International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.335-352
http://dx.doi.org/10.14257/ijsia.2014.8.6.29

Construction and Reduction Methods of Vulnerability Index System in Power SCADA

Yuancheng Li and Shengnan Chu

School of Control and Computer Engineering,
North China Electric Power University, Beijing, China
ycli@ncepu.edu.cn, shengnanchu@163.com

Abstract

Electric power SCADA (Supervisory Control and Data Acquisition) system gradually transforming from a separate private network to an open public network, seriously increases the vulnerability risk in electric power SCADA. In order to assess the vulnerability risk in electric power SCADA system, the paper firstly uses Delphi method and AHP (Analytic Hierarchy Process) to build an index system of vulnerability risk assessment, to fully represent the vulnerability of electric power SCADA system. As index data of vulnerability risk assessment in power SCADA is characterized by strong relation and high dimensionality, the method of Autoencoder is proposed to reduce dimensionality of index data by representing high-dimensional data in a low dimensional space. Auto encoder method can obtain the optimal initial weight in pre-training and then back-propagate error derivatives adjusting weights with the initial weights to minimize the reconstruction error finally getting the best reconstructed results. The paper conducts simulation experiments about reconstruction error in pre-training and fine-tuning process in MATLAB experimental platform, and the experimental results show that dimensional code received by reducing dimensionality of data can basically fully represent high-dimensional data. The low-dimensional code as input can significantly reduce the complexity in the construction of model of vulnerability risk assessment in Electric power SCADA system in later work.

Keywords: *electric power SCADA system, index system of vulnerability assessment, Autoencoder, reducing dimensionality*

1. Introduction

In recent years, the events that ICS (Industrial Control Systems) is under attack , have occurred frequently, especially Iran's nuclear power plant attacked by Stuxnet virus in 2010, and SCADA systems attacked by Night Dragon leading confidential data stolen in 2011, which caused enormous economic losses, so security problems of industrial ICS have gradually attracted international awareness. SCADA system is one of important systems of ICS, and ICS of electric power is one of the critical infrastructures, so vulnerability risk assessment of power SCADA system is of great practical significance. SCADA system plays an important role in power system, including two main functions: monitoring and controlling. Power SCADA system continuously collects data from the remote terminal units, and returns control commands to ensure the continued normal operation of the terminal equipment to maintain a normal working environment for operation of power systems. With the ever-changing information technology, power SCADA systems necessarily require introduction of new IT (Information Technology) in order to meet the needs of continued normal operation of the power system. Power SCADA system will gradually transit from a closed, isolated private

ISSN: 1738-9976 IJSIA

network to an open public network, and communication protocols of open network and related equipment linked to the network make SCADA system become a target of cyber-attacks. As a result, research on security issues of SCADA system needs to get more attention.

Vulnerability in power SCADA system is likely to be an entry point for network attacks, so it is necessary to deeply analyze its vulnerability. Complex structures and function in power SCADA system lead to its vulnerability indicators associated with strong correlation and high dimensionality. However, there are few studies of reduction aspects of vulnerability risk indicators of the SCADA system. A lot of researches on dimension reduction of high-dimensional data have been done both at home and abroad. The method of rough set is proposed to solve the problem of attribute reduction in classification aspects and reviews about hybrid methods, rough set combined with the fuzzy sets, neural networks [1] and meta-heuristic algorithms, are given to solve the reduction problem [2]. For the problem of attribute reduction of the rough set ignoring edge information, an improved rough set is proposed [3]. Reduction set is obtained only depending on lowering values of dependence function. In [3], reduction set is determined by two conditions, respectively values of dependence function and distance matrix. And the experimental results show that the reduction set obtained by this method contains more useful information. In [4], the rough set theory is applied to the interlaced system to reduce the database and remove unnecessary information, and improves the accuracy of the operation. In [5], the method of fuzzy rough set is proposed to reduce attributes, and the data in UCI machine learning database is used to verify the effectiveness of the method, but the method has the disadvantage of large amount of calculation and time-consuming. In [6-7], local linear embedding algorithm is proposed to map high-dimensional data into a low-dimensional coordinate system and the method cannot introduce problems of local minima. But it is not easy to select the number of neighbors. In [8], the principal component analysis (PCA) for feature attribute reduction can effectively remove irrelevant features, but the method is not suitable for attribute reduction with abstract and nonlinear relationships as a linear dimension reduction method. In [9], the algorithm preventing generation of redundant implicants (RIs) is proposed to reduce the space complexity. The process of discernibility function (DF) of a dataset conversing to the disjunctive normal form (DNF) can cause the problem of RIs, which cause too larger number of minimal subsets and space complexity. The algorithm solves the problem but it may not be applied to some large-sized datasets. In [10], PTS (partial transmit sequence) combination with ABC (artificial bee colony) is proposed to solve high peak-average power ratio (PAPR) and successfully reduces computational complexity to find optimal phase factor in OFDM (orthogonal frequency division multiplexing). But the method to solve reduction of attributes is still to be researched. Due to the limitations to linear attribute reduction and the drawbacks of present method, Auto encoder is proposed to solve the problem of attributes reduction in the paper. The method of Autoencoder makes high-dimensional data represented in the low-dimensional space and can recover and reconstruct dimensions and geometry of the original data. It also can reflect the reconstruction as reconstruction errors [11].

The paper constructs the index system of vulnerability risk assessment in power SCADA system to more comprehensively assess vulnerability risk of the system. With the characteristics of vulnerability index in power SCADA system, the paper introduces the method of Autoencoder to reduce high-dimensional index to reduce the complexity of building model of vulnerability risk in later work. First, the method trains and adjusts the weights at many times in the pre-training process to obtain the optimal initial weights, and expands the each RBM (restrictions Boltzmann machine) network to get Autoencoder network. Then the best reconstructed data is obtained with the initial weights in the fine-tuning process through fine-tuning weights and minimizing the reconstruction error. Finally, reduced data is used instead of the original data for the latter part of the work [12]. Matlab simulation experiments on reconstruction error in pre-training process and fine-tuning process have verified the effectiveness of the method.

The main work of the remaining part of the paper is as follows: Section 2 analyzes the vulnerability in Electric power SCADA systems and builds index system of vulnerability risk assessment. Section 3 introduces Auto encoder method combined with work of the paper, and describes a process for indicators reduction of vulnerability risk in power SCADA system. Section 4 gives the experimental results and detailed analysis of reconstruction error in pre-training process and fine-tuning process. Section 5 summarizes work of the paper and discusses future work.

2. Establishment of Index System of Vulnerability Risk Assessment

2.1. Index of Vulnerability Risk Assessment

Power SCADA system consists of four subsystems, information collection and execution commands subsystem, information transmission subsystem, information collection, processing and control subsystem and the man-machine interface subsystem [13]. Its main functions include data collection and transmission functions, accidents recall function, historical data storage capabilities, reporting capabilities and special operations capabilities. According to the power structure and function of the SCADA system, power SCADA system occupies an important position as an important subsystem of power dispatching automation system. So it is necessary to assess vulnerability risk. The most important step is to build a risk assessment index system in vulnerability risk assessment process. The vulnerability indicators of power SCADA can directly affect the later work of vulnerability risk assessment. According to the basic principles, including feasibility, scientific nature, comprehensiveness, stability, comparability and hierarchy, this paper finalizes the risk assessment indicators in power SCADA systems based on norms of information security techniques - information security risk assessment combined with the Delphi method [14]. In order to build a more comprehensive and scientific index system, the paper selects 15 experts involved in scoring and judging based on the principles and conditions for the selected experts. First, the problem to consult needs to determine, namely index set of vulnerability risk in electric power SCADA system. Then the index set must be tentatively set according to the relevant literature and security specifications of related information systems, and the experts must also be determined in advance. Second, relevant information should be released to the experts, and the experts put forward their views on the initial set of indicators, combined with the information they collect. Finally, the views will be aggregated into a chart to compare and feedback to the experts. The experts' opinion is not modified after several revisions, which means indicators set are completed. The process of construction of indicators set is shown in Figure 1.

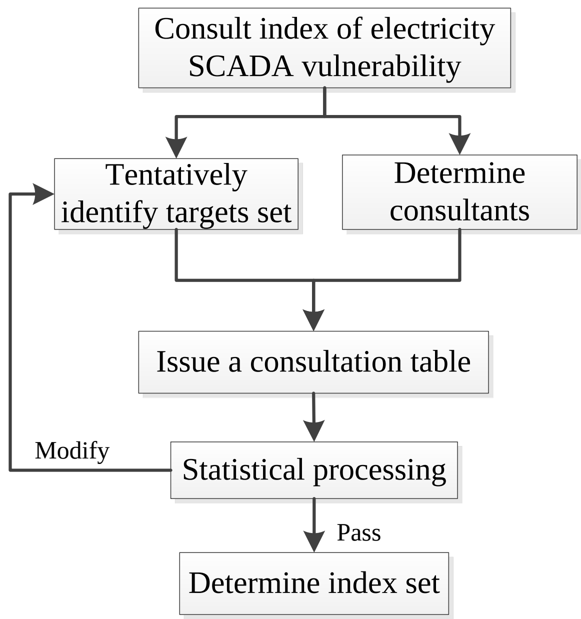

Figure 1. The Flowchart of Construction of Index Set using Delphi

2.2. Hierarchical Structure of Index System

Based on analysis above, the paper built index set of vulnerability risk assessment in power SCADA system with Delphi, and constructed hierarchical index system using Analytic Hierarchy Process (AHP) [15]. The topmost layer is also called as target layer, which is vulnerability in power SCADA system. Vulnerability of power SCADA system includes management vulnerability and technology vulnerability, called sub-target layer. Sub-target layer is also divided into level indicators. The level indicators of management vulnerability include system security construction management, personnel security management, security system, security management organization structure, system operation and maintenance management, security requirements and safety measures. The level indicators of technology vulnerability include physical environment, network architecture, and business operating environment, system software, application middleware, application system, host systems, network platform, and communication protocol. The secondary indicators or the bottom layer can be obtained by the level indicators. The paper includes a total of 85 indexes. The hierarchical structures of management vulnerability and technology vulnerability in power SCADA system are shown in Figure 2 and Figure 3 respectively.

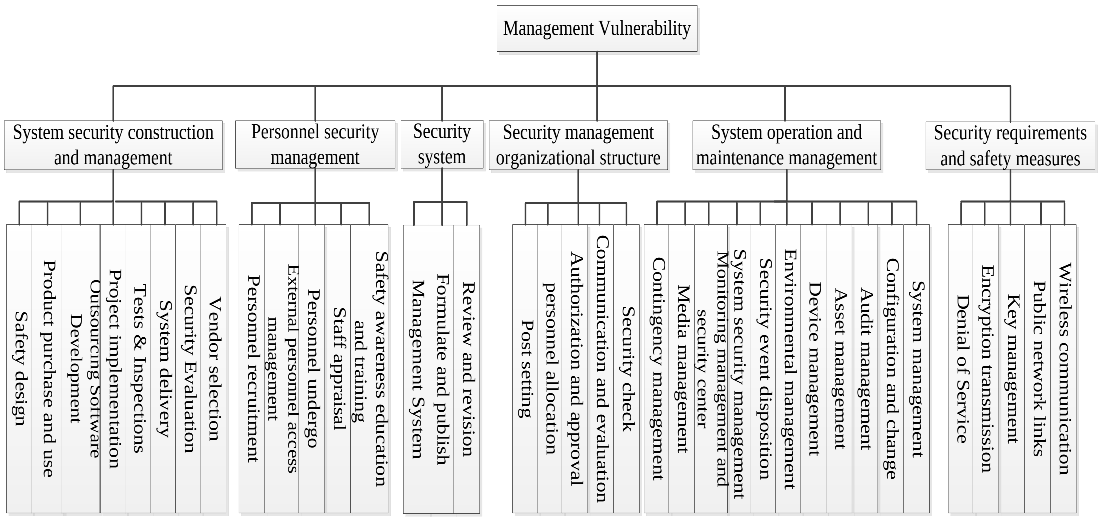

Figure 2. The Hierarchical Structure of Management Vulnerability of Power SCADA

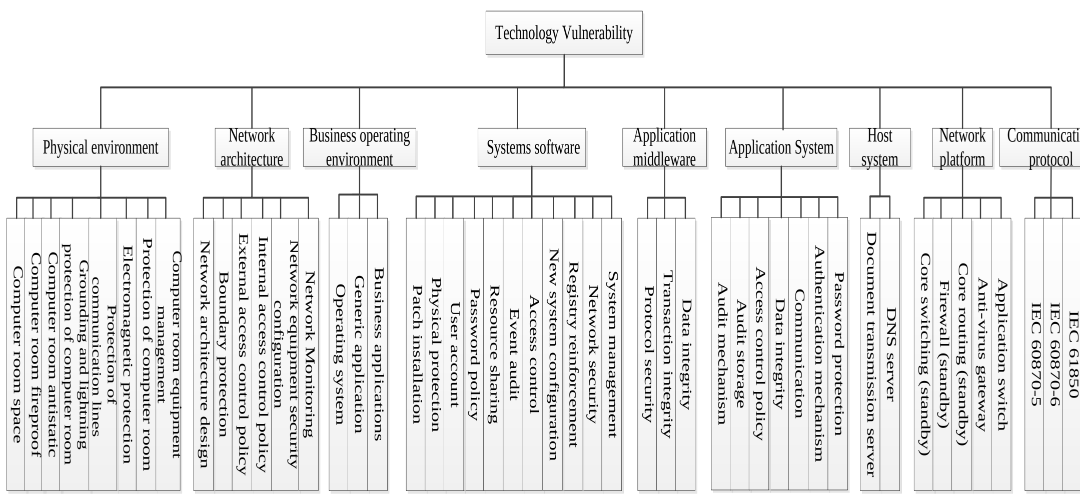

Figure 3. The Hierarchical Structure of Technology Vulnerability of Power SCADA

In Figure 2 and Figure 3, the management vulnerability includes 37 indexes and the technology vulnerability includes 48 indexes. In summary, vulnerability of power SCADA is a total of 85 indexes.

2.3. Construction of Assessment Samples

As data in power SCADA system is characterized by confidentiality, real sample data is difficult to obtain. The paper firstly designs questionnaires by access to large amounts of safety information related power SCADA system, and then seeks to nearly 15 experts in power SCADA system security scoring for each vulnerability indexes, obtaining partial of vulnerability risk ranks in power SCADA system. Power SCADA vulnerability risk is divided into five levels representing different levels of risk, which is shown in Table 1.

Table 1. Risk Level and Assignment in Power SCADA System Vulnerability

Risk level	5	4	3	2	1
Risk identification	Very high	High	Middle	Low	Very low
Risk assignment	0.8~1	0.6~0.8	0.4~0.6	0.2~0.4	0~0.2

From Table 1, we can see that vulnerability risk includes five risk levels and the higher extent of the risk represent the higher risk level. We can get the corresponding range of risk value according to the risk level. Values of vulnerability risk level for different indexes are obtained according to the expert scoring and norms of related information system. Target values of vulnerability risk are acquired by the method of fuzzy theory [16]. The paper first receives a fuzzy consistent judgment matrix by the experts scoring in comparing with degree of importance of indexes. Then rating value is obtained finally by multi-level fuzzy comprehensive evaluation, as the sample values. Although the sample values obtained in this paper which cannot be fully representative of the entire power SCADA system vulnerabilities, the method is feasible for verifying the effect of reduction and reconstruction process with Autoencoder.

3. Attribute Reduction based on the Network Structure of Autoencoder

3.1. Autoencoder

Autoencoder was introduced by G. E. Hinton in 2006 [11], as a kind of nonlinear reduction method developed based on multi-layer deep neural network. Autoencoder includes "encoder network" and "decoder network". "Encoder network" is characterized by adaptive and multi-layers network structure, which achieves the function that transforms high-dimensional data into low-dimensional code. "Decoder network" completes the function and obtains high-dimensional data with recovery from the low-dimensional code. Autoencoder can extract the low-dimensional code (feature) from high-dimensional data as a representative of high-dimensional data and can also establish a mapping between input data and output code. Deep Autoencoder often includes one visible layer and multiple hidden layers, and visible layers and hidden layers consist of multiple neurons, corresponding respectively called visible units and hidden units. The number of visible units is decided by the input, while the number of hidden units based on the human or experiences.

3.2. Network Structure Design of Autoencoder

For the 85 dimensional samples of the power SCADA systems, the experiment selects Autoencoder of three hidden layers, the number of which determines complexity of nonlinear structure of Autoencoder. Generally, the number of hidden layer is selected as 3-5 layers. The number hidden units gradually decrease and finally get desired reduction dimension which can be determined by experience. The paper designs the network structure of Autoencoder as 85-200-50-25-5. Autoencoder can be considered as a kind of nonlinear reduction method, and the initial weight is obtained by training weights and minimizing discrepancy between the original and reconstructed code in pre-training process. The required gradients can be obtained by back-propagating reconstruction error first through encoder network and then through decoder network to the optimal reconstructed results in fine-tuning process. The network structure in pre-training consists of four RBMs (restricted Boltzmann machines), the structure of which are 85-200, 200-100, 100-50, 50-5. The output of each hidden layers of RBM is the input of visible layers of next RBM respectively. The network structure of Autoencoder is 85-200-100-50-5 as encoder network and 5-50-100-200-85 as decoder network in fine-tuning process. The upper neurons can capture high correlation of the lower neurons and accurately describe the nonlinear relationship as the output by learning between layers and layers. Important features can be given greater weights and redundant features small by adjusting weights. High-dimensional data is nonlinearly mapped to a low-dimensional space, so that the final output can fully contain all of the information of high-dimensional data. The experimental designs the network structure of Autoencoder through which the 85 dimensional data is mapped to the 5 dimensional nonlinear spaces. It will significantly reduce the complexity of modeling of the risk assessment in the latter work.

3.3. Reduction Process of Autoencoder

Strictly speaking, realization process of Autoencoder mainly consists of two processes, pre-training process and fine-tuning process. The transition process from pre-training to fine-tune can be called unrolling process. The size of initial weight directly affects the effects of reconstruction in fine-tuning process. If the initial weight is too large, it is difficult to find a local minimum. With weights too small, the gradient of the front layers is too small, so that it has difficulty in training the Autoencoder network containing multiple hidden layers. Pre-training process can adjust weights to obtain a more appropriate initial weight, and unrolls to generate encoder network and decoder network which use the initial weights in fine-tuning process. Fine-tuning process uses the back propagation (BP) algorithm to fine-tune the weight to obtain a better reconstruction.

(1) Pre-training process and the unrolling process

Pre-training learns the stack, which consist of a number of restrictions Boltzmann machines (RBMs), to adjust network weights of each layer. RBM consists of two layer networks, called visible layer and the hidden layer, which is constituted by a plurality of neurons. The structure of RBM is shown in Figure 4.

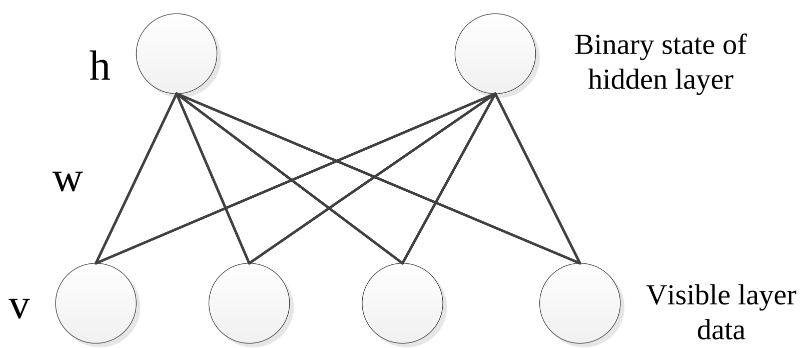

Figure 4. The Structure of RBM

The network can change the states of each layer according to the probability of each layer, while the probability is related to the energy function of every RBM. The probability is determined by the energy. The equations of energy and probability are shown as follows

$$E(v,h) = -\sum_{i\in data} b_i v_i - \sum_{j\in features} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{1}$$

$$p(v) = \sum_{h\in H} p(v,h) = \frac{\sum_h \exp(-E(v,h))}{\sum_{u,g} \exp(-E(u,g))} \tag{2}$$

where v_i and h_j are the binary states of original data i and feature j, b_i and b_j are biases, w_{ij} is the weight between them, H is the set of all possible binary hidden vectors. The probability and the energy can be changed by adjusting the weight to fit the problem you want to solve.

Weight updated formulas of visible layer and hidden layer is shown as follows:

$$\Delta w_{ij} = \varepsilon \left(\left\langle v_i h_j \right\rangle_{data} - \left\langle v_i h_j \right\rangle_{recon} \right) \tag{3}$$

where ε is a learning rate, $\left\langle v_i h_j \right\rangle_{data}$ is product of visible unit i and hidden unit j as the binary state the value of which is driven by the input data, and $\left\langle v_i h_j \right\rangle_{recon}$ is driven by the reconstructed data. The output of one RBM is obtained by learning and adjusting the weight, considered as the input of next RBM for learning weight. And the hidden units of the RBM serve as the visible units of next RBM for learning weights. Finally, the optimal initial weight is obtained by learning layer and layer until it returns the times of pre-training set in advance. The process of the weights learning of RBM is shown in Figure 5.

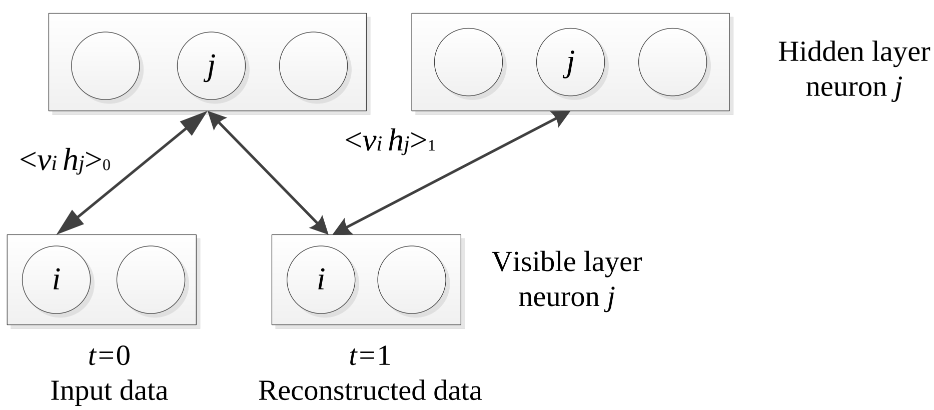

Figure 5. The Learning Process of RBM

Figure 5 shows the learning process of RBM, where first the data is inputted from the visible layer and the state of hidden unit j is updated according to visible unit i, and then the hidden unit j constructs the state of the visible unit i, and finally the visible unit i constructs the state of the hidden unit j. The process described above is complete process of training and learning of the RBM, in which the weight can be adjusted. After learning for many passes, the result of the final pass severs as the input of the next RBM and then the learning process is repeated as many times in the next RBM.

The hidden layer of each RBM combines with the visible layer of the next RBM as one layer. The Autoencoder can be obtained by the combination and unrolling the RBMs, where encoder network is symmetrical to decoder network. The pre-training and unrolling processes of Autoencoder are shown as Figure 6.

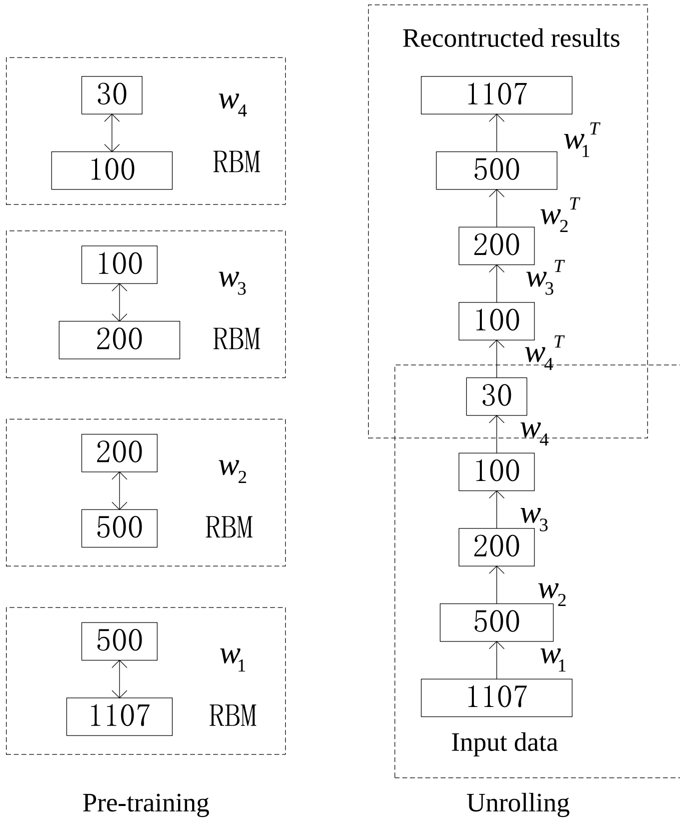

Figure 6. The Pre-training and Unrolling Processes of Autoencoder

From Figure 6, the pre-training process consists of four independent RBMs. The weight between the visible layer and the hidden layer is trained by the number iterations preset. The pre-training process finally gets the adjusted initial weight. The hidden layer of the RBM and the visible layer of the next RBM combine into one layer forming encoder and decoder networks, which use the weights from the pre-training process. And the process can be considered as the unrolling process.

(2) Fine-tuning process

The optimal reconstruction is obtained by minimizing reconstruction error in fine-tuning process using the initial weight from the pre-training process [17]. The input is transformed into a low dimensional code first through encoder network, and then the low dimensional code reconstructs the original date through decoder network. We defined the input dataset as $D = \{x_1, \ldots, x_n\}$, encoder function (encoder) as f_θ, decoder function (decoder) as g_θ. The set of parameters is $\theta = \{W, b, W', d\}$, where b and d are bias vectors of encoder and decoder, and W and W' are weight matrices of encoder and decoder.

We respectively defined the high dimensional input transformed into low dimensional code through encoder and constructed data recovered from low dimensional code through decoder according to the formula as follows:

$$h_n = f_\theta(x_n) \tag{4}$$

$$r = g_\theta(h) \tag{5}$$

Generally, fine-tuning process adjusts the weights by back-propagating reconstruction error first through the encoder network and then through the decoder network. The parameters have little impact on the reconstruction. The precise value of the initial weight from the pre-training process doesn't matter on the fine-tuning process while the region of weight is main factor. The reconstruction error function is selected by the domain range and character of the input. If the input is continuous real number, the reconstruction error function is shown as formula (6). However, if the input data is the binary, formula (7) is used as the reconstruction error function called as a binary cross-entropy loss.

$$L(x,r) = \left\| x - r \right\|^2 \tag{6}$$

$$L(x,r) = -\sum_{i=1}^{d_x} x_i \log(r_i) + (1-r_i)\log(1-r_i) \tag{7}$$

The method of Autoencoder fine-tunes the weight by conjugate gradient method to minimize the reconstruction error in the fine-tuning process. The optimal reconstruction can be obtained in the experiment by setting the epochs of fine-tuning in advance. The structure used in the fine-tuning process is shown as Figure 7.

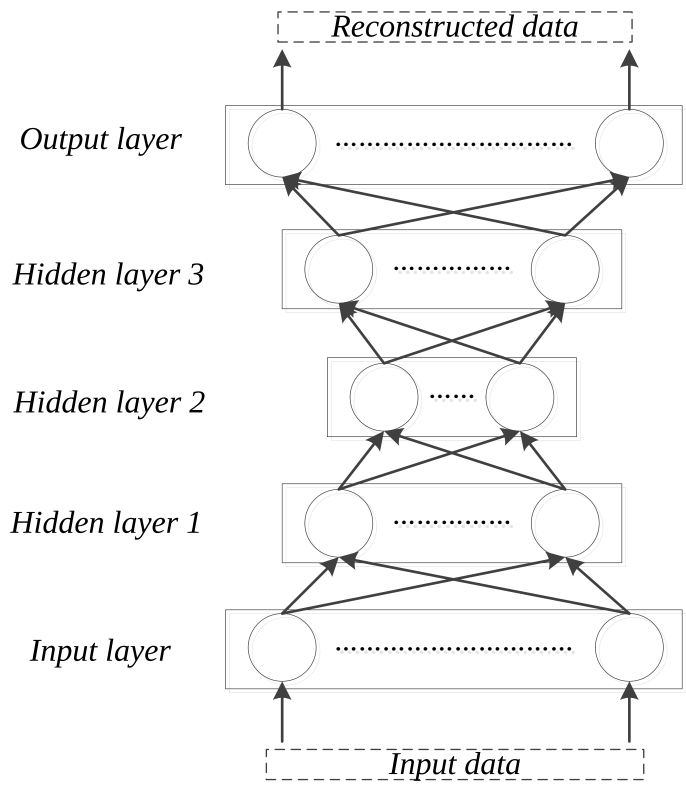

Figure 7. The Fine-tuning Process of Autoencoder

Can be seen from Figure 7, the Autoencoder consists of three hidden layers. The hidden layer, which is regarded as the output of the encoder and the input of the decoder, is called as the coder layer. The weight is fine-tuned through back-propagating reconstruction error. Finally, the reduction result is output.

3.4. Reduction of Vulnerability Indexes of Power SCADA

Autoencoder is proposed to reduce the vulnerability indexes of power SCADA. This work is implemented in MATLAB experimental environment. The concrete steps are shown as follows.

(1) Samples construction

Firstly, the risk level of indexes can be obtained through relative materials such as criterion of risk assessment in our country. However, the target value (the risk value of power SCADA) is difficult to get. This work calculates the target value by the method of fuzzy theory.

(2) Initialization

We initialize the parameters including the reduced dimension, maximum iterations, the number of hidden layers and visible layers, the number of neurons of hidden layers and visible layers and the initial weights randomly generated.

(3) Training of RBM

The parameters trained include the weight between the visible layer and the hidden layer, and the bias of each neuron. The weight is trained by transforming the input data from the visible layer to the hidden layer and reverse process as shown in Figure 5. The final weight can be obtained after maximum iterations.

(4) Formation of multilayers network

The hidden layer of the RBM and the visible layer of the next RBM are combined into one layer. Independent RBMs are connected into a multilayers network Autoencoder as shown in Figure 6.

(5) Back-propagation training

Adjust the weights from step (3) by back-propagating reconstruction error to minimize the cross entropy and get the optimal weight. The algorithm stops through the max iterations.

(6) Reduction

Construct reduction model with the optimal weight. Input test sample and output reduction result and reconstruction error.

The flowchart of reduction process is shown as Figure 8.

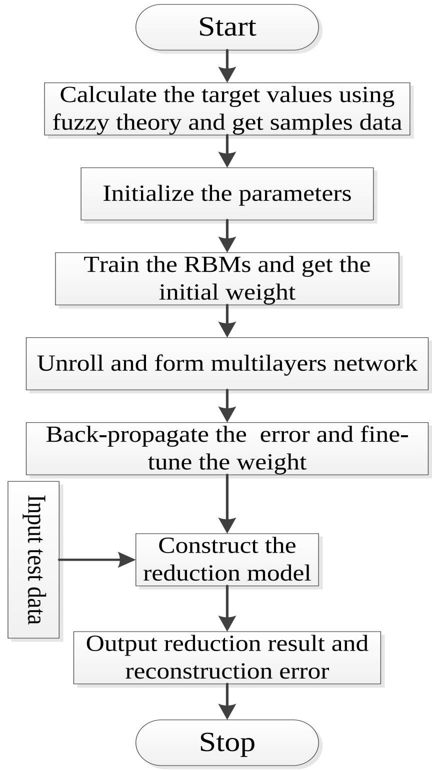

Figure 8. The Flowchart of Autoencoder

4. Experiment and Analysis

4.1. Evaluation Criterion

The experiment selects MSE (Mean Squared Error) as the evaluation criterion of measuring the reconstruction error of pre-training process and fine-tuning process to verify the effect of training the initial weight in pre-training process and the reduction effect in fine-tuning process. MSE is defined as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_{data} - y_{recon})^2 \tag{8}$$

where N is the size of training sample or test samples, y_{data} is the training data or the test data, and y_{recon} is reconstructed data through different RBM in pre-training process or reconstructed data through Autoencder in fine-tuning process. However, as the variables involved in the experiment are almost expressed in matrix form, y_{data} and y_{recon} needs complex calculation and processing and they are transformed into the form applicable to formular (8). As result, the code of implementation process for calculating the reconstruction error in experiment is very complicated and can be detailedly learned in the experiment of Hinton's paper [11,18].

4.2 Experiment Result

The experiment randomly selects 150 samples from 220 samples as training samples, the remaining 70 samples as test samples. 85-demensional data can be mapped into 5-

demensional spaces through Autoencoder, of which the network structure is 85-200-100-50-5, which would greatly reduce the complexity of modeling of risk assessment in the latter work. We set the learning rate as 0.1, 10 epochs of pre-training and 10 epochs of fine-tuning at the experiment. The epochs of pre-training mean the epochs of training for each RBM. Simulated data of the experiment is bounded between 0 and 1. We respectively do experiments on reconstruction error of pre-training and fine-tuning to verify Autoencoder method for reducing the dimensionality of data of vulnerability risk assessment in power SCADA system.

The pre-training process consists of 4 layers of RBM, which includes two layer of networks respectively called as the visible layer and the hidden layer. The visible layer of the first layer of RBM includes the number of neuron units equal to the number of the input samples, which are 85. The number of hidden units is determined by the dimensionality reduced. According to the experience and the dimensionality of reduction required, we set the number of hidden layer of the first layer of RBM as 200. The number of visible units and hidden units of the first layer of RBM are 85, 200, which can be denoted by 85-200. In general, the learning results of the upper hidden layer serve as the input of the lower visible layer, which means the number of the hidden units in the upper layer is equal to the number of visible units in the lower layer. What's more, the number of layer-by-layer neurons is decreased by half. We design the structure of the remaining RBM, which are denoted by RBM 200-100, RBM 100-50, RBM 50-5, in order to obtain the desired structure of the experiment. Except the hidden units of the top RBM, each hidden unit has two states 0 and 1. The visible units have the real-valued activities, which is bounded between 0 and 1. The hidden units of the top RBM also has real-valued states, which is drawn from a unit Gaussian function. The mean of the unit Gaussian function is determined by the input of visible units. The hidden unit j is set to 1 with the probability $\sigma(b_j + \sum_j v_i w_{ij})$, where $\sigma(x) = 1/\left[1 + \exp(-x)\right]$ is the Sigmoid logistic function, b_j is a bias of hidden unit j , and v_i , w_{ij} are respectively the state value and the weight of hidden unit j combined with visible unit i .According to the hidden unit, the state of visible unit i also can be set to 1, with the probability $\sigma(b_i + \sum_j h_j w_{ij})$, where v_i , w_{ij} are respectively the state value and the weight of visible unit i combined with hidden unit j . The hidden units are updated for many times as described above to adjust the weight according to formula (3). So the optimal initial weight can be obtained for fine-tuning. The experiment sets 10 epochs for pre-training to get the reconstruction error of different RBM. The result is shown in Figure 9.

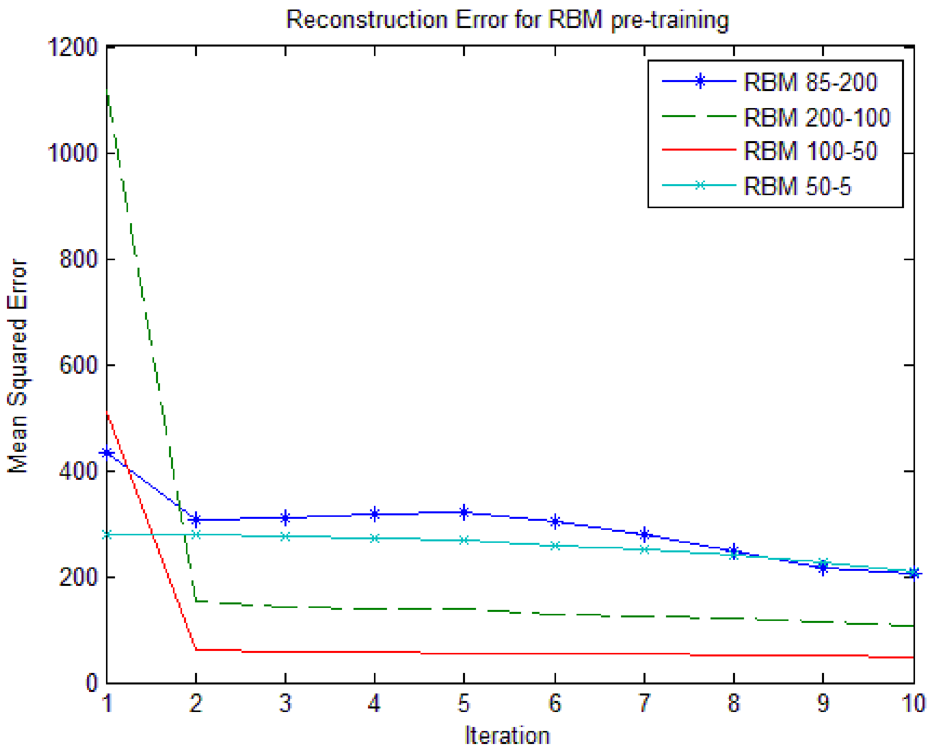

Figure 9. The Reconstruction Error of RBM for Pre-training

Figure 9 shows the changing process of reconstruction error of the four layer of RBM in different epochs for pre-training. As the initial weight for pre-training is generated randomly, the pre-training process aims to obtain the new optimal initial weight for fine-tuning by updating the weight repeatedly. From Figure 9, we can see that the construction error is larger when the epochs of pre-training is 0, which means the initial weight generated randomly is worse for pre-training. When pre-training for 2 times, the reconstruction error is substantially in equilibrium and the optimal weight for fine-tuning can be obtained for pre-training for 2 times or more than 2 times. Figure 9 shows that the range of the weight adjustment is different. The reason is that the weight required in different RBM layers is different for distinct number of nerve units. The process of RBM is shown in Figure 5. However, there are downward trends for four curves. The pre-training process is responsible for searching for suitable weights, which will be used in the fine-tuning. The process of weight adjustment can be seen from the downtrend of the curves of reconstruction error for pre-training in Figure 9. So we can set the epochs as 2 or more than 2 for pre-training to obtain the new initial weight for fine-tuning.

After pre-training, the RBM can be unrolled to be a multilayer neural network - Autoencoder, in which the hidden layer of the upper RBM combines with the lower RBM as one layer. The encoder network and decoder network of Autoencoder are exactly symmetrical. The fine-tuning process minimizes the reconstruction error function, for example binary cross-entropy loss function, and back-propagates the error derivatives to fine-tune the weight. The process of fine-tuning is shown as Figure 7. We define the set input data $\{x_0, x_1, \ldots, x_n\}$, where x_0 is the bias set to 1. The total weighted sum of the hidden units is $net_j = \sum_{i=0}^{n} w_{ij} x_i$, where w_{ij} is the weight between unit x_j and x_i, and the output accordingly is $z_j = \sigma(net_j)$. The last hidden layer serves as the input of the output layer. The value of unit k of the output layer is $net_k = \sum_{j=0}^{n} w_{jk} x_j$.The weight is updated through the visible layer and hidden layer and finally the reconstruction error achieves minimum. The weights of the visible layer and hidden layer are updated according to $\Delta w_{jk} = \eta(x_k - \hat{x}_k) f'(net_k) z_j$ and $\Delta w_{ij} = \eta \left(\sum_{k=0}^{n} (x_k - \hat{x}_k) f'(net_k) w_{jk} \right) f'(net_j) x_i$ [19]. We can obtain the optimal reconstruction by the fine-tuning described above. In order to demonstrate the proposed method applies to the index reduction of vulnerability risk in power SCADA, we do experiments on the comparison between the reconstruction of the training samples and the reconstruction of the testing samples in fine-tuning process. We set the epochs of fine-tuning as 10 and the experimental result is shown as Figure 10.

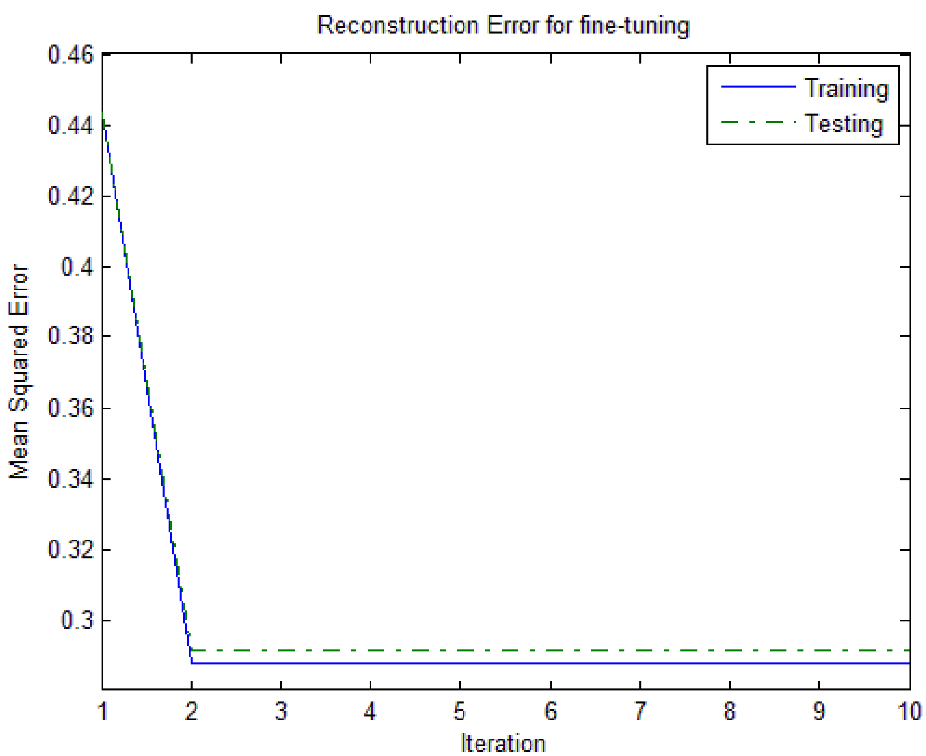

Figure 10. The Comparison of Reconstruction Result for Fine-tuning

Figure 10 shows a comparison of the reconstruction error of training samples and testing samples with 10 epochs for fine-tuning. The final weight from the pre-training process servers as the initial weight of fine-tuning, and the constructed data is obtained through 10 times for fine-tuning. The reconstruction error represents the discrepancy between the original data and its reconstruction. Can be seen from Figure 10, the level of discrepancy is reduced to very low through 2 times of iterations. When the epochs for fine-tuning reach 2, the construction error almost keeps constant and the construction achieves optimal. The minimal reconstruction error of the training samples is 0.287 and the minimal reconstruction error of the testing samples is 0.291 according to the experimental result in MATLAB. The discrepancy between the result of training samples and testing samples is very small, so reduced data contains fully information, which can represent the original data. The epochs of fine-tuning can be set to 2 or more than 2, so that the optimal reconstruction can be obtained. The paper uses the method of Autoencoder to make 85 dimensional data mapped into 5 dimensional spaces. 150 training samples and 70 testing samples are reduced to be 5 dimensional data, from which we select one sample from training and testing reduced samples. The reduced sample is shown as Table 2.

Table 2. 5 Features from Reduced Samples of Autoencoder

Samples	Feature1	Feature2	Feature3	Feature4	Feature5
Training	-2.104e-001	-9.971e-001	-8.532e-001	4.473e-001	-8.383e-002
Testing	-2.100e-001	-9.970e-001	-8.530e-001	4.474e-001	-8.361e-002

Table 2 shows the reduction result of the original data from one training sample and one test sample, which are randomly selected from the training sample and test sample. The 5-dimensional features can represent the 85-dimensional original data and contain the total information nearly. From the analysis above, the method avoid the problems of the influence of subjective and the difficulty to decide the minimal subset. The results also demonstrate the effectiveness and applicability of the method to solve the problem of reduction of vulnerability indexes in power SCADA.

5. Conclusion

Attribute reduction problem is not a new issue, but the Autoencoder is applied to solve vulnerability index system in power SCADA is relatively less. In this paper, we propose the method of Autoencoder for indexes reduction of vulnerability risk assessment in power SCADA system. Autoencoder can automatically adjust the weights and the important indicators are given to a larger weight while the redundancy indexes are given a smaller weight to reduce the objectivity of weight assigned by experts. The experiment evaluates the reduction result through the reconstruction error. The reconstruction error is very low and discrepancy between the training samples and test sample is 0.004. The proposed method achieves good results in the reduction process of vulnerability index system in power SCADA. And the paper verifies the validity of the method by experimental analysis.

Acknowledgements

This paper is supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry ([2013]693).

References

[1] N. S. Jaddi and S. Abdullah, "Hybrid of genetic algorithm and great deluge for rough set attribute reduction", Turkish Journal of Electrical Engineering & Computer Sciences, vol. 21, no. 21, **(2013),** pp. 1237–1750.

[2] K. Thangavel and A. Pethalakshmi, "Dimensionality reduction based on rough set theory: A review", Applied Soft Computing, vol. 9, **(2009),** pp. 1-12.

[3] N. Parthalain, Q. Shen and R. Jensen, "A distance measure approach to exploring the rough set boundary region for attribute reduction", IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 3, **(2010),** pp. 305-317.

[4] G. Jeon, D. Kim and J. Jeong, "Rough sets attributes reduction based expert system in interlaced video sequences", IEEE Transactions on Consumer Electronics, vol. 52, no. 4, **(2006),** pp. 1348-1355.
[5] E. C. C. Tsang, D. Chen, D. S. Yeung, X. Z. Wang and J. W. T. Lee, "Attributes reduction using fuzzy rough sets, IEEE Transactions on Fuzzy Systems", vol. 16, no. 5, **(2008),** pp. 1130-1141.
[6] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding", Science, vol. 290, **(2000),** pp. 2323-2326.
[7] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds", The Journal of Machine Learning Research, vol. 4, **(2003),** pp. 119-155.
[8] Y. Bengio, A. Courville and P. Vincent, "Representation learning: a review and new perspectives", Pattern Analysis and Machine Intelligence, vol. 35, no. 8, **(2013),** pp. 1798-1828.
[9] M. Hacibeyoglu, F. Basciftci and S. Kahramanli, "A logic method for efficient reduction of the space complexity of the attribute reduction problem", Turkish Journal of Electrical Engineering & Computer Sciences, vol. 21, no. 19, **(2011),** pp. 643-656.
[10] N. Taspinar, D. Karaboga, M. Yildirim and B. Akay, "PAPR reduction using artificial bee colony algorithm in OFDM systems", Turkish Journal of Electrical Engineering & Computer Sciences, vol. 21, no. 19, **(2011),** pp. 47-58.
[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", Science, vol. 313, **(2006),** pp. 504-507.
[12] P. Baldi, "Autoencoders, unsupervised learning, and deep Architectures", Journal of Machine Learning Research-Proceedings Track, vol. 27, **(2012),** pp. 37-50.
[13] J. McDonald, "Electric Power Substations Engineering", 3rd ed, Boca Raton, FL, USA: CRC Press, **(2012)**.
[14] D. V. Jeste, M. Ardelt, D. Blazer, H. C. Kraemer, G. Vaillant and T. W. Meeks, "Expert consensus on characteristics of wisdom: A Delphi method study", Gerontologist, vol. 50, **(2010),** pp. 668-680.
[15] V. Podvezko, "Application of AHP technique", Journal of Business Economics and Management, vol. 10, **(2009),** pp. 181-189.
[16] J. W. Lin, C. W. Chen and C. Y. Peng, "Potential hazard analysis and risk assessment of debris flow by fuzzy modeling", Natural hazards, vol. 64, no. 1, **(2012),** pp. 273-282.
[17] V. Miranda, J. Krstulovic, H. Keko, C. Moreira and J. Pereira, "Reconstructing Missing Data in State Estimation With Autoencoders", IEEE Transactions on Power Systems, vol. 27, **(2012),** pp. 604-611.
[18] G. E. Hinton and R. R. Salakhutdinov, "Supporting online material for reducing the dimensionality of data with neural networks", Science, vol. 313, **(2006),** pp. 504-507.
[19] C. C. Tan and C. Eswaran, "Performance comparison of three types of autoencoder neural networks", In: IEEE 2008 Second Asia International Conference on Modeling & Simulation, **(2008)** May 13-15, pp.213-218, Kuala Lumpur, Malaysia.

Authors

Yuancheng Li, he received the Ph.D. degree from University of Science and Technology of China, Hefei, China, in 2003. From 2004 to 2005, he was a postdoctoral research fellow in the Digital Media Lab, Beihang University, Beijing, China. Since 2005, he has been with the North China Electric Power University, where he is a professor and the Dean of the Institute of Smart Grid and Information Security. From 2009 to 2010, he was a postdoctoral research fellow in the Cyber Security Lab, Pennsylvania State University, Pennsylvania, USA. His current research interests include Smart Grid operation and control, information security in Smart Grid.

Shengnan Chu, she is now a Master of School of Control and Computer Engineering, North China Electric Power University since 2012. Her current research topic is Vulnerability analysis and network security risk assessment of electric power SCADA system.