shown in Figure 6.

We test the EEG dataset using five comparative methods and the method proposed in this paper and sim-

ulate three different unlabeled percentages of 90%, 70%, and 50%. The accuracy test results of the methods are shown in Table 4.

According to the comparison results in Table 4, when the percentage of unlabeled dataset is higher than

Table 4

Classification results in EEG dataset

Unlabeled ratio	Label Propagation	Self-Training	Pseudo Label	MTL	SSTSC	DA-FSSSC
90%	0.533	0.435	0.533	0.600	0.700	0.715
70%	0.567	0.555	0.639	0.852	0.875	0.867
50%	0.633	0.607	0.673	0.903	0.910	0.913

90%, the accuracy of the semi-supervised classification method proposed in this paper is higher than 0.700, with an accuracy of 0.715. When the percentage of unlabeled dataset is 70%, the classification accuracy of SSTSC is highest, and the semi-supervised classification method proposed in this paper ranks second. When the percentage of unlabeled dataset is 50%, the classification accuracy of the proposed method is higher than 0.900. Also, the classification accuracy of MTL and SSTSC is higher than 0.900, indicating that the classification accuracy of the integrated model is higher than that of the single model. Semi-supervised classification of EEG epilepsy data improves the efficiency and accuracy of epilepsy monitoring and diagnosis, reduces healthcare costs, and is important for early identification of epilepsy patients and prediction of seizures.

4.4. Plant Electrical Signal Datasets

When plants encounter stimuli, they rapidly send electrical signals to different parts of the plant and even to the whole plant, plant electrical signals are a

record of electrical activity in plant cells and tissues [20]. Plant electrical signals can more quickly reflect external stimuli received by the plant. In the absence of labels or with only a small amount of labeled data, semi-supervised learning helps to identify and interpret patterns of plant electrical signals, enabling faster understanding and prediction of plant responses to environmental changes. In this paper, we utilize plant electrical signal data obtained from actual laboratory measurements, selected from 300 ml of NaCl concentration stimulation, to identify the salt tolerance of wheat [41]. The goal of the semi-supervised classification method is to be able to correctly identify the signaling dynamics of wheat leaves DeKang961 (salt-tolerant) and Langdon (salt-sensitive) under continuous stress of NaCl concentration. The Plant electrical signals dataset is shown in Figure 7.

We test the Plant electrical signal dataset using five comparative methods and the method proposed in this paper and simulated three different unlabeled ratios of 90%, 70%, and 50%. The accuracy test results of the methods are shown in Table 5.

Figure 7

Sample display of plant electrical dataset

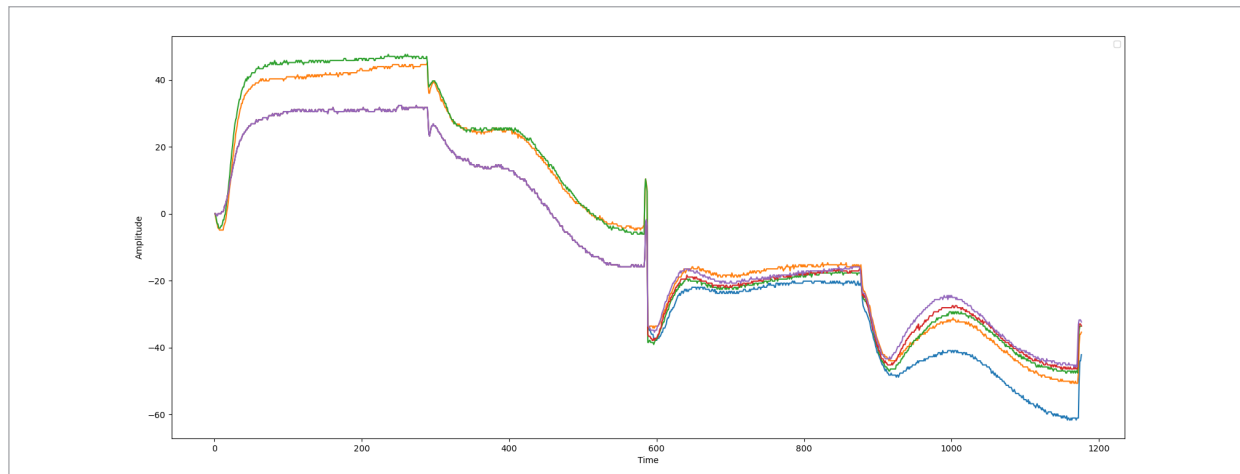


Table 5

Classification results in plant electrical signal dataset

Unlabeled ratio	Label Propagation	Self-Training	Pseudo Label	MTL	SSTSC	DA-FSSSC
90%	0.721	0.808	0.621	0.600	0.710	0.784
70%	0.774	0.854	0.684	0.852	0.787	0.888
50%	0.885	0.907	0.733	0.903	0.835	0.910

According to the comparison results in Table 5, the Self-Training model has the highest classification accuracy when the proportion of unlabeled dataset is higher than 90%, which indicates that Self-training is able to maximize the use of these limited labeled data to bootstrap the learning of unlabeled data, adapts to the new plant electrical signal data distribution in each iteration, and improves the adaptability of the model by continuously updating itself. The accuracy of the DA-FSSSC method ranked second among all methods. When the proportion of unlabeled data is 50% and 70%, the DA-FSSSC method utilizes both labeled data and a large amount of unlabeled data, combined with data augmentation, a small amount of labeled data can also generate more training samples, which can be used for training together with unlabeled data to improve the efficiency and effectiveness of model learning, and the model achieves satisfactory performance with limited plant electrical signal data. The classification accuracy of the semi-supervised classification method proposed in this paper is higher than the other compared models at both 70% and 50% of unlabeled datasets. Successful semi-supervised classification of plant electrical signal data allows depth study of different adaptive and response mechanisms of plants to salt stress, helps to reveal the physiological, genetic and molecular mechanisms of plants, contributes to the selection of varieties with strong salt tolerance, and provides an important basis for plant genetic improvement and breeding.

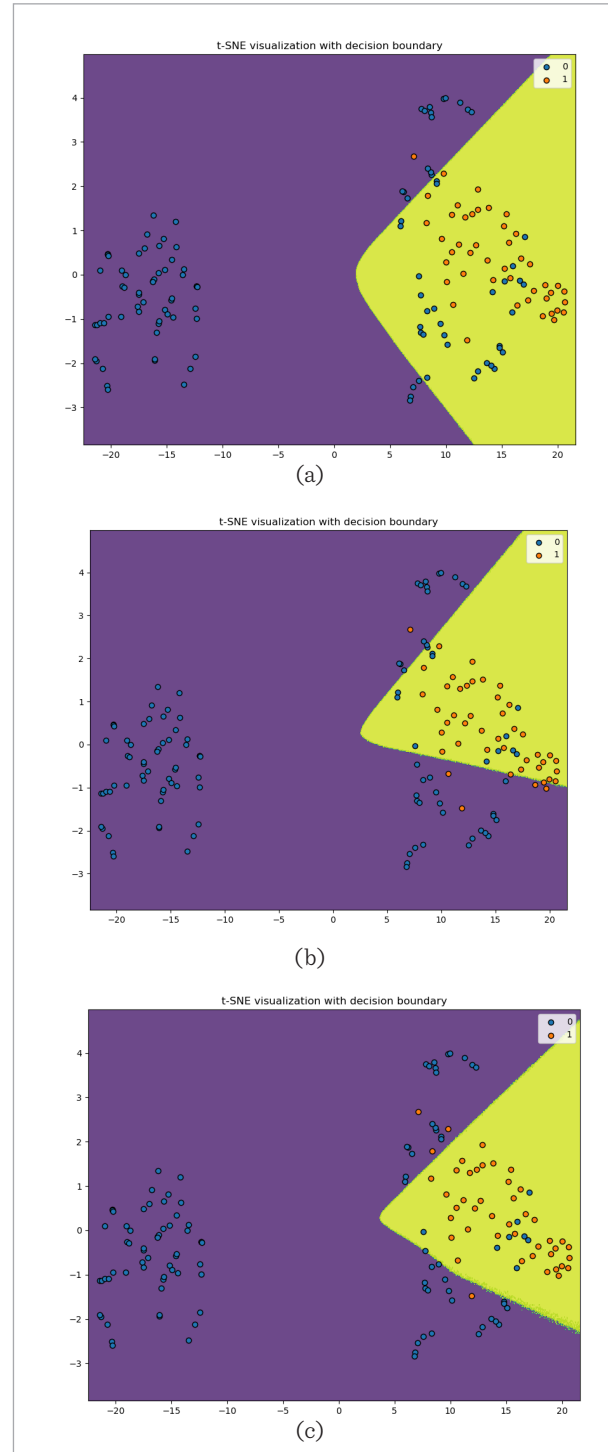
4.4.1. Visual Analysis of Results

t-SNE is a nonlinear dimensionality reduction technique that preserves the local structure in the data so that similar data points remain close in the lower dimensional space[10]. To better explain and visualize the semi-supervised classification strategy of our model for the plant electrical dataset, we used a dimensionality reduction technique to map the plant electrical data from a high-dimensional space to a two-dimensional space, to better understand the classification process of the semi-supervised classification module by drawing decision boundaries. Figure 8 illustrates the categorical t-SNE decision boundary plots for plant electrical signal data for different proportions of unlabeled data.

The analysis presented in Figure 8 demonstrates that our model effectively captures significant features in unlabeled data, and both self-supervised learning

Figure 8

Visualization of t-SNE and decision boundaries for the plant electric dataset (a) Rate of unlabeled dataset is 90%, (b) Rate of unlabeled dataset is 70%, (c) Rate of unlabeled dataset is 50%



and supervised learning techniques contribute to improved classification of plant electrical signal time series data. In general, the more labeled data there is, the better the model performs because the labeled data provides the model with more information about the true class labels, which helps the model learn decision boundaries and representations more accurately. When the proportion of unlabeled data is 50%, DA-FSSSC has more labeled examples to learn from and more information to train on, with higher classification accuracy. Although some plant electrical signal data could not be classified successfully for the time being, our model achieved good classification performance in most cases.

4.4.2. Ablation Study

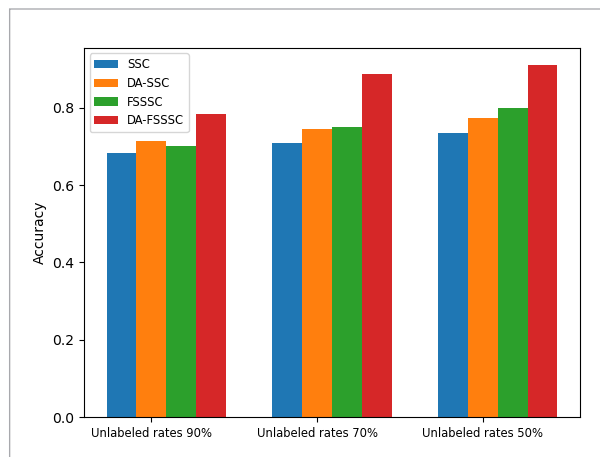
To demonstrate that each module in the semi-supervised classification model utilizes the strengths of the module, we conducted an ablation study of semi-supervised classification methods. Ablation studies were performed on the plant electrical signal dataset. The basic model is the semi-supervised classification method SSC, the second method is the semi-supervised classification method DA-SSC with the addition of a data enhancement module, the third model is the semi-supervised classification method FSSSC with the addition of a feature extraction and fusion module, and the fourth method is the semi-supervised classification model DA-FSSSC with the addition of a data enhancement module and a feature extraction and fusion module. Figure 9 illustrates the results of

the ablation study. As can be seen from Figure 9, with different proportions of unlabeled data, the classification accuracy of the DA-SSC method is higher than that of the SSC model, which indicates that the semi-supervised classification network is sufficiently trained after adding time series data, and the accuracy of classification is improved, although the network training sacrifices sometimes. CGAN generates data to enhance model robustness and prevent overfitting, and is an effective strategy to improve model quality in the presence of insufficient data. The classification accuracies of the FSSSC method are higher than that of the SSC method, which indicates that the feature extraction module can effectively learn the potential information of time series data and improve classification accuracy. The results of the ablation study effectively demonstrate that each module of DA-FSSSC contributes to the semi-supervised classification task under small sample conditions.

5. Results and Future Work

This study introduces a novel semi-supervised classification approach that involves augmenting the labeled samples through data generation, extracting the shape features of the time series, and employing a combination of self-supervised learning and supervised learning strategies for semi-supervised classification. The performance of the proposed model is initially assessed using the UCR dataset. The classification accuracy of the method was better than the comparison method at 90%, 70%, and 50% of unlabeled samples. Meanwhile, to verify the applicability of the method, we select ECG, EEG, and plant electrical signal datasets to test again, visualization analysis and ablation studies are carried out on the measured plant electrical signal datasets, and the results show that the method proposed in this paper achieves a better classification effect in real scenarios as well. CGAN can effectively augment the dataset by generating additional training samples to improve the diversity of the data, the fast shapelet algorithm is able to efficiently extract meaningful shapes from time-series data to improve the accuracy and efficiency of classification, The semi-supervised classification method enables the model to maintain a high level of flexibility and accuracy when dealing with time-series data with complex patterns and dynamic changes. Therefore, the method in this paper provides

Figure 9
Results of ablation studies



a powerful and flexible solution for dealing with the problem of classifying complex time-series data with small samples. The next step is to combine the improved feature learning and label learning methods to further improve the accuracy of the model classification with very few labeled time series samples through innovative methods and sufficient computational resources. In addition, it is important to consider how to improve the model when dealing with large datasets.

Acknowledgement

We thank prof. Hoang Anh Dau with the Department of Computer Science and Engineering, University of California for allowing us to use the UCR datasets. We thank A. Paeschke, Department of Communication Sciences, Technical University of Berlin, for the public dataset German Emotional Speech. We thank Postdoc Piyush Swami at Technical University of Denmark, Kongens Lyngby & Hvidovre Hospital for

allowing us to use the EEG Epilepsy Datasets.

Funding

This work was supported by the National Natural Science Foundation of China (Grant no. 62271488 and no. 61571443).

Data availability

The UCR datasets are available at https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

The ECG Datasets are available at <https://www.kaggle.com/datasets/devavratatripathy/ecg-dataset>.

The EEG Epilepsy Datasets are available at https://www.researchgate.net/publication/308719109-EEG_Epilepsy_Datasets.

The plant electrical signal datasets are available at <http://www.plantes.cn/>.

References

1. Abayomi-Alli, O. O., Damaševičius, R., Maskeliūnas, R., Abayomi-Alli, A. BiLSTM with Data Augmentation Using Interpolation Methods to Improve Early Detection of Parkinson Disease. 2020, 6-9 Sept. 2020. Paper Presented at the 2020 15th Conference on Computer Science and Information Systems (FedCSIS). <https://doi.org/10.15439/2020F188>
2. Abayomi-Alli, O. O., Sidekerskienė, T., Damaševičius, R., Šilka, J., Połap, D. Empirical Mode Decomposition Based Data Augmentation for Time Series Prediction Using NARX Network. 2020. Paper Presented at the International Conference on Artificial Intelligence and Soft Computing. https://doi.org/10.1007/978-3-030-61401-0_65
3. Al-Saegh, A., Dawwd, S. A., Abdul-Jabbar, J. M. Cut-Cat: An Augmentation Method for EEG Classification. *Neural Networks*, 2021, 141, 433-443. <https://doi.org/10.1016/j.neunet.2021.05.032>
4. Amini, M.-R., Feofanov, V., Pauletto, L., Devijver, E., Maximov, Y. J. A. E.-P. Self-Training: A Survey. 2022. arXiv:2202.12040. <https://ui.adsabs.harvard.edu/abs/2022arXiv220212040A> doi:10.48550/arXiv.2202.12040
5. Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., Rabbat, M. J. A. E.-P. Semi-Supervised Learning of Visual Features by Non-Parametrically Predicting View Assignments with Support Samples. 2021. arXiv:2104.13963. <https://ui.adsabs.harvard.edu/abs/2021arXiv210413963A> <https://doi.org/10.1109/ICCV48922.2021.00833>
6. Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S., Keogh, E. The UCR Time Series Archive. *IEEE/CAA Journal of Automatica Sinica*, 2019, 6(6), 1293-1305. <https://doi.org/10.1109/JAS.2019.1911747>
7. Degirmenci, A., Karal, O. Efficient Density and Cluster Based Incremental Outlier Detection in Data Streams. *Information Sciences*, 2022, 607, 901-920. <https://doi.org/10.1016/j.ins.2022.06.013>
8. Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwok, C. K., Li, X., Guan, C. Self-Supervised Contrastive Representation Learning for Semi-Supervised Time-Series Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, 1-15. <https://doi.org/10.1109/TPAMI.2023.3308189>
9. Fan, H., Zhang, F., Wang, R., Huang, X., Li, Z. Semi-Supervised Time Series Classification by Temporal Relation Prediction. 2021, 6-11 June 2021. Paper Presented at the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). <https://doi.org/10.1109/ICASSP39728.2021.9413883>
10. Gisbrecht, A., Schulz, A., Hammer, B. Parametric Non-linear Dimensionality Reduction Using Kernel t-SNE.

- Neurocomputing, 2015, 147, 71-82. <https://doi.org/10.1016/j.neucom.2013.11.045>
11. Goschenhofer, J. Deep Semi-Supervised Learning for Time-Series Classification. In *Deep Learning Applications, 2022, Volume 4* (pp. 361-384): Springer. https://doi.org/10.1007/978-981-19-6153-3_15
 12. Han, J., Gu, X., Lo, B. Semi-Supervised Contrastive Learning for Generalizable Motor Imagery EEG Classification. 2021, 27-30 July 2021. Paper Presented at the 2021 IEEE 17th International Conference on Wearable and Implantable Body Sensor Networks (BSN). <https://doi.org/10.1109/BSN51625.2021.9507038>
 13. He, Y., Song, K., Dong, H., Yan, Y. Semi-Supervised Defect Classification of Steel Surface Based on Multi-Training and Generative Adversarial Network. *Optics and Lasers in Engineering*, 2019, 122, 294-302. <https://doi.org/10.1016/j.optlaseng.2019.06.020>
 14. Jawed, S., Grabocka, J., Schmidt-Thieme, L. Self-Supervised Learning for Semi-Supervised Time Series Classification. Paper Presented at the *Advances in Knowledge Discovery and Data Mining, Cham*. 2020. https://doi.org/10.1007/978-3-030-47426-3_39
 15. Ji, C., Zhao, C., Liu, S., Yang, C., Pan, L., Wu, L., Meng, X. A Fast Shapelet Selection Algorithm for Time Series Classification. *Computer Networks*, 2019, 148, 231-240. <https://doi.org/10.1016/j.comnet.2018.11.031>
 16. Kendall, A., Gal, Y., Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. 2018. Paper presented at the *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
 17. Langlais, É. L., Thériault-Lauzier, P., Marquis-Gravel, G., Kulbay, M., So, D. Y., Tanguay, J.-F., Avram, R. Novel Artificial Intelligence Applications in Cardiology: Current Landscape, Limitations, and the Road to Real-World Applications. *Journal of Cardiovascular Translational Research*, 2023, 16(3), 513-525. <https://doi.org/10.1007/s12265-022-10260-x>
 18. Lao, Z., He, D., Jin, Z., Liu, C., Shang, H., He, Y. Few-shot Fault Diagnosis of Turnout Switch Machine Based on Semi-Supervised Weighted Prototypical Network. *Knowledge-Based Systems*, 2023, 274, 110634. <https://doi.org/10.1016/j.knosys.2023.110634>
 19. Lee, D.-H. Pseudo-label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. Paper Presented at the *Workshop on Challenges in Representation Learning, ICML*. 2013.
 20. Li, J.-H., Fan, L.-F., Zhao, D.-J., Zhou, Q., Yao, J.-P., Wang, Z.-Y., Huang, L. Plant Electrical Signals: A Multidisciplinary Challenge. *Journal of Plant Physiology*, 2021, 261, 153418. <https://doi.org/10.1016/j.jplph.2021.153418>
 21. Lin, K., Xu, X., Xiao, F. MFFusion: A Multi-level Features Fusion Model for Malicious Traffic Detection Based on Deep Learning. *Computer Networks*, 2022, 202, 108658. <https://doi.org/10.1016/j.comnet.2021.108658>
 22. Liu, D., Wang, P., Wang, T., Abdelzaher, T. Self-Contrastive Learning Based Semi-Supervised Radio Modulation Classification. Paper Presented at the *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*. 2021, 29 Nov.-2 Dec. 2021. <https://doi.org/10.1109/MILCOM52596.2021.9652914>
 23. Liu, H., Yu, C., Wu, H., Duan, Z., Yan, G. A New Hybrid Ensemble Deep Reinforcement Learning Model for Wind Speed Short Term Forecasting. *Energy*, 2020, 202, 117794. <https://doi.org/10.1016/j.energy.2020.117794>
 24. Liu, Z., Ma, Q., Ma, P., Wang, L. Temporal-Frequency Co-training for Time Series Semi-supervised Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, 37(7), 8923-8931. <https://doi.org/10.1609/aaai.v37i7.26072>
 25. Ma, Y., Shi, H., Tan, S., Tao, Y., Song, B. Consistency Regularization Auto-Encoder Network for Semi-Supervised Process Fault Diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 2022, 71, 1-15. <https://doi.org/10.1109/TIM.2022.3184346>
 26. Mao, W., Tian, S., Fan, J., Liang, X., Safian, A. Online Detection of Bearing Incipient Fault with Semi-Supervised Architecture and Deep Feature Representation. *Journal of Manufacturing Systems*, 2020, 55, 179-198. <https://doi.org/10.1016/j.jmsy.2020.03.005>
 27. Memarzadeh, M., Matthews, B., Templin, T. Multiclass Anomaly Detection in Flight Data Using Semi-Supervised Explainable Deep Learning Model. *Journal of Aerospace Information Systems*, 2021, 19(2), 83-97. <https://doi.org/10.2514/1.1010959>
 28. Irza, M., Osindero, S. J. A. E.-P. Conditional Generative Adversarial Nets. 2014. arXiv:1411.1784. <https://ui.adsabs.harvard.edu/abs/2014arXiv1411.1784M> doi:10.48550/arXiv.1411.1784
 29. Rakthanmanon, T., Keogh, E. Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets. In *Proceedings of the 2013 SIAM International Conference on Data Mining (SDM)* (pp. 668-676). <https://doi.org/10.1137/1.9781611972832.74>
 30. Ratna Prakarsha, K., Sharma, G. Time Series Signal Forecasting Using Artificial Neural Networks: An Application on ECG Signal. *Biomedical Signal Processing and Control*, 2022, 76, 103705. <https://doi.org/10.1016/j.bspc.2022.103705>
 31. Rezaei, S., Liu, X. J. A. E.-P. How to Achieve High Classification Accuracy with Just a Few Labels: A Semi-supervised Approach Using Sampled Packets. 2018. arXiv:1812.09761.

- <https://ui.adsabs.harvard.edu/abs/2018arXiv181209761R>
doi:10.48550/arXiv.1812.09761
32. Roy, S. K., De, D. Genetic Algorithm based Internet of Precision Agricultural Things (IopaT) for Agriculture 4.0. *Internet of Things*, 2022, 18, 100201. <https://doi.org/10.1016/j.iot.2020.100201>
 33. Šubelj, L. J. A. E.-P. Label Propagation for Clustering. 2017. arXiv:1709.05634. <https://ui.adsabs.harvard.edu/abs/2017arXiv170905634S> doi:10.48550/arXiv.1709.05634
 34. Swami, P., Panigrahi, B., Nara, S., Bhatia, M., Gandhi, T. EEG Epilepsy Datasets. 2016.
 35. Wang, H., Zhang, Q., Wu, J., Pan, S., Chen, Y. Time Series Feature Learning with Labeled and Unlabeled Data. *Pattern Recognition*, 2019, 89, 55-66. <https://doi.org/10.1016/j.patcog.2018.12.026>
 36. Wei, C., Wang, Z., Yuan, J., Li, C., Chen, S. Time-frequency Based Multi-task Learning for Semi-supervised Time Series Classification. *Information Sciences*, 2023, 619, 762-780. <https://doi.org/10.1016/j.ins.2022.11.040>
 37. Wu, X., Zhang, Y., Cheng, C., Peng, Z. A Hybrid Classification Autoencoder for Semi-supervised Fault Diagnosis in Rotating Machinery. *Mechanical Systems and Signal Processing*, 2021, 149, 107327. <https://doi.org/10.1016/j.ymsp.2020.107327>
 38. Xi, L., Yun, Z., Liu, H., Wang, R., Huang, X., Fan, H. Semi-supervised Time Series Classification Model with Self-supervised Learning. *Engineering Applications of Artificial Intelligence*, 2022, 116, 105331. <https://doi.org/10.1016/j.engappai.2022.105331>
 39. Xi, W., Jain, A., Zhang, L., Lin, J. J. A. E.-P. LB-SimTSC: An Efficient Similarity-Aware Graph Neural Network for Semi-Supervised Time Series Classification. 2023. arXiv:2301.04838. <https://ui.adsabs.harvard.edu/abs/2023arXiv230104838X> doi:10.48550/arXiv.2301.04838
 40. Yang, Y., Truong, N. D., Eshraghian, J. K., Nikpour, A., Kavehei, O. Adaptive, Unlabeled and Real-time Approximate-Learning Platform (AURA) for Personalized Epileptic Seizure Forecasting. 2021.2009.2030.21264287. <https://doi.org/10.1101/2021.09.30.21264287>
 41. Yao, J.-P., Wang, Z.-Y., de Oliveira, R. F., Wang, Z.-Y., Huang, L. A Deep Learning Method for the Long-term Prediction of Plant Electrical Signals Under Salt Stress to Identify Salt Tolerance. *Computers and Electronics in Agriculture*, 2021, 190, 106435. <https://doi.org/10.1016/j.compag.2021.106435>
 42. Yongtao, H., Jinfeng, G., Qiang, Z., Zheng, F. Bearing Fault Diagnosis Based on Deep Semi-supervised Small Sample Classifier. Paper Presented at the 2019 Prognostics and System Health Management Conference (PHM-Qingdao). 2019, 25-27 Oct. 2019. <https://doi.org/10.1109/PHM-Qingdao46334.2019.8943025>
 43. Zhao, X., Ding, S., An, Y., Jia, W. Applications of Asynchronous Deep Reinforcement Learning Based on Dynamic Updating Weights. *Applied Intelligence*, 2019, 49(2), 581-591. <https://doi.org/10.1007/s10489-018-1296-x>
 44. Zheng, Q., Tian, X., Yu, Z., Jiang, N., Elhanashi, A., Saponara, S., Yu, R. Application of Wavelet-packet Transform Driven Deep Learning Method in PM2.5 Concentration Prediction: A Case Study of Qingdao, China. *Sustainable Cities and Society*, 2023, 92, 104486. <https://doi.org/10.1016/j.scs.2023.104486>
 45. Zheng, Q., Tian, X., Yu, Z., Jin, B., Jiang, N., Ding, Y., Kpalma, K. Application of Complete Ensemble Empirical Mode Decomposition Based Multi-stream Informer (CE-EMD-MsI) in PM2.5 Concentration Long-Term Prediction. *Expert Systems with Applications*, 2024, 245, 123008. <https://doi.org/10.1016/j.eswa.2023.123008>
 46. Zheng, Q., Tian, X., Yu, Z., Wang, H., Elhanashi, A., Saponara, S. DL-PR: Generalized Automatic Modulation Classification Method Based on Deep Learning with Priori Regularization. *Engineering Applications of Artificial Intelligence*, 2023, 122, 106082. <https://doi.org/10.1016/j.engappai.2023.106082>
 47. Zheng, Q., Zhao, P., Wang, H., Elhanashi, A., Saponara, S. Fine-Grained Modulation Classification Using Multi-Scale Radio Transformer with Dual-Channel Representation. *IEEE Communications Letters*, 2022, 26(6), 1298-1302. <https://doi.org/10.1109/LCOMM.2022.3145647>
 48. Zheng, Q., Zhao, P., Zhang, D., Wang, H. MR-DCAE: Manifold Regularization-Based Deep Convolutional Autoencoder for Unauthorized Broadcasting Identification. *International Journal of Intelligent Systems*, 2021, 36(12), 7204-7238. <https://doi.org/10.1002/int.22586>
 49. Zhou, K., Diehl, E., Tang, J. Deep Convolutional Generative Adversarial Network with Semi-Supervised Learning Enabled Physics Elucidation for Extended Gear Fault Diagnosis Under Data Limitations. *Mechanical Systems and Signal Processing*, 2023, 185, 109772. <https://doi.org/10.1016/j.ymsp.2022.109772>

