SCIENCE CHINA
Information Sciences

• **Supplementary File** •

A Sparse Autoencoder-Based Approach for Cell Outage Detection in Wireless Networks

Ma Ziang[1], Pan Zhiwen[1,2*] & Liu Nan[1]

[1]*National Mobile Communications Research Laboratory, Southeast University, Nanjing, Jiangsu* 210096, *China;*
[2]*Purple Mountain Laboratories, Nanjing, Jiangsu* 211111, *China*

Appendix A System Model

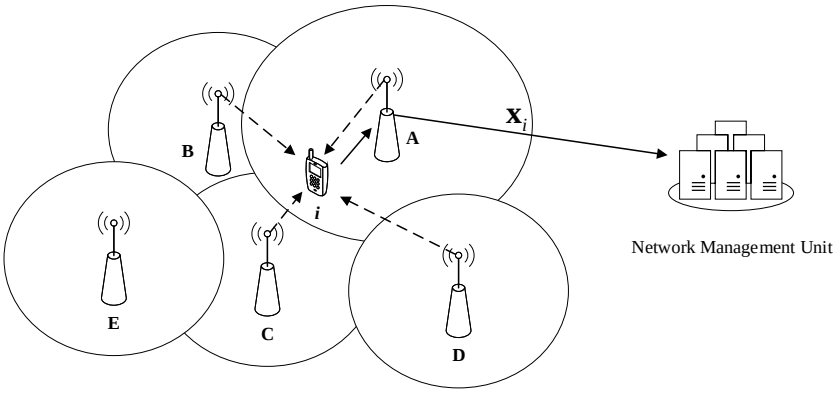

Figure A1 System model

Appendix B Proposed Algorithm

The proposed algorithm is shown in Fig. B1, V and U denote the outputs of SMOTE and the outputs in the hidden layer of sparse autoencoder [2] respectively.

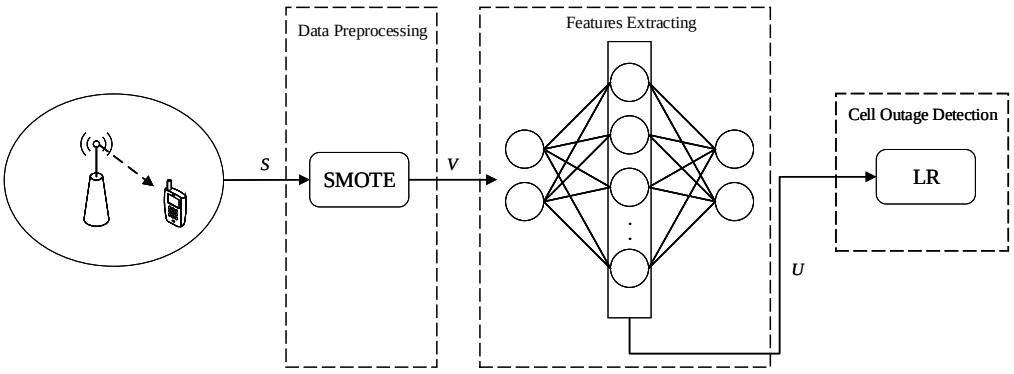

Figure B1 The proposed algorithm

* Corresponding author (email: pzw@seu.edu.cn)

Appendix B.1 Synthetic Minority Over-sampling Technique-Based Data Preprocessing

Due to the lack of samples with cell outage events, SMOTE [1] is firstly employed to preprocess the original dataset S by oversampling. Denote S_0 as the minority samples (*i.e.* samples with cell outage events) dataset. For each sample $\mathbf{x}_i$ in S_0, the SMOTE algorithm selects K nearest samples according to Eq. (B1).

$$\begin{aligned} \mathbf{x}_k &= \underset{\mathbf{x}_k \in S_0, k \neq i}{\arg\min} \|\mathbf{x}_k - \mathbf{x}_i\|_2 \\ S_0 &= S_0 - \{\mathbf{x}_k\} \\ T &= T \cup \{\mathbf{x}_k\} \end{aligned} \tag{B1}$$

where T is the set containing K nearest samples of $\mathbf{x}_i$.

Then, for each sample $\mathbf{x}_j$ in T, the new sample $\mathbf{x}_{new}$ is generated according to Eq. (B2).

$$\mathbf{x}_{new} = \mathbf{x}_i + \alpha * (\mathbf{x}_j - \mathbf{x}_i) \tag{B2}$$

where α is a uniformly distributed random variable from 0 to 1.

Fig. B2 briefly shows the process for processing dataset S using SMOTE. The SMOTE algorithm is summarized in Algorithm B1, where operation $|S_0|$ denotes the total number of elements in set S_0.

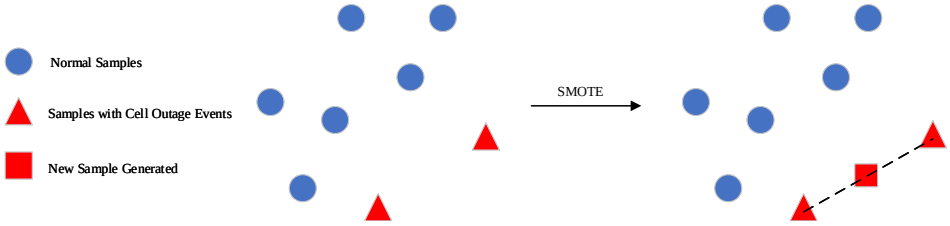

Figure B2 Dataset preprocessing using SMOTE

Algorithm B1 SMOTE

Input: dataset S, number of nearest samples K.
Output: the preprocessed dataset V.

1: Divide S into two subsets according to the label of the samples: S_0 and S_1, where S_0 is the minority samples dataset;
2: **for** each sample $\mathbf{x}_i$ in S_0 **do**
3: Select K nearest samples in S_0 to $\mathbf{x}_i$, obtain set T according to Eq. (B1);
4: **for** each sample $\mathbf{x}_j$ in T **do**
5: Generate $\mathbf{x}_{new}$ according to Eq. (B2);
6: Update S_0: $S_0 = S_0 \cup \{\mathbf{x}_{new}\}$;
7: **if** $|S_0| \geqslant |S_1|$ **then**
8: **return** $V = S_0 \cup S_1$.
9: **end if**
10: **end for**
11: **end for**

Appendix B.2 Sparse Autoencoder-Based Data Features Extracting

Predictably, there is still room for improvement since SMOTE is only an oversampling method which does not change the "primitive structure" of samples. In this paper, after obtaining the preprocessed dataset V via SMOTE, sparse autoencoder is used to extract the high-level features of original samples characterized by RSRP and SINR. It yields a funtion $f : \mathbb{R}^{\dim \mathbf{v}} \to \mathbb{R}^{s_2}$ that transforms the input $\mathbf{v}$ to a high-level representation $\mathbf{u}$ shown in Eq. (B3), where s_2 is the number of hidden-layer neurons.

$$\mathbf{u} = f(\mathbf{v}) \in \mathbb{R}^{s_2} \tag{B3}$$

Sparse autoencoder is one kind of feed-forward neural network [4]. It is noted that sparse autoencoder has more complex structure such as stacked sparse autoencoder [5], but for simplicity, in this paper the classical three-layer sparse autoencoder (shown in Fig. B3) is used, where the activation function $a(x)$ is sigmoid function.

$$a(x) = \frac{1}{1 + e^{-x}} \tag{B4}$$

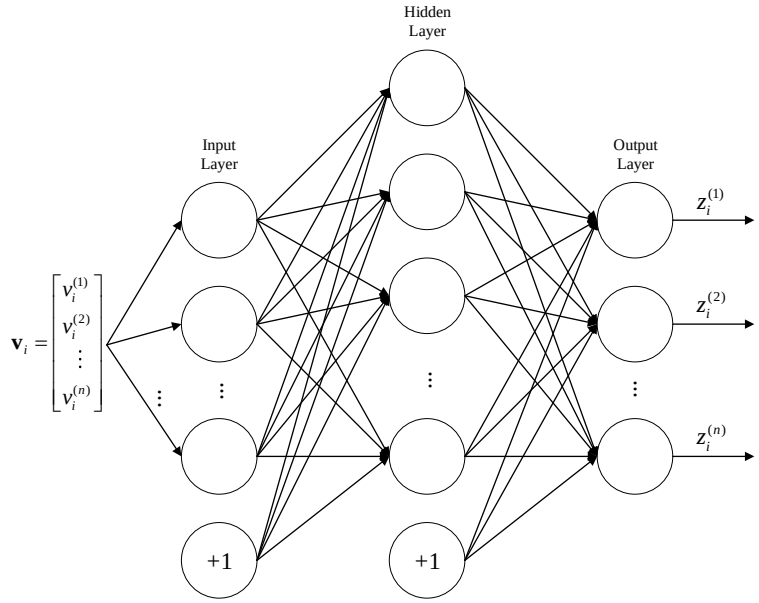

Figure B3 The three-layer sparse autoencoder

Table B1 Enumeration of the symbols used in sparse autoecnoder

Symbol	Description
$\mathbf{W}^{(l)}$	weight matrix between the l-th layer and the $(l+1)$-th layer
w_{ji}^{l}	the weight between the i-th neuron in the l-th layer and the j-th neuron in the $(l+1)$-th layer
$\mathbf{b}^{(l)}$	bias vector between the l-th layer and the $(l+1)$-th layer
b_j^l	bias between the bias unit (*i.e.* the neuron marked "+1" in Fig. B3) in the l-th layer and the j-th neuron in the $(l+1)$-th layer
N	number of elements in V
$\mathbf{v}_i$	the i-th element in V
$\mathbf{z}_i$	output of the sparse autoencoder for the input $\mathbf{v}_i$
λ	regularization coefficient used to reduce weights to decrease overfitting [7]
s_l	number of neurons in the l-th layer
β	weight of the penalty factor $\sum_{j=1}^{s_2} \mathrm{KL}(\rho\|\|\rho_j)$
ρ	sparsity parameter indicating the desired activation degree of each hidden-layer neuron
ρ_j	average activation degree of the j-th hidden-layer neuron for all inputs
$a_j^{(2)}(\mathbf{v}_i)$	output of the j-th hidden-layer neuron under the condition of input $\mathbf{v}_i$

The process of training sparse autoencoder is to adapt the weight matrices $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ and bias vectors $\mathbf{b}^{(1)}$, $\mathbf{b}^{(2)}$ to minimize the following cost function, which can be solved by back propagation algorithm [6].

$$\mathrm{J}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|\mathbf{v}_i - \mathbf{z}_i\right\|_2^2 + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(w_{ji}^{l}\right)^2 + \beta\sum_{j=1}^{s_2}\mathrm{KL}(\rho\|\|\rho_j) \tag{B5}$$

where,

$$\mathrm{KL}\left(\rho\|\|\rho_j\right) = \rho\ln\frac{\rho}{\rho_j} + (1-\rho)\ln\frac{1-\rho}{1-\rho_j}, \rho_j = \frac{1}{N}\sum_{i=1}^{N}a_j^{(2)}(\mathbf{v}_i) \tag{B6}$$

The symbols in Eq. (B5) and Eq. (B6) are summarized in Table B1.

The purpose of minimizing Eq. (B5) is to adapt the weight matrices $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ and bias vectors $\mathbf{b}^{(1)}$, $\mathbf{b}^{(2)}$ such that inputs are restored at the output layer as much as possible (*i.e.* $\mathbf{v}_i \approx \mathbf{z}_i$), which is so called "autoencoding".

Sparse autoencoder extracts high-level features of inputs in the hidden layer by minimizing Eq. (B5). For each input $\mathbf{v}_i$, the output $\mathbf{u}_i$ in the hidden layer is shown in Eq. (B7), which will be used for cell outage detection.

$$\mathbf{u}_i = a(\mathbf{W}^{(1)^T}\mathbf{v}_i + \mathbf{b}^{(1)}) \tag{B7}$$

Appendix B.3 Logistic Regression-Based Cell Outage Detection

After training the sparse autoencoder, outputs of its hidden layer are utilized to detect cell outages via Logistic Regression [3] (LR).

LR is a classic classification model, which is to determine the maximum value of the log-likelihood function defined as follows:

$$\mathrm{L}(\mathbf{h}, c) = \sum_{i=1}^{N} y_i(\mathbf{h}\cdot\mathbf{u}_i + c) - \ln(1 + e^{\mathbf{h}\cdot\mathbf{u}_i + c}) \tag{B8}$$

where $\mathbf{u}_i$ represents the i-th input, y_i (same as section 2) indicates the label of $\mathbf{v}_i$. y_i has two possibile values: 1 or 0. $\mathbf{h}$ is the weight vector and c is the bias.

The maximum value of Eq. (B8) can be found by gradient descent algorithms [8]. Denote the solution of maxmizing Eq. (B8) as $\mathbf{h}_{opt}$ and c_{opt}.

The proposed algorithm is summarized in Algorithm B2.

Algorithm B2 Cell outage detection algorithm based on sparse autoencoder

Input: dataset S, number of nearest samples K, regularization coefficient λ, weight of penalty factor β, sparsity parameter ρ and number of hidden-layer neurons s_2.

Output: two variables determining the LR model: $\mathbf{h}_{opt}$ and c_{opt}.

1: Obtain the preprocessed dataset V by SMOTE algorithm;
2: Take $\mathbf{v}_i \in V$ as input to train the sparse autoencoder, let $U = \emptyset$;
3: **for** each sample $\mathbf{v}_i$ in V **do**
4: Obtain the output $\mathbf{u}_i$ of the hidden layer in the sparse autoencoder according to Eq. (B7);
5: Update U: $U = U \cup \{\mathbf{u}_i\}$;
6: **end for**
7: Train LR model: taking $\mathbf{u}_i \in U$ as input, determine the maximum value of Eq. (B8);
8: Denote the weight vector and bias after training as $\mathbf{h}_{opt}$ and c_{opt} respectively;
9: **return** $\mathbf{h}_{opt}$ and c_{opt}.

Appendix C More simulation Results

The more detailed parameter configuration is listed in Table C1, where d (in meter) is the distance between the base station and the user.

The effect of the neuron number on performance is stuied. Under the same sample size, Fig. C1 illustrates the detection rate versus the number of hidden-layer neurons. As the number of neurons increases, the performance for each neuron number improves. We can see that it is essential to combine SMOTE and sparse autoencoder since under the similar performance SMOTE greatly reduces the number of hidden-layer neurons in sparse autoencoder, which saves the training time of it, as shown in Fig. C2.

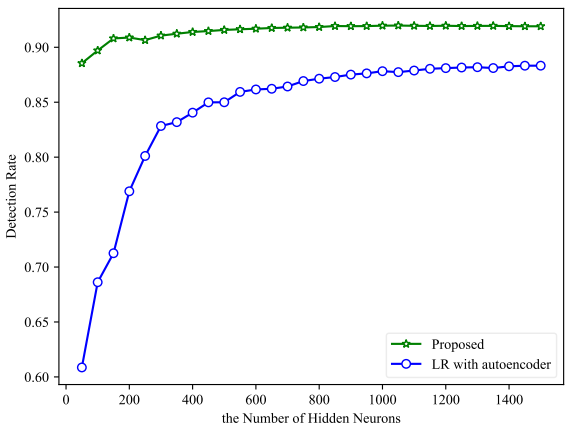

Figure C1 Detection rate versus number of hidden neurons

Table C1 Simulation Parameters

Parameter	Value
Transmit Power	16 dBm
Carrier Frequency	5G Hz
Channel Bandwidth	5M Hz
Thermal Noise Power	-174 dBm/Hz
Minimum Distance between Base Stations and Users	2 m
Pathloss Model	$40.4 + 22 * \lg(d)$ dB [10]
Shadow Fading Standard Deviation	6.8 dB [10]
Shadow Fading Decorrelation Distance [9]	5 m
Shadow Fading Cross-correlation between Base Stations [9]	0.5
Minimal Sensitive Signal Strength of Devices [11]	-112.5 dBm
Number of Nearest Samples K	1
Regularization Coefficient λ	$3 * 10^{-3}$
Weight of Penalty Factor β	3
Sparsity Parameter ρ	0.1
Number of Hidden-layer Neurons s_2	250

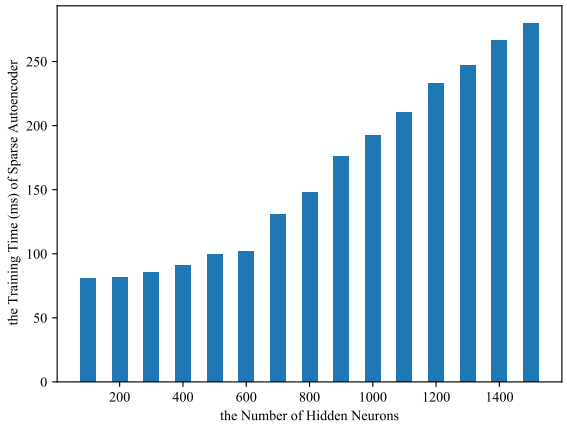

Figure C2 Training time versus number of hidden neurons

References

1 Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 2002, 16: 321-357
2 Ng A. Sparse autoencoder. CS294A Lecture notes, 2011, 72(2011): 1-19
3 Bishop C M. Pattern recognition and machine learning. New York: Springer-Verlag, 2006. 205-206
4 Svozil D, Kvasnicka V, Pospichal J. Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems, 1997, 39(1): 43-62
5 Xu J, Xiang L, Liu Q, et al. Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE transactions on medical imaging, 2015, 35(1): 119-130
6 Van Ooyen A, Nienhuis B. Improving the convergence of the back-propagation algorithm. Neural networks, 1992, 5(3): 465-471
7 Hawkins D M. The problem of overfitting. Journal of chemical information and computer sciences, 2004, 44(1): 1-12
8 Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016
9 Zhang C, Chen X, Yin H, et al. Two-dimensional shadow fading modeling on system level. In: 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). Sydney, 2012. 1671-1676
10 Goldsmith A. Wireless communications. Cambridge: Cambridge university press, 2005. 45-51
11 Wang W, Liao Q, Zhang Q. COD: A cooperative cell outage detection architecture for self-organizing femtocell networks. IEEE Transactions on Wireless Communications, 2014, 13(11): 6007-6014