

Research Article

Research on Healthy Anomaly Detection Model Based on Deep Learning from Multiple Time-Series Physiological Signals

**Kai Wang,¹ Youjin Zhao,¹ Qingyu Xiong,^{2,3} Min Fan,¹ Guotan Sun,¹
Longkun Ma,¹ and Tong Liu¹**

¹*School of Automation, Chongqing University, Chongqing, China*

²*Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University,
Ministry of Education, Chongqing, China*

³*School of Software Engineering, Chongqing University, Chongqing, China*

Correspondence should be addressed to Kai Wang; akyle@163.com

Received 16 June 2016; Accepted 17 August 2016

Academic Editor: X. Wang

Copyright © 2016 Kai Wang 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.

Health is vital to every human being. To further improve its already respectable medical technology, the medical community is transitioning towards a proactive approach which anticipates and mitigates risks before getting ill. This approach requires measuring the physiological signals of human and analyzes these data at regular intervals. In this paper, we present a novel approach to apply deep learning in physiological signals analysis that allows doctor to identify latent risks. However, extracting high level information from physiological time-series data is a hard problem faced by the machine learning communities. Therefore, in this approach, we apply model based on convolutional neural network that can automatically learn features from raw physiological signals in an unsupervised manner and then based on the learned features use multivariate Gauss distribution anomaly detection method to detect anomaly data. Our experiment is shown to have a significant performance in physiological signals anomaly detection. So it is a promising tool for doctor to identify early signs of illness even if the criteria are unknown a priori.

1. Introduction

Over the years, all kinds of intelligent devices and modern life are more and more inseparable [1]. People can record various kinds of physiological time-series data through those devices at any time and any places [2, 3]. Analyzing those physiological time-series data [4] gets a lot of information about our body. Although countries invested heavily in the development of biomedicine, the incidence of various types of chronic noncommunicable diseases is increasing. So the medical community is transitioning towards a proactive direction. Different from the previous one, the approach aims at analyzing physiological time-series data and identifying potential risk of illness and mitigation measures are taken before getting ill.

Many methods can be used to help us get a better understanding of our physical condition. Machine learning [5] is a fundamental and significant research in many fields. It is

widely used in industry [6], power system [7, 8], weather forecast [9], transit systems [10], computer-aided detection and diagnosis systems [11], and so on. Some companies have also launched deep learning related project via collecting and analyzing massive amounts of data and applied to anomaly detection or others applications. It is also an important assistant means for medical and has important application value in the field of medical care [12]. In this paper, we propose a lightweight approach for detecting the anomaly data by analyzing the physiological signals. Empirically, physiological signals can be obtained from biosensors in various ways.

Though anomaly detection is widely used in other fields, the problem of physiological signals anomaly detecting in the context of human-computer interaction still remains complex and largely unexplored. The anomaly detection of physiological signals is, primarily, by using machine learning techniques for learning features [13] from physiological signals and then constructing computational models [14]

of anomaly detection. The main components of the model consists of two parts: processing of input signals (learned features) and detecting anomaly data.

Feature extraction and feature selections are the key in understanding and training an anomaly detection algorithm [15, 16]. Physiological signals are usually correlated to time and space [17]; they belong to high-dimensional time-series data. Time-series data is high-dimensional and complex with unique properties that make them challenge to analyze and model. One of the major challenges in healthy anomaly detection is to extract features in multivariate physiological signals, which can be used to detect the anomaly data correctly [18]. Traditional approaches for modeling sequential data include the estimation of parameters from an assumed time-series model such as autoregressive models and Linear Dynamical Systems (LDS) and the popular Hidden Markov Model (HMM). The estimated parameters can then be used as features in a classifier to perform classification. The restricted Boltzmann machine (RBM) is a generative probabilistic model between input units (visible), x , and latent units (hidden). Several RBMs can be stacked and trained in a greedy manner to form so-called Deep Belief Networks (DBN), which are probabilistic generative neural network composed of multiple layers of restricted Boltzmann machine. DBNs are graphical models which learn to extract a deep hierarchical representation of the training data. Another model that had been used for modeling sequential data is the Recurrent Neural Network (RNN). Generally, an RNN is obtained from the feed forward network by connecting the neurons' output to their inputs. Hand-designed feature extractors require a human expert to find the suitable data manipulations that will lead to good evaluation performance. To determine important features and pick the effective ones to handle a new application may be labor-intensive and time-consuming. It inherit a number of critical limitations that make their use cumbersome in highly complex multimodal input spaces. In this paper, we present a new approach to the automatic physiological signals anomaly detection. The focus is to develop unsupervised feature learning method to learn meaningful feature representations from unlabeled physiological signals. Our hypothesis is that use of nonlinear unsupervised and multivariate Gauss distribution model methods [19–21] relying on the principle of deep learning can eliminate the limitations of the current feature extraction and feature selection in physiological signals anomaly detection. Unsupervised feature learning techniques [4, 22] are a way of learning feature representations [23] that a human expert might not be aware of and could reflect the essence the healthy states feature representations. A secondary contribution of the proposed method is using the features extracting from convolutional neural network (CNN) feed to anomaly detection model and giving the anomaly data to doctor to evaluate mitigating risks before illness. CNN's convolution and pooling can help us deal with high-dimensional data more quickly. By analyzing physiological signals, if anomaly data occurs, the special prevention should be done in advance to reduce the risk of disease.

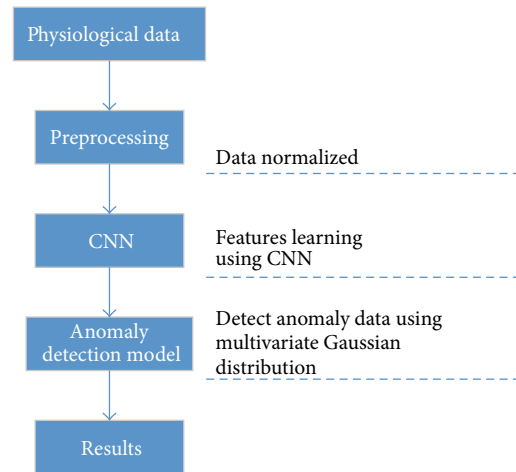


FIGURE 1: The block diagram of the algorithm.

2. Materials and Methods

In this section, we describe a common framework used for feature learning. For concreteness, we will focus on the application of these algorithms to learn features from physiological signals, though our approach is applicable to other forms of data as well. Furthermore, the studies cover the two key research pillars of this paper: (1) defining feature set to extract relevant bits of information from objective data signals; (2) creating models that map a feature set into multivariate Gaussian anomaly detection model to predict the anomaly physiological signals.

At a high level, our algorithm performs the following steps (see Figure 1) to learn feature representation:

- (a) Dividing physiological signals into a number of segments from unlabeled training data.
- (b) Applying a preprocessing stage to the segments and normalizing the raw data.
- (c) Extracting high level information using an unsupervised learning algorithm.
- (d) Using Gauss model to detect the anomaly physiological signals.

Now we describe the components of this pipeline and its parameters in more detail.

2.1. Feature Learning. In the context of healthy state anomaly detection, feature learning refers to the process of transforming the raw signals captured by the hardware into a set of inputs suitable for a computational evaluation of anomaly data. Usually, learning feature from one-dimensional continuous signals is simple statistical features [24, 25], such as average and standard deviation values, calculated on the time or frequency domains of the raw or the normalized signals. Physiological signals anomaly detection based on signals with more than one dimension typically boils down to physiological anomaly data detection from all kinds of physiological signals. The focus of this paper is on convolutional

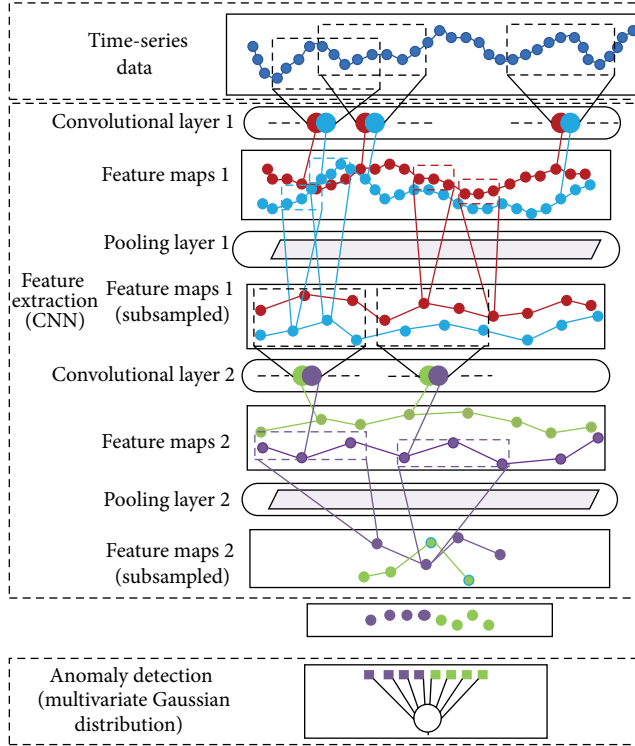


FIGURE 2: The structure of deep ANN architecture.

neural networks [5, 13, 26] methods that can automatically extract new features or unknown features in an unsupervised manner from those data.

Convolutional neural networks, as a popular technique, could be used in many fields such as image and video classification natural language processing, pedestrian detection, generic visual recognition, face recognition, and image recognition [10, 27]. They are very similar to ordinary neural networks. It is composed of a number of neurons that have learnable weights and biases. Each neuron receives some inputs and performs a dot product with a nonlinearity function Sigmoid. A convolutional neural network is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of an ANN is designed to include both the feature extractor and the anomaly detection, as shown in Figure 2.

Usually, in convolutional neural networks, each layer is composed of two operations: convolution and max-pooling. At a convolution layer, the previous layer feature maps are convolved with learnable kernels and then put through the activation function to form the output feature map. Each output map may combine convolutions with multiple input maps. Supposing X denote the input signal is 2-dimensional data of size $m \times n$, where m and n are positive integers. A feature map is obtained by convolution of the input signal with a linear filter, adding a bias term and then applying a nonlinear function f . If we denote the k th feature map at a given layer as h^k , whose filters are determined by the weights

w^k and bias b^k , M_j means the selection of input maps in layer $k - 1$. Then the feature map h^k is obtained as follows:

$$h_j^k = f \left(\sum_{i \in M_j} X_i^{k-1} * W_{ij}^k + b^k \right). \quad (1)$$

Sensitivity computation process:

$$\delta_j^k = \beta_j^{k+1} (f' \circ \text{up}(\delta_j^{k+1})). \quad (2)$$

$\text{up}(\cdot)$ function is exactly an inverse process of the downsampling.

Compute the gradient,

$$\frac{\partial J}{\partial W_{ij}^k} = \sum \delta_j^k * p_i^{k-1}, \quad (3)$$

$$\frac{\partial J}{\partial b_j^k} = \sum \delta_j^k.$$

$J(\cdot)$ is cost function. The new variable p_i^{k-1} here, which means the patch in X_i^{k-1} that was multiplied element wise by W_{ij}^k .

Gradients in subsampling layers:

$$h_j^k = f(\beta_j^k \text{down}(h_j^{k-1}) + b_j^k). \quad (4)$$

Here, $\text{down}(\cdot)$ means the subsampling function, such as average-pooling or max-pooling.

$$\delta_j^k = f' \circ q_j^k. \quad (5)$$

q_j^k is a matrix which has the same size of δ_j^k , and its element is the convolution between all sensitivities of nodes in layer $k+1$ that have connection with the node in layer k and the weights are defined as W_j^k , and we could compute the gradients for β and b in subsampling layers as the following equations:

$$\frac{\partial J}{\partial \beta_j^k} = \sum \delta_j^k \circ \text{down}(h_j^k), \quad (6)$$

$$\frac{\partial J}{\partial b_j^k} = \sum \delta_j^k.$$

The input of each neuron determines the size of patch. Each neuron contains a number of trainable weights equal to the number of its inputs and additional bias parameter; the output is calculated by applying an activation function to the weighted sum of the input and bias. (Activation function: tanh, ReLU, Sigmoid, and Softplus. tanh: $y = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$. The real worth to compress input ranges in -1 to 1 , so that it is substantially 0 mean; ReLU: $y = \max(0, x)$. The network can be introduced into the sparsity, and the performance of ReLU is better than other activation functions in the case of no pretraining; Sigmoid: $y = (1 + \exp(-x))^{-1}$. The activation function in the neural network learning can push towards key features to the central and the nonkey features to both sides of the

zone; Softplus: $y = \log(1 + \exp(x))$. The activation function makes the relationship between output and input keep the nonlinear monotonic rise and fall and fault tolerance of neural network is good.) Each neuron scans sequentially the input, assessing at each patch location the similarity to the pattern encoded on weights. The consecutive outputs generated at every location of the input assemble a feature map. The output of convolution layers is the set of feature maps repeated application of a nonfunction across subregions of entire input. In the following, we provide details on CNN architectures in Figure 2. The neurons in the convolutional layer take as input a patch 8×1 on the input time-series data. The input patch is applied with a sliding-window stride of 8 along time-axis. Each of the neurons calculates h^k and then obtains features.

As soon as feature maps have been generated, a pooling layer aggregates consecutive values of the feature maps resulting from the previous convolution layer, reducing their resolution with a pooling function. The maximum or average values are the two most commonly used pooling functions providing max-pooling and average-pooling layers, respectively. It is common to periodically insert a pooling layer in between successive convolutional layer in a ConvNet architecture. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network and hence to also control overfitting. In this paper, the pooling layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2×2 applied with a stride of 2 downsamples in every depth slice in the input by 2 along both width and height, discarding 75% of the activation. Because physiological signals are time-series data, we designed the filters of size 2×1 with a stride of 2 downsamples along time-axis.

As we described above, a simple CNN is a sequence of layers, and each layer of a CNN transforms one volume of activation to another through a differentiable function. We use three main types of layers to build CNN architectures: convolutional layer, pooling layer, and fully connected layer (exactly as seen in regular neural networks). In our work, it is a hierarchical model that alternate convolution and pooling layers in order to process large input spaces in which a spatial or temporal relation among the input exists such as time-series data, speech, or physiological signals. Therefore, hierarchical analysis and learning architectures are the key to success in anomaly detection.

2.2. Autoencoders. We must train our network weights with a kind of unsupervised method because our training set is unlabeled. Usually physiological signals are unlabeled, so we need to take other methods to train ConvNet weights. An autoencoder neural network (see Figure 3) is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs [28]. In this paper, we use autoencoders to train all convolution layers of our CNN. Now we have only a set of unlabeled training examples $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots\}$, where $x^{(i)} \in R^n$. An autoencoder takes an input $x \in [1, 0]^d$ and first maps it (with an encoder) to

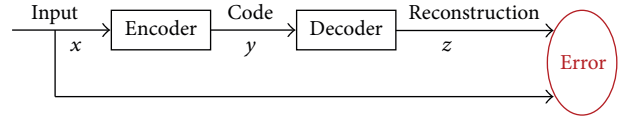


FIGURE 3: The block diagram of autoencoders.

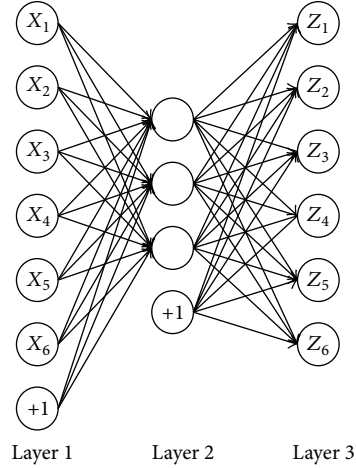


FIGURE 4: The structure of autoencoder.

a hidden representation $y \in [1, 0]^d$ through a deterministic mapping:

$$y = s(Wx + b), \quad (7)$$

where s is a nonlinearity function Sigmoid. The latent representation y or code is then mapped back (with a decoder) into construction z of the same shape as x . The mapping happens through a similar transformation:

$$z = s(W'y + b'). \quad (8)$$

z should be seen as a prediction of x , given the code y .

An autoencoder is a model that transforms an input space into a new distributed representation by applying a deterministic parameterized function called the encoder (see Figure 4). The autoencoder learns how to map the output of the encoder into the input space, with a parameterized decoder, to have small reconstruction error on the training examples; that is, the original and corresponding decoded inputs are similar. The encoder weights (used to obtain the output representation) are also used to reconstruct the input. By defining the reconstruction error as the sum of squared differences between the inputs and the reconstructed input, then use gradient descent method such as backpropagation to train the weights of the ConvNet. The reconstruction error can be measured in many ways, depending on the appropriate distributional assumptions on the input given the code. The traditional squared error cost function is given by

$$J(xz) = \|x - z\|^2. \quad (9)$$

Here x is raw data and z is reconstruction data.

In this paper, autoencoders are used to train unsupervised CNN to transpose subsets of the raw input signals into learned features. In turn, the outputs of learned features extracted from the input layer may feed any function approximation or classifier that attempts to find a mapping between the input signal and a target output. In this paper, we use multivariate Gaussian anomaly detection model to detect the anomaly physiological signals for a user based on the learned features of his physiological signals.

2.3. Multivariate Gaussian Distribution. The multivariate Gaussian distribution is a generalization of the univariate normal to two or more variables. It is a distribution for random vectors of correlated variables, each element of which has a univariate normal distribution. A vector-valued random variable $X = [X_1, X_2, \dots, X_n]^T$ is said to have a multivariate Gaussian distribution with mean vector $\mu \in R^n$ and covariance matrix Σ . Its probability density function is given by

$$p(x, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{-1}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right), \quad (10)$$

$$\Sigma = E[(X - \mu)(X - \mu)^T].$$

We write this as $X \sim N(\mu, \Sigma)$.

Anomaly detection is an unsupervised learning method, using density estimation to evaluate data is normal or not. The expression is as follows:

$$\text{if } p(x) \begin{cases} \leq \varepsilon & \text{anomaly} \\ > \varepsilon & \text{normal.} \end{cases} \quad (11)$$

When $p(x)$ is greater than threshold ε , the data is normal and $p(x)$ is less than threshold ε , and the data is anomaly.

3. Results and Discussion

In the experiments, we focused on evaluating the efficacy of using CNN to construct a model of physiological signals anomaly detection and we test our algorithm on eight physiological signals on DEAP dataset, a dataset for emotion analysis using EEG, physiological, and video signals. We expect that information relevant to anomaly detection can be extracted more effectively using CNN methods directly on the raw physiological signals automatic selection of features than on a set of designer-selected extracted features. The hardware and software environment in experiment are as follows: hardware environment: Intel(R) Core(TM) i3-2330 CPU @ 2.2 GHz RAM 2.00 GB; software environment: Windows 7, Python 2.7, and Matlab R2014a.

3.1. Training Models of Physiological Signals Anomaly Detection. In the approach presented here, we investigate an effective method of learning models that map signals of user physiological to detect anomaly data. In the feature extraction stage, we use a deep model composed from a multilayer convolutional neural network that transforms the raw signals into reduced set features. In the anomaly detection stage, we feed those features to an anomaly detection model which uses the multivariate Gaussian distribution to detect anomaly physiological signals (see Figure 2). Before new unlabeled time-series physiological signals enter the model, first, make the time-series physiological signals normal. Then extract features in the original network parameters. Then last, use multivariate Gaussian distribution to detect anomaly data in new unlabeled time-series physiological signals. The deep ANN architecture contains two convolutional layers, two pooling layers, and a multivariate Gaussian anomaly detection model. The first convolutional layer (patch length of 12 raw physiological signals) processes physiological signals, which is then propagated forward to a maximum-pooling layer (window length of 2 features). The second convolutional layer (patch length of 5 subsampled features) processes the subsampled feature maps and the resulting feature maps of the second pooling layer (window length of 2 features). The final subsampled feature maps form the outputs of the CNN which provides a number of learning features feeding the input of Gaussian anomaly detection model. Our hypothesis is that the automation of feature extraction via deep learning will yield anomaly physiological signals of higher predictive power, which, in turn, deliver evaluation models of higher accuracy.

Before feeding the raw data to CNN, in order to cause the reconstruction error convergence, we normalized the raw data use:

$$X_{\text{norm}} = \frac{X - \mu}{\text{std}(X)}. \quad (12)$$

μ and std , respectively, represent the average value of raw data and the standard deviation of raw data. Then feed the normalized data X_{norm} to CNN learning features of raw data. By analyzing the reconstruction error J_w between the inputs data X and the reconstruction inputs Z and the number of iterations, we can obtain preferable learned features. In theory, the greater the iterations number, the smaller the reconstruction error J_w .

With the increase of the number of iterations, the cost function J_w is tending towards zeros and then keeping stability, we can draw a conclusion that the reconstruction data are nearly the same. Therefore, the learned features from raw data are effective and we can view those features as high level expression of the raw data. As Figure 5 shows, we take 5000 iterations, the cost function J_w is tending to zeros, and we can get the weights of the ConvNet and high level features of the eight physiological signals.

As soon as features have been learned, feed them to Gaussian mixtures models to detect the anomaly features and get the coordinate. We consider the data and the learned features to be subject to Gauss distribution.

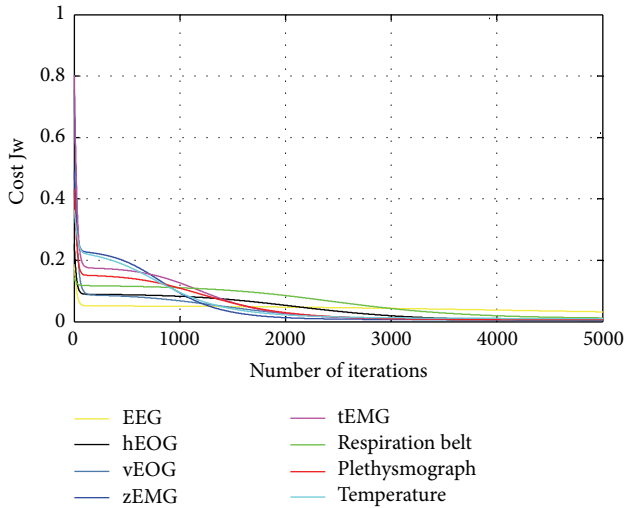


FIGURE 5: Learned features of the best-performing convolutional neural networks. Lines plotted are the values of cost function for each physiological signal. The x -axis displays the number of iterations.

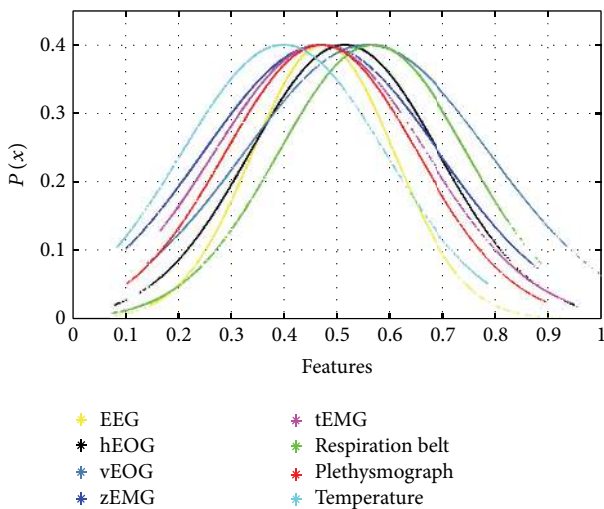


FIGURE 6: Gauss distribution of features.

As Figure 6 shows, the features of the eight physiological signals are subject to Gauss distribution signals.

3.2. Results. Then we choose a fine threshold ε to get the features which is $p(x_i) \leq \varepsilon$ and then get the coordinate and the corresponding raw data. All anomaly detection algorithms can be set at different threshold ε , which may make results correspond to reality. If the ratio of anomaly data is 1% and threshold $\varepsilon = 0.2$ we can get the anomaly features and the coordinate of raw physiological signals. In addition, if the ratio of anomaly data is 5% and threshold $\varepsilon = 0.3$ we can get other anomaly features and the coordinate of raw physiological.

In this paper, we use a series of detection thresholds ε 0.2, 0.23, 0.25, and 0.26, and the ratios of anomaly data are 1%, 2%, 3%, and 5%, respectively. The percentage controls the number of physiological signals to be considered anomaly ranked by severity. We get four sets of anomaly data as Figure 7 shows, (a) and (b), (c) and (d), (e) and (f), and (g) and (h). Finally, it is easy to obtain the anomaly raw physiological signals according to anomaly features point coordinate and doctor can quickly analyze those anomaly physiological signals to help users understand the healthy states at present.

3.3. Discussion. The testing on dataset showed that the method can detect anomaly physiological signals and some may exhibit early signs of illness. Therefore, the method could be a tool to help doctor identify the underlying disease.

There is no “medical instance” for performance evaluation or a benchmark dataset in which every physiological signal is labeled “normal or anomaly” in definitive terms. In this paper, the detection threshold ε is artificially chosen to evaluate physiological signals normally or anomaly. Therefore, one limitation of this paper is that we do not provide an evaluation of the detection accuracy in a traditional sense, such as false negatives and false positives. A new method is needed to evaluate the performance of an anomaly detection method that does not rely on preexisting criteria but is capable of detecting unknown issues.

4. Conclusions

This paper introduces the application of deep learning to the construction of an anomaly detection model built on physiological signals manifestations of anomaly data. To detect the anomaly data, key point is learning effective features in the raw feature. The algorithm proposed employs a number of convolution layers that learn to extract relevant features from the input signal and then feed those features to multivariate Gaussian distribution to detect anomaly features. The algorithm was tested on eight physiological signals. Result, in general, suggests that algorithm is highly efficient to learn high level features from raw physiological signals and multivariate Gaussian distribution anomaly detection.

Feature learning directions in our algorithm can be outlined as follows. First, a wide range of datasets each of which has different characteristics from different part of body should be employed in order to demonstrate the effectiveness of the method. Second, since the database is a key in our method, the collecting of data is still going on. We cannot really evaluate the results because of the database being unlabeled. Therefore, collecting of some labeled physiological signals is a way to enhance the performance of the algorithm. Further research is needed to comprehensively evaluate the performance of the algorithm in detecting unknown issues. Based on this paper, future research can be from the following aspects: first, collecting of some labeled physiological signals to comprehensively evaluate and improve the performance of the algorithm; second, because of lack of comparative tests, it is necessary to do comparative tests with other algorithms to verify the performance of the algorithm.

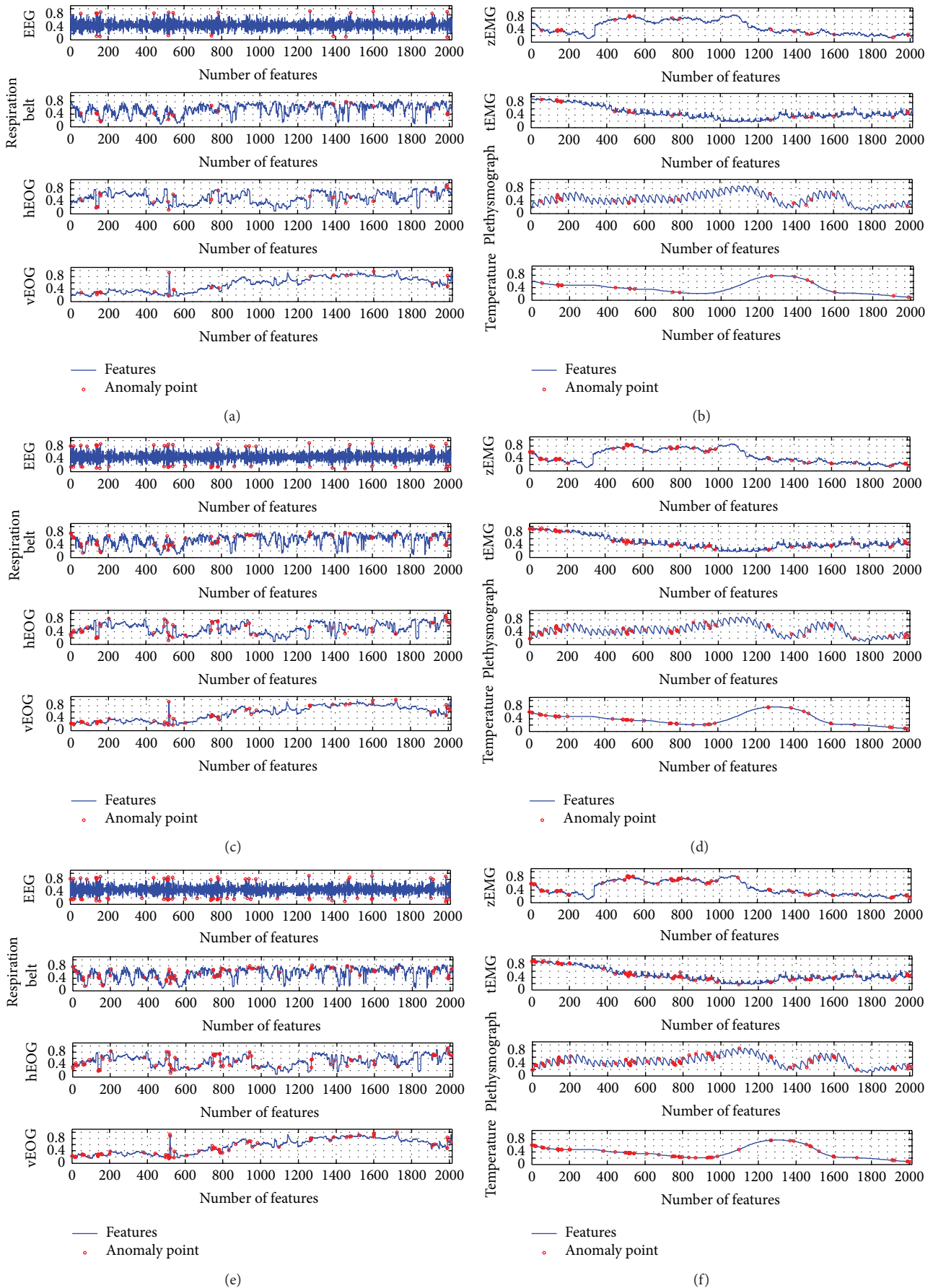


FIGURE 7: Continued.

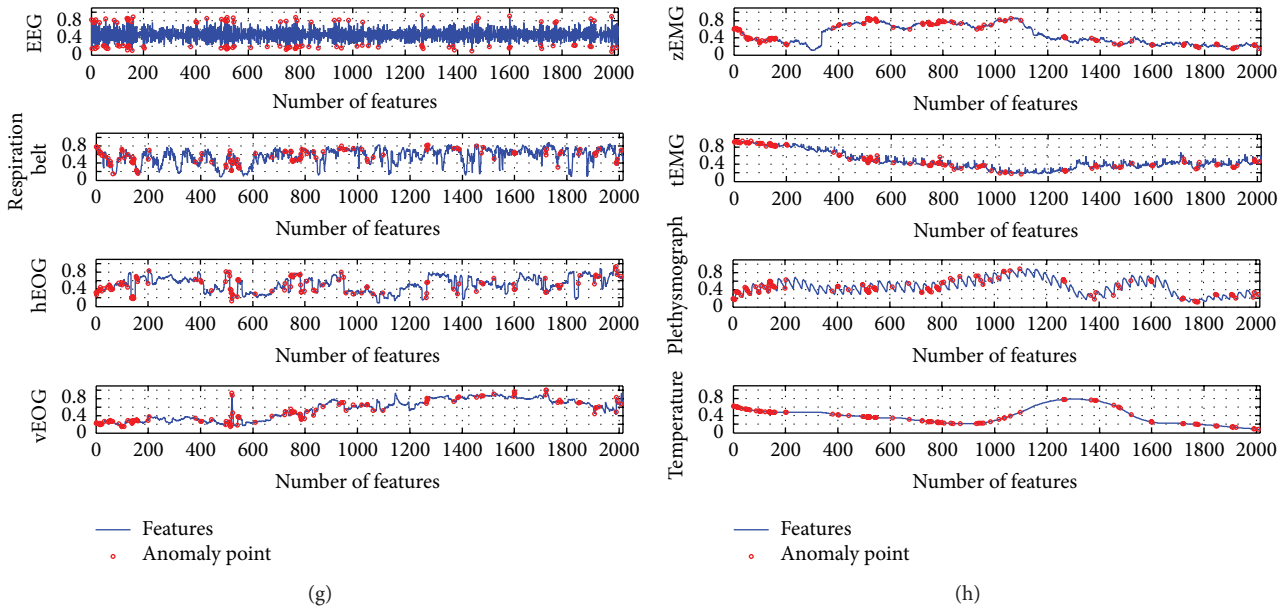


FIGURE 7: Anomaly features detection with different thresholds.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

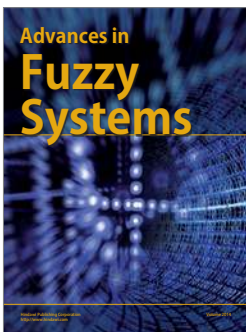
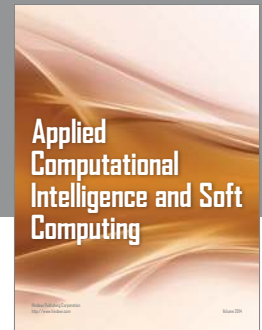
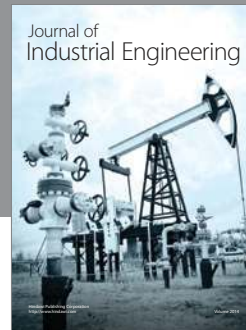
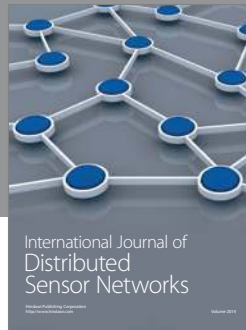
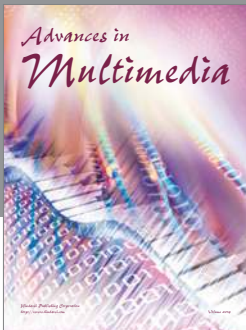
Acknowledgments

This work is supported by Major State Basic Research Development Program of China (973 Program) under Project no. 2013CB328903 and by State Scholarship Fund of China (CSC) under File no. 201306055018. The physiological signals are provided by DEAP dataset, and the URL is <http://www.eecs.qmul.ac.uk/mmv/datasets/deap/>.

References

- [1] A. Szczepański and K. Saeed, "A mobile device system for early warning of ECG anomalies," *Sensors*, vol. 14, no. 6, pp. 11031–11044, 2014.
- [2] H. Zhao, R. Xu, M. Shu, and J. Hu, "Physiological-signal-based key negotiation protocols for body sensor networks: a survey," *Simulation Modelling Practice and Theory*, vol. 65, pp. 32–44, 2016.
- [3] T. Li, Z. Wang, D. Zhao, and L. Huang, "Development of a portable multi-channel system for plant physiological signal recording," *Information Processing in Agriculture*, vol. 3, no. 2, pp. 124–132, 2016.
- [4] M. Långkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, no. 1, pp. 11–24, 2014.
- [5] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [6] M. Zhang, X. Liu, and Z. Zhang, "A soft sensor for industrial melt index prediction based on evolutionary extreme learning machine," *Chinese Journal of Chemical Engineering*, vol. 24, pp. 1013–1019, 2016.
- [7] J. Heinermann and O. Kramer, "Machine learning ensembles for wind power prediction," *Renewable Energy*, vol. 89, pp. 671–679, 2016.
- [8] P. Tang, D. Chen, and Y. Hou, "Entropy method combined with extreme learning machine method for the short-term photovoltaic power generation forecasting," *Chaos, Solitons & Fractals*, vol. 89, pp. 243–248, 2016.
- [9] M. Gocic, D. Petković, S. Shamshirband, and A. Kamsin, "Comparative analysis of reference evapotranspiration equations modelling by extreme learning machine," *Computers and Electronics in Agriculture*, vol. 127, pp. 56–63, 2016.
- [10] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.
- [11] J. Y. Choi, D. H. Kim, K. N. Plataniotis, and Y. M. Ro, "Classifier ensemble generation and selection with multiple feature representations for classification applications in computer-aided detection and diagnosis on mammography," *Expert Systems with Applications*, vol. 46, pp. 106–121, 2016.
- [12] W. Huang, S. Zeng, M. Wan, and G. Chen, "Medical media analytics via ranking and big learning: a multi-modality image-based disease severity prediction study," *Neurocomputing*, vol. 204, pp. 125–134, 2016.
- [13] I. Mrazova and M. Kukacka, "Can deep neural networks discover meaningful pattern features?" *Procedia Computer Science*, vol. 12, pp. 194–199, 2012.
- [14] S. Aras and T. D. Kocakoç, "A new model selection strategy in time series forecasting with artificial neural networks: IHTS," *Neurocomputing*, vol. 174, pp. 974–987, 2016.
- [15] I. J. Goodfellow, D. Erhan, P. L. Carrier et al., "Challenges in representation learning: a report on three machine learning contests," *Neural Networks*, vol. 64, pp. 59–63, 2015.
- [16] K. Kim, H. Lin, J. Y. Choi, and K. Choi, "A design framework for hierarchical ensemble of multiple feature extractors and multiple classifiers," *Pattern Recognition*, vol. 52, pp. 1–16, 2016.

- [17] A. L. V. Coelho and C. A. M. Lima, "Assessing fractal dimension methods as feature extractors for EMG signal classification," *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 81–98, 2014.
- [18] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.
- [19] A. Aiyer, K. Pyun, Y.-Z. Huang, D. B. O'Brien, and R. M. Gray, "Lloyd clustering of Gauss mixture models for image compression and classification," *Signal Processing: Image Communication*, vol. 20, no. 5, pp. 459–485, 2005.
- [20] B. Konomi, G. Karagiannis, and G. Lin, "On the Bayesian treed multivariate Gaussian process with linear model of coregionalization," *Journal of Statistical Planning and Inference*, vol. 157/158, pp. 1–15, 2015.
- [21] A. O'Hagan, T. B. Murphy, I. C. Gormley, P. D. McNicholas, and D. Karlis, "Clustering with the multivariate normal inverse Gaussian distribution," *Computational Statistics & Data Analysis*, vol. 93, pp. 18–30, 2016.
- [22] M. Siddula, F. Dai, Y. Ye, and J. Fan, "Unsupervised feature learning for objects of interest detection in cluttered construction roof site images," *Procedia Engineering*, vol. 145, pp. 428–435, 2016.
- [23] B. Zoefel and R. VanRullen, "EEG oscillations entrain their phase to high-level features of speech sound," *NeuroImage*, vol. 124, pp. 16–23, 2016.
- [24] R. Jegadeeshwaran and V. Sugumaran, "Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines," *Mechanical Systems and Signal Processing*, vol. 52–53, no. 1, pp. 436–446, 2015.
- [25] N. Sengupta, M. Sahidullah, and G. Saha, "Lung sound classification using cepstral-based statistical features," *Computers in Biology and Medicine*, vol. 75, pp. 118–129, 2016.
- [26] K. Audhkhasi, O. Osoba, and B. Kosko, "Noise-enhanced convolutional neural networks," *Neural Networks*, vol. 78, pp. 15–23, 2016.
- [27] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.
- [28] Y. Wang, Z. Xie, K. Xu, Y. Dou, and Y. Lei, "An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning," *Neurocomputing*, vol. 174, pp. 988–998, 2016.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

