

Research Article

A Contrastive Predictive Coding-Based Classification Framework for Healthcare Sensor Data

Chaoxu Ren ¹, Le Sun ¹, and Dandan Peng ²

¹Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing, China

²School of Computer Science and Network Engineering, Guangzhou University, Guangzhou, China

Correspondence should be addressed to Le Sun; lesun1@nuist.edu.cn

Received 22 January 2022; Revised 8 February 2022; Accepted 11 February 2022; Published 15 March 2022

Academic Editor: Bhagyaveni M.A

Copyright © 2022 Chaoxu Ren et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Supervised learning technologies have been used in medical-data classification to improve diagnosis efficiency and reduce human diagnosis errors. A large amount of manually annotated data are required for the fully supervised learning process. However, annotating data information will consume a large amount of manpower and resources. Self-supervised learning has great advantages in solving this problem. Self-supervised learning mainly uses pretext tasks to mine its own supervised information from large-scale unsupervised data. And this constructed supervised information is used to train the network to learn valuable representations for downstream tasks. This study designs a general and efficient model for the diagnosis and classification of medical sensor data based on contrastive predictive coding (CPC) in self-supervised learning, called TCC, which consists of two steps. The first step is to design a pretext task based on the idea of CPC, which aims to extract effective features between different categories using its encoder. The second step designs a downstream classification task with lower time and space complexity to perform a supervised type of training using the features extracted by the encoder of the pretext task. Finally, to demonstrate the performance of the proposed framework in this paper, we compare the proposed framework with recent state-of-the-art works. Experiments comparing the proposed framework with supervised learning are also set up under the condition of different proportions of labeled data.

1. Introduction

Healthcare as an important part of smart cities directly affects the quality of smart city construction. In recent years, the rapid growth of urban population density, population aging, and various chronic diseases have brought challenges to the development of smart healthcare [1]. This no longer meets the requirements of sustainable urban development, prompting a shift from hospital-centered to family-centered healthcare [2]. The application of various deep learning algorithms has made it less difficult to automatically classify diseases and has greatly improved the accuracy of disease classification [3, 4]. The classification model can be paired with various IoT devices for real-time diagnosis [5], and patients can grasp their health status at home without having to go to the hospital for checkups every time, which will ease

the tension on medical resources and help the construction of smart medical care to achieve sustainable urban development.

However, traditional supervised learning training requires a large amount of labeled data to achieve good results. For medical data with few labels and a high labeling threshold [6], traditional supervised training is no longer suitable [7]. Self-supervised learning can well solve the problem of unlabeled medical data by creating pseudolabels [8]. Self-supervised learning methods learn more general features rather than task-specific features, so models using self-supervised learning can be reused for different tasks in the same domain and can better perform the task of classifying medical sensor data [9].

In this paper, we use contrastive predictive coding in self-supervised learning to accomplish the classification of

medical sensor data. We build a two-step CPC-based classification framework (TCC) for medical sensor data and conduct experiments on two types of medical sensor data: electroencephalogram (EEG) and electrocardiogram (ECG). By establishing a model for real-time automatic classification, it helps to alleviate the increasing strain on medical resources and promote the sustainable development of smart cities.

In summary, the main contributions of this work are as follows.

We propose a two-step TCC model according to the architecture and ideas of contrastive predictive coding in self-supervised learning. First step, designing a contrastive predictive coding (CPC)-based pretext task for medical sensor data classification, then redesigning the arrangement of positive sample pairs and negative sample pairs. The second step is to design a lightweight and simple downstream classification model, which further improves the classification accuracy, achieving a very good result.

In order to verify that the pretext task is indeed learning useful features, we designed the classification experiments using fully supervised learning and the pretext task in the case of different numbers of sample labels (10%, 30%, 50%, 70%, and 100%). Experiments have proved that the pretext task is indeed learning useful features. When the number of sample labels is small, after using the CPC-based pretext task, the classification accuracy is still maintained at a very high level.

The rest of this paper is organized as follows: Section 2 introduces the related work, including two aspects; Section 3 presents TCC, which contains CPC-based pretext task (first step) and a downstream classification task (second step); Section 4 shows the experiment procedure and experiment results; and Section 5 concludes this paper and gives some future research directions.

2. Related Works

Many deep learning technologies have been applied to medical data classification and have achieved great success [10]. Automatic recognition of sleep classification through feature extraction started a long time ago [11]. Automatic classification of sleep states based on EEG has been a hot research topic in the field of health informatics.

2.1. Supervised Learning Classification Methods. Akara et al. [12] proposed a two-step training method to train their model, which is named DeepSleepNet. In their model, they utilized convolutional neural networks (CNN) to extract z-time-variable features and bidirectional-long short-term memory (Bi-LSTM) to learn transition rules among sleep stages automatically from EEG epochs. Sajad et al. [13] proposed a deep learning model called SleepEEGNet, which is composed of a convolutional neural network to capture time-variable features and frequency information. The model also used a sequence-to-sequence model to capture the complex and long short-term context dependencies between sleep epochs and scores. Huy et al. [14] proposed a

hierarchical recurrent neural network named SeqSleepNet which treated the task as a sequence-to-sequence classification problem. Koushik et al. [15] performed end-to-end training on the EEG dataset using a time-distributed convolutional neural network.

2.2. Unsupervised Learning Classification Methods. Emadeldeen et al. [16] proposed a model based on unsupervised learning named TS-TCC and designed contrastive learning through weak data augmentation and strong data augmentation. A cross-view prediction task is one of the highlights of this paper. Hogeon et al. [17] proposed a model named IITNet which utilized residual neural networks and bidirectional-long short-term memory networks for sleep classification. Mohsenvand et al. [18] extended the SimCLR [19] framework to time-series data and realized different classification tasks. Yang et al. [20] proposed a self-supervised learning model called ContraWR and conducted experiments on three EEG datasets. Zhang et al. [21] proposed a generative adversarial network-based data enhancement method to improve accuracy and prevent overfitting.

3. TCC Framework

3.1. Contrastive Predictive Coding. Contrastive predictive coding was proposed in 2018. The purpose is to predict future features from past features by training a neural network, which can be used on pictures or data with time-series features. The core idea of this method is contrastive learning. We can learn more global and meaningful structures instead of small irrelevant details by predicting far into the future. The core of contrastive learning is to learn a mapping function f and encode the sample x into its representation $f(x)$. The core of contrastive learning is to make this $s(f(x), f(x^+)) \gg s(f(x), f(x^-))$.

$$s(f(x), f(x^+)) \gg s(f(x), f(x^-)). \quad (1)$$

Here x^+ is a sample similar to x , and x^- is a sample that is not similar to x . $s()$ is a function that measures the degree of similarity between samples. A typical score function is the vector inner product. That is to optimize the following expectations:

$$E_{x, x^+, x^-} \left[-\log \left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + e^{f(x)^T f(x^-)}} \right) \right]. \quad (2)$$

Contrastive predictive coding is an approach for unsupervised learning from high-dimensional data by translating a generative modeling problem to a classification problem. The contrastive loss, or InfoNCE loss, in CPC, inspired by noise contrastive estimation (NCE) [22], uses cross-entropy loss to measure how well the model can classify the “future” representation amongst a set of unrelated “negative” samples. Such design is partially motivated by the fact that the unimodal loss like MSE has not had enough capacity but learning a full generative model could be too expensive. The (3) represents the mutual information

between x and c that we want to maximize, where c is the potential content representation vector and x is the sample. By doing so, we extract the underlying latent variables that the inputs have in common.

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x, c)}{p(x)}. \quad (3)$$

For EEG signals, we have made a little innovation here, which is to predict by establishing positive and negative sample pairs instead of predicting the future. For positive sample pairs, they belong to the same category, and the features extracted by train data should be used for prediction. It is highly similar to the coding features of waiting train data. For negative sample pairs, because they belong to different categories, when predicting the features extracted by train data, the less similar the coding features of the waiting train data, the better. So, the goal is to maximize the similarity between positive sample pairs and minimize the similarity between negative sample pairs.

We establish positive and negative sample pairs, where the positive sample pair contains 8 different samples belonging to the same category, and the four left and four right of the negative sample pair belong to the same category, but the left and right are different categories. The label of the positive sample pair is 1, and the label of the negative sample pair is 0. The left half of the training sample is called the training set, and the right half is called the waiting training set. Figure 1 describes the details. Algorithm 1 describes the process to establish positive sample pairs and negative sample pairs.

3.2. Pretext Task. The structure of the pretext task model is shown in Figure 2. Giving a batch of train set samples x^t and a batch of waiting train set samples x^w , an encoder g_{enc} maps the input into Z_j^T ($0 \leq j \leq t$), Z_j^W ($0 \leq j \leq t$), respectively. Next, a GRU model g_{ar} summarizes all Z_j^T ($0 \leq j \leq t$) in the latent space and produces a context latent representation c . Finally, we use the content vector c for multistep prediction and calculate the loss value with Z_j^W ($0 \leq j \leq t$). The loss function uses `binary_crossentropy`, the formula is as follows:

$$L_{pre} = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)), \quad (4)$$

where y is the true label (1 for positive sample pairs and 0 for negative sample pairs) and $p(y)$ is the calculated probability of being a positive sample.

The encoder part contains four identical blocks, and each block contains a dense layer, a batch normalization layer, an activation layer, and finally a dense layer to output the coding features. It is worth noting that the quality of the pretext task training directly affects the performance of the downstream classification model, so the model of the pretext task needs to be fully trained. Here we have trained 20 epochs. At the same time, since the training samples are randomly selected, in order to ensure the probability of the

samples being selected, each epoch is trained thousands of times to ensure that the pretext task can be fully trained.

3.3. Classification Task. The downstream classification task uses the encoder part of the pretext task. The encoder part saves the model parameters after the pretext task is trained and loads the model parameters directly. The classification model structure is shown in Figure 3. We can see that the model is very lightweight and concise, and no particularly complex structure is used. The classification model contains two Conv1 layers that are not exactly the same; they have different filters and kernel size.

In order to maintain the dimensionality of the input data of the encoder layer, a sample is copied four times before classification. For example, for a sample x_1 , the shape of its input model should be $[x_1, x_1, x_1, x_1]$. In order to speed up the convergence of the model and get good results, monitor the change of the validation set loss. When the performance is not improved within two epochs, the learning rate will be reduced to 1/3 of the original, and the initial learning rate is set to 0.001. The loss function here uses `categorical_crossentropy`, which is used as a loss function for multiclass classification models where there are two or more output labels. The output label is assigned a one-hot category encoding value in the form of 0 and 1. Algorithm 2 describes the overall classification model.

4. Experiments and Results

4.1. Datasets. The American Academy of Sleep Medicine (AASM) divides sleep data into five stages, namely awake (W), stages 1–3 (N1, N2, and N3), and rapid eye movement (REM) [23]. In addition, N1, N2, and N3, respectively, represent transitional sleep, light sleep, and deep sleep, respectively. We aim to classify the input EEG signal into one of five classes and download the sleep-EDF dataset from the PhysioBank. The sleep-EDF database contains 197 whole-night polysomnographic (PSG) sleep recordings, containing EEG, EOG, chin EMG, and event markers, where we used a single EEG channel (Fpz-Cz) with a sampling rate of 100 Hz [24]. Table 1 shows the total number for each class. Figure 4 shows the waveform variations for each category.

4.2. Pretext Task Results. Figure 4 shows the trend of the accuracy of the training set and test set in the pretext task. It can be seen from Figure 5 that the result of the training set is more than 99%, and the result of the test set is more than 98%. If the pretext task is not fully trained, the accuracy of the downstream classification task is about 70%.

4.3. Classification Task Results. Table 2 shows the confusion matrix after inputting all the datasets into the classification model. The last three columns represent the performance indicators of each category according to the confusion matrix. It can be seen that the classification effect for all sleep stages is very good, especially the N1 category, which shows a good classification effect

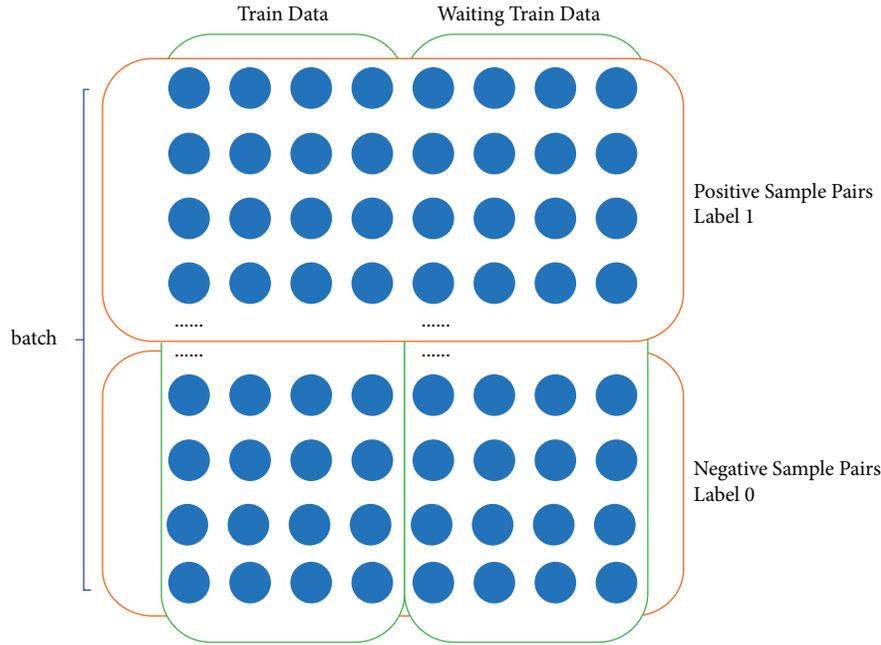


FIGURE 1: The division method of the positive sample pair, the negative sample pair, and the training set to be trained. Looking from the top down, the top half and the bottom half are positive and negative sample pairs, respectively. Viewed from left to right, the left half is the training set, and the right half is the waiting training set.

```

(i) Input: nput training set data $\{X\}_{\text{train}}$ .  $X_{\text{train}}$  is a sample of  $\{X\}_{\text{train}}$ .
(ii) Output: train data batch, train label batch
(1) 2003for epochs do
(2)   Create an empty train data batch and an empty train label batch, the shape is (32, 8), (32, 1), respectively.
(3)   for  $i < 32$  do
(4)     for  $j > 16$  do
(5)       Random select 16 numbers from the total categories. Each number is repeated 8 times and the shape is (16, 8). This is the
       upper part of Figure 1.
(6)       Fill train label batch as 1
(7)     end for
(8)     for  $k$  in (16, 32) do
(9)       Random select 16 numbers from the total categories. Each number is repeated 4 times and the shape is (16, 4). This is the
       lower left part of Figure 1.
(10)      Select 16 numbers different from the previously selected category. Each number is repeated 4 times and the shape is (16, 4).
       This is the lower right part of Figure 1.
(11)      Fill train label batch as 0.
(12)    end for
(13)  end for
(14)  Compose training data
(15)  Randomly select  $X_{\text{train}}$  according to the selected sample category to fill the train data batch.
(16)  Randomly disrupted train data batch and train label batch.
(17)  return train data batch, train label batch
(18) end for

```

ALGORITHM 1: Establish sample pairs.

compared to other models [12, 13], which shows that our pretext task is quite effective. The average value of F1 is 88.09, and the overall accuracy is 88.70. We compare the performance using two metrics namely the accuracy (ACC) and the macro-averaged F1-score (MF1), with other proposed models. Table 3 shows the details.

4.4. Few Data Results. Inspired by [25], we did this experiment. Figure 6 shows the change trend of the accuracy of the classification model prediction when the pretext task is used and the supervised learning is used when samples of different proportions are used. The supervised learning here refers to the model without using the encoder part

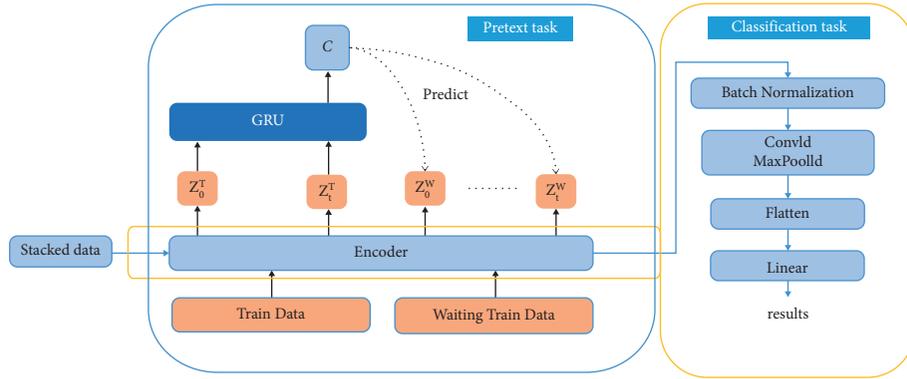


FIGURE 2: Model architecture.

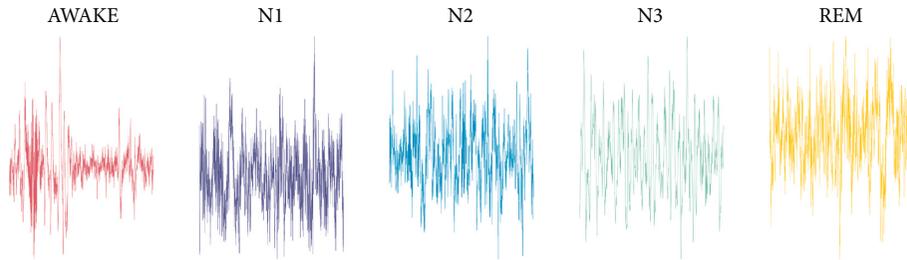


FIGURE 3: Waveform diagram for each category.

- (i) Input: train data batch, train label batch
 - (1) Train the CPC-based model.
 - (2) Save the parameters of the Encoder part of the trained model.
 - (3) Load model parameters as the first layer of the classification model
 - (4) Train the classification model.

ALGORITHM 2: Overall classification model.

TABLE 1: The sleep-EDF dataset.

	Awake	N1	N2	N3	REM	Total
Total number	8285	2804	17799	5703	7717	42308

parameters saved by the pretext task, directly use the encoder part for training. The results show that when the number of sample labels is small, the accuracy of the model can still be maintained at a high level after using the pretext task.

4.5. Experiment: Sustainability of CPC-Based Model. This experiment validates the sustainability of the model on another dataset: the MIT-BIH supraventricular arrhythmia database (MIT-BIH-SUP). This dataset includes 78 half-hour ECG recordings chosen to supplement the examples of supraventricular arrhythmias in the MIT-BIH arrhythmia database [26]. The Association for Advancement of Medical Instrumentation (AAMI) classifies the heartbeats of arrhythmia patients into five classes: normal beat (N), supraventricular ectopic beat (S), ventricular ectopic beat (V), fusion beat (F), and unclassifiable beat (Q) [27]. Since the number of F and Q

data is very small, we use this model to perform three classification experiments on N, S, and V. We resample the sampling rate from 128 Hz to 251 Hz and divide the dataset into a training set and test set according to the ratio of 9:1 and the results are shown in Table 4. The accuracy of the deep learning model proposed in [28] is only 88.2%. In contrast, TCC has a huge improvement.

4.6. An Industry Application of TCC for Improving the Development of Sustainable Smart Cities. Among all the facilities provided by smart cities to citizens, smart medical treatment is the most important and most concerned about the well-being of the people. Smart medical combines intelligent technology with medical health and can use a variety of wearable devices to obtain human health data. Doctors, researchers, and healthcare professionals can analyze these data to obtain better-personalized diagnoses and solutions. By deploying the classification model on small mobile devices and cooperating with the use of various sensors, patients can master their health status in real-time, avoiding the various complicated steps of going to the hospital every

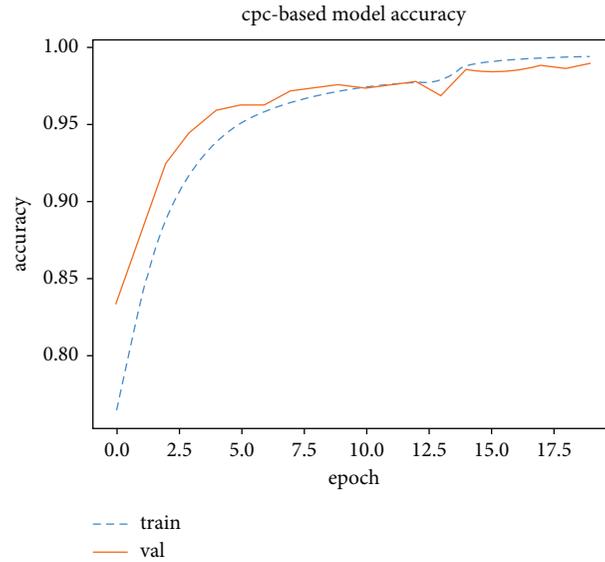


FIGURE 4: Pretext task training results.

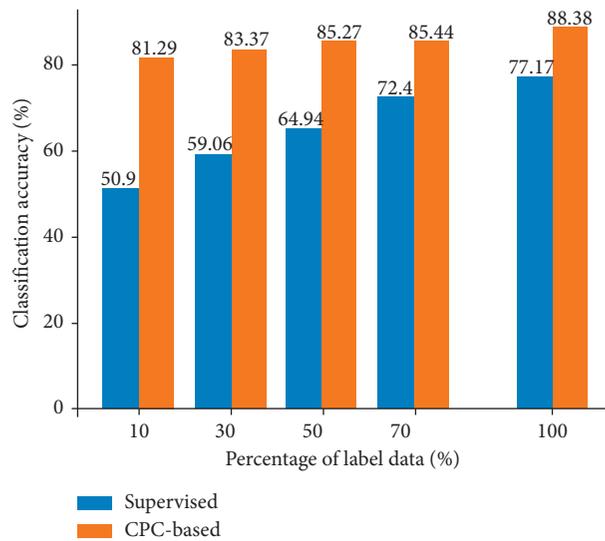


FIGURE 5: The change trend of the accuracy of different numbers of samples in different models.

TABLE 2: Confusion matrix and various evaluation indicators.

	Awake	N1	N2	N3	REM	PR (%)	RE (%)	F ₁ (%)
Awake	7270	235	349	79	352	92.55	87.75	90.09
N1	55	2572	75	4	98	78.58	91.73	84.65
N2	278	269	15418	396	1438	93.25	86.62	89.81
N3	69	2	506	5126	0	91.29	89.88	90.58
REM	183	195	186	10	7143	79.09	92.56	85.30
Macro avg								88.09
ACC	88.70%							

time, which is conducive to the construction of a sustainable smart city. Figure 6 shows an industry application of TCC for sustainable smart cities.

Patients can select appropriate medical monitoring equipment according to their actual situation. This equipment will transfer the obtained medical sensor data

to the TCC system, and the system will analyze whether the medical data is abnormal in real-time. In the event of an abnormality, a warning will be issued to prompt the patient to go to the hospital on time, and the abnormal medical data flow will be recorded to facilitate the doctor's diagnosis and analysis.

TABLE 3: Comparison between our proposed model against others.

	ACC (%)	MF1 (%)
SSL-ECG [9]	74.58	65.44
SimCLR [9]	78.91	68.60
TS-TCC [9]	83.00	73.57
DeepSleepNet [6]	82.00	76.88
IITNet [10]	84.00	77.70
SleepEEGNet [7]	84.26	79.66
CPC-based (ours)	88.70	88.09

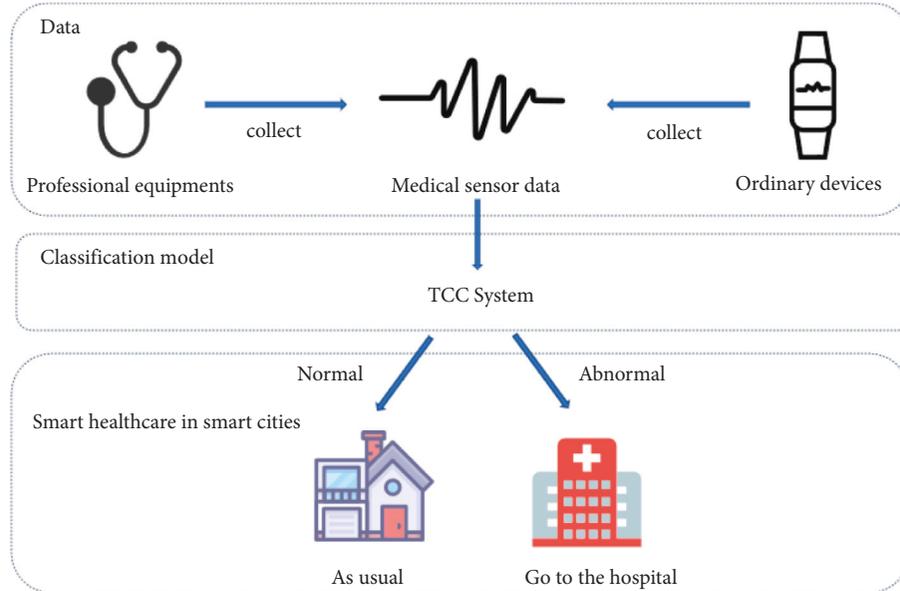


FIGURE 6: An industry application of TCC for sustainable smart cities.

TABLE 4: Confusion matrix and various evaluation indicators.

ACC (%)	N(%)			S(%)			V(%)		
	PR	RE	F ₁	PR	RE	F ₁	PR	RE	F ₁
97.30	97.90	99.42	98.65	89.80	77.28	83.07	95.24	87.65	91.29

5. Conclusion

We exploit a self-supervised deep learning framework for sleep stage classification. Based on the architecture and ideas of contrastive predictive coding, this paper proposes a CPC-based pretext task that uses positive sample pairs and negative sample pairs to design contrastive learning, and the model finally extracts different types of effective features. Using the encoder part of the pretext task, a very lightweight classification model is designed, which achieves very good results on the dataset. The F_1 -scores of classifying awake, N1, N2, N3, and, REM sleep stages are 90.09%, 84.65%, 89.81%, 90.58%, and 85.30%, respectively. At the same time, we verified that in the case of a small amount of data labels, the model still achieved good results, and the performance of the model exceeded the supervised learning. We extend the experiments in another dataset, which shows the robustness and

sustainability of the model more efficiently. In the future, we plan to use more complex or time-based classification models to further improve the accuracy of model classification. Although the sample imbalance did not affect the final experimental results, we still plan to utilize some machine learning methods, such as the synthetic minority oversampling technique to solve this problem.

Data Availability

The labeled datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest with this study.

References

- [1] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.
- [2] S. Alromaihi, W. Elmedany, and C. Balakrishna, "Cyber security challenges of deploying iot in smart cities for healthcare applications," in *Proceedings of the 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pp. 140–145, Barcelona, Spain, August 2018.
- [3] S.-Y. Hu, S. Wang, W.-H. Weng et al., "Self-supervised pretraining with DICOM metadata in ultrasound imaging," in *Proceedings of the 5th Machine Learning for Healthcare Conference*, F. Doshi-Velez, J. Fackler, K. Jung et al., Eds., pp. 732–749, August 2020.
- [4] W. Tan, P. Huang, X. Li, G. Ren, Y. Chen, and J. Yang, "Analysis of segmentation of lung parenchyma based on deep 180 learning methods," *Journal of X-Ray Science and Technology*, vol. 29, no. 6, 2021.
- [5] Y. Wang, L. Sun, and S. Subramani, "Cab: classifying arrhythmias based on imbalanced sensor data," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 7, pp. 2304–2320, 2021.
- [6] H. Banville, I. Albuquerque, A. Hyvarinen, G. Moffat, D.-A. Engemann, and A. Gramfort, "Self-supervised representation learning from electroencephalography signals," in *Proceedings of the 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Pittsburgh, PA, USA, October 2019.
- [7] H. Banville, O. Chehab, A. Hyvarinen, D.-A. Engemann, and A. Gramfort, "Uncovering the structure of clinical signals with self-supervised learning," *Journal of Neural Engineering*, vol. 18, no. 4, Article ID 046020, 2021.
- [8] A. Chowdhury, J. Rosenthal, J. Waring, and R. Umeton, "Applying self-supervised learning to medicine: review of the state of the art and medical implementations," *Informatics*, vol. 8, no. 3, p. 59, 2021.
- [9] P. Sarkar and A. Etemad, "Self-supervised learning for ecg-based emotion recognition," in *Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3217–3221, Barcelona, Spain, May 2020.
- [10] W. Tan, L. Zhou, X. Li, X. Yang, Y. Chen, and J. Yang, "Automated vessel segmentation in lung ct and cta images via deep neural networks," *Journal of X-Ray Science and Technology*, vol. 29, no. 6, 2021.
- [11] N. Schaltenbrand, R. Lengelle, M. Toussaint et al., "Sleep stage scoring using the neural network model: Comparison between visual and automatic analysis in normal subjects and patients," *Sleep*, vol. 19, no. 1, pp. 26–35, 1996.
- [12] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.
- [13] S. Mousavi, F. Afghah, and U. R. Acharya, "Sleeppegnet: automated sleep stage scoring with sequence to sequence deep learning approach," *PLoS One*, vol. 14, no. 5, Article ID e0216456, 2019.
- [14] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. De Vos, "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging network for sequence-to-sequence automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 400–410, 2019.
- [15] A. Koushik, J. Amores, and P. Maes, "Real-time smartphone-based sleep staging using 1-channel eeg," in *Proceedings of the 2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1–4, Chicago, USA, May 2019.
- [16] E. Eldele, M. Ragab, Z. Chen et al., "Time-series representation learning via temporal and contextual contrasting," <https://arxiv.org/abs/2106.14112>.
- [17] H. Seo, S. Back, S. Lee, D. Park, T. Kim, and K. Lee, "Intra- and inter-epoch temporal context network (iitnet) using sub-epoch features for automatic sleep scoring on raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 61, Article ID 102037, 2020.
- [18] M. N. Mohsenvand, M. R. Izadi, and P. Maes, "Contrastive representation learning for electroencephalogram classification," in *Proceedings of the Machine Learning for Health*, pp. 238–253, PMLR, August 2020, <https://proceedings.mlr.press/v136/mohsenvand20a.html>.
- [19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International conference on machine learning*, pp. 1597–1607, PMLR, 2020, <https://proceedings.mlr.press/v119/chen20j.html>.
- [20] C. Yang, D. Xiao, M. B. Westover, and J. Sun, "Self-supervised eeg representation learning for automatic sleep staging," arXiv preprint arXiv:2110.15278.
- [21] Z. Zhang, S.-h. Zhong, Y. Liu, and Ganser, "A self-supervised data augmentation framework for eeg-based emotion recognition," arXiv preprint arXiv:2109.03124.
- [22] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2021.
- [23] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, and B. V. Vaughn, "The aasm manual for the scoring of sleep and associated events, Rules, Terminology and Technical Specifications, Darien, Illinois," *American Academy of Sleep Medicine*, vol. 176, 2012.
- [24] A. L. Goldberger, L. A. Amaral, L. Glass et al., "Complex physiological signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [25] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proceedings of the International Conference on Machine Learning*, pp. 4182–4192, PMLR, 2020, <https://arxiv.org/abs/1905.09272>.
- [26] S. D. Greenwald, R. S. Patil, and R. G. Mark, "Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information," *Biomedical Instrumentation & Technology*, IEEE, vol. 26, no. 2, 1992.
- [27] L. Sun, Y. Wang, Z. Qu, N. N. Xiong, and Beatclass, "Beat-Class: a sustainable ECG classification system in IoT-based eHealth," *IEEE Internet of Things Journal*, p. 1, 2021.
- [28] J. He, J. Rong, L. Sun, H. Wang, and Y. Zhang, "An advanced two-step dnn-based framework for arrhythmia detection," *Advances in Knowledge Discovery and Data Mining*, vol. 12085, pp. 422–434, 2020.