# The method of communication system fault diagnosis based on deep belief net

Juan Li[1], Bin Chen[2]
[1]Department of Information Engineering, Wuchang Institute of Technology,
Wuhan, 430065
China
[2]Electronic Engineering College, Naval University of Engineering,
Wuhan, 430032
China

Received: February 26, 2021. Revised: July 12, 2021. Accepted: August 10, 2021. Published: August 12, 2021.

**Abstract—To meet the need of fault diagnosis for military communication system, an effective method based on deep belief (DBN) net is proposed. During the fault diagnosis, the bottom layer of DBN model is used to receive the input fault signals to extract the fault features and the fault classification results will be outputted after softmax classified. Accordingly, algorithms for DBN model and training and RBM parameter learning have been designed. To reduce the running time, parallel solutions based on MapReduce framework have been provided. In order to test and verify the effect of DBN fault diagnosis, the communication experiment system is built in the laboratory which the output signals of the transmitter and the receiver are measured and collected as the original data for further learning and training. Compared with the traditional fault diagnosis methods, it can be found that DBN method has high accuracy in fault diagnosis and the process is simple and friendly. It is impossible to realize real-time diagnosis and online diagnosis for the communication system. The research can be applicated to the health management of communication equipment, and it will provide advanced technical support and software program for the health of communication equipment.**

**Keywords—Deep Belief Net, Deep Learning, Fault Diagnosis, Military Communication System**

## I. INTRODUCTION

The communication system is becoming more and more complex. It has complex system characteristics in structure, composition, and function, which leads to the complexity of system fault [1]-[3]. Fault diagnosis is one of the core technologies to ensure the healthy operation of communication system. It should determine fault location and cause quickly and accurately, and then estimate the severity and trend. Scholars have carried out extensive research and put forward many feasible methods, which greatly promoted the development and maturity of fault diagnosis technology [4]- [9]. Deep learning is one of the latest trends in machine learning and artificial intelligence. There are two core points in deep learning. Artificial neural network with multiple hidden layers has excellent learning ability. The characteristics obtained from learning have more essential description of data, which is conducive to classification. The difficulty of deep neural network in training can be overcome through layer-wise pretraining. Deep learning emphasizes the depth of the model structure, highlights the importance of characteristic learning. Using big data to learn characteristics can describe the rich internal information, so as to make classification or prediction easier [10]-[13]. The complexity of communication system fault increases the difficulty of fault diagnosis, and the classical fault diagnosis methods are difficult to meet the needs. The multi hidden layer machine learning model in deep learning can learn the fault characteristics of communication system with massive training data to achieve accurate fault diagnosis. After decades of development, deep learning structure and algorithm are increasingly rich and perfect. As the one of main models of deep learning, deep belief net (DBN), is more suitable for fault diagnosis of communication system.

## II. DIAGNOSTIC PRINCIPLES

DBN consists of several layers of unsupervised restricted Boltz-mann machine (RBM) and one layer of backpropagation (BP) classifier. DBN training can be divided into two steps. The first step is the pre-training stage, which uses the greedy layer-wise training method to train each layer of RBM separately and unsupervised. The second step is the fine-tuning stage, which uses the supervised training entity relationship classifier of BP network to spread the error information back to each layer of RBM and fine tune the whole DBN network.

*A. Feature Extraction*

During the fault diagnosis of communication system, the bottom layer of DBN model is used to receive the input fault signals (unlabeled data), and then extract the fault features through multiple stacked RBM layers to lay a foundation for fault classification [14]-[17]. RBM is the basic module of DBN model which determines the capability of DBN model.

RBM is an energy based generation model, which can provide a learning method for data with unknown distribution by learn the internal attributes of data [18]-[20]. The parameter which needs to be learned in sample training determines the performance of RBM. The probability distribution represented by RBM should be as consistent as possible with the given training sample distribution. The parameter can be learned from the training samples through maximum likelihood estimation and stochastic gradient descent to maximize the $P(v)$ (1) and (2).

$$L_s(\theta)=\prod P(v), \log L_s(\theta)=\sum \log P(v) \tag{1}$$

$$\frac{\partial \log L_s(\theta)}{\partial \theta}=-\sum_h p(h\mid v)\frac{\partial E(v,h)}{\partial \theta}+\sum_v\sum_h p(v,h)\frac{\partial E(v,h)}{\partial \theta} \tag{2}$$

Because of the complexity of calculation, p(h|v) and p(v,h) should be obtained from training samples through alternating Gibbs sampling (AGS) instead of direct calculating.

*B. Fault Classification*

The basic principle of DBN fault classification is shown in Fig. 1. Basing on the feature extraction and dimension-reduction achieved by multilayer RBM, dimension-reduced data of fault feature is used as the input of softmax classifier. Finally, the fault classification results will be outputted after softmax classified.

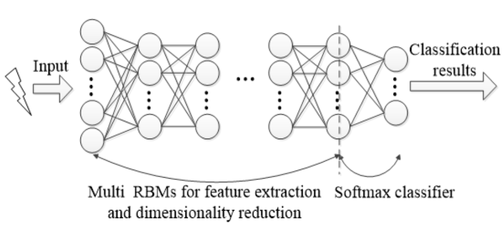


Fig. 1 basic principle of DBN fault classification

In DBN, the layering RBM is trained from the bottom to the top layer by layer. The hidden layer neurons of the top RBM are output as the high-level feature set of fault signals. Accordingly, softmax classifier will classify the faults through the BP neural network setting in the last layer of DBN.

BP network is a supervised classifier according to the basic ideas of signal forward propagation and error back propagation. The training data is transmitted forward to the output layer. If the actual output is not consistent with the expected output, error back propagation is carried out. The error back propagation stage is start from the error generated in the output layer, and then back propagation is done from the upper to the lower. By the gradient descent method, the parameters are corrected to minimize the final network error.

In the pre-training phase of DBN, the highest RBM hidden layer is used as the initialization of BP network weight parameters to overcome the limitation of long training time and local optimum of BP network. In the fine-tuning stage, the back propagation can modify the RBM parameters of each layer, so that the multilayer RBM feature mapping can be optimized as much as possible. After DBN training, a fault classification model is generated, which can diagnose the fault according to the fault signal.

## III. DIAGNOSTIC ALGORITHMS

*A. Related Definition*

Basing on DBN, the fault diagnosis of communication system can be described by (3). The purpose of fault diagnosis is to confirm the corresponding relationship between signal and fault, which can be obtained through diagnosis of DBN model. Among them, signal can be regarded as an n-tuple, and fault value range is all fault types or fault types concerned by users.

$$\begin{aligned}&DBNDiagnose=\{(Signal,Fault)\mid Signal\xrightarrow{DBN}Fault\}\\&Signal=(s_1,s_2,...s_n), Fault\in Faults=\{F_1,F_2,...,F_m\}\\&TrainData=\{(Signal,Fault)\mid Signal\rightarrow Fault\}\\&TestData=\{(Signal,Fault)\mid Signal\rightarrow Fault\}\end{aligned} \tag{3}$$

The set of train data named TrainData is used for DBN model training, and the set of test data named TestData is used for DBN model testing. The diagnostic performance of DBN model can be analyzed through the mathematical statistics of test results. The main performances include accuracy, false positive and false negative. For a fault type, the calculation formulas of DBN diagnostic performances are as follows.

$$DBNDiagnoseAbility=(Accuracy,Positive,Negative) \tag{4}$$

$$Accuracy(Fault)=\frac{count(\{(Signal,Fault).in.DBNDiagnose.and.TestData)}{count((Signal,Fault).in.TestData)} \tag{5}$$

$$Positive(Fault)=\frac{count(\{(Signal,Fault).in.DBNDiagnose.but.not.in.TestData)}{count((Signal,Fault).in.TestData)} \tag{6}$$

$$Negative(Fault)=\frac{count(\{(Signal,Fault).not.in.DBNDiagnose.but.in.TestData)}{count((Signal,Fault).in.TestData)} \tag{7}$$

*B. DBN Model and Training*

Based on the above definition and description, the specific algorithm of communication system fault diagnosis and test is as follows.

*Algorithm Communication DBN Diagnose (TrainData, TestData. level, DBN Diagnose Ability)*

*//Build and test DBN fault diagnosis model based on training data set and test data set of communication system*

*//input: TrainData, TestData, level of DBN*

*//output: DBN Diagnose Ability*

*Normalize TrainData and TestData*

*Initialize DBN with*

*dimension of signal in TrainData / Testdata for DBN visible level*

*amount of fault in TrainData / Testdata for DBN output*

*level for DBN levels*

*DBNTrain (TrainData)*

*for each signal TestData*

*find Fault according to DBN Diagnose*

*for each Fault in TrainData*

*calculate Accuracy, Positiv, Negative*

The core of algorithm Communication DBN Diagnose is DBN Train, the training algorithm of DBN model. According to the principle of DBN fault diagnosis, the algorithm is designed as follows.

*Algorithm DBN Train (TrainData, DBN)*

*//DBN training according to the training data set to determine the DBN model*
*//input: Initialized DBN, TrainData*
*//output: trained DBN*
*Initialize visible with signal in TrainData*
*for each RBM in DBN from bottom to level-2*
*Initialize hide with random*
*RBM Learn (visible, hide)*
*visible = hide*
*for each target data in TrainData*
*train top level with softmax classifier*
*for each layer in DBN*
*feedforward and back-propagation to get the gradients of the weight*
*fine-tuning* ($RBM_{layer}$)

### C. RBM Parameter Learning

RBM training is involved in both two steps of DBN training, bottom-up unsupervised learning and top-down supervised learning. In a sense, RBM determines the performance of the whole DBN model. So, the learning of RBM parameter   is the key of the whole DBN model training.

As mentioned in DBN fault diagnosis principle, RBM parameter learning needs to use Gibbs sampling method to extract appropriate sample data from training samples for parameter calculation. However, it is difficult to ensure the convergence speed of Gibbs sampling. By defining the distance difference of probability distribution, the algorithm of contrast divergence (CD) can improve the algorithm of Gibbs sampling to quickly improve the calculation speed on the premise of ensuring the accuracy. The CD algorithm takes the initial value of the visible layer from the training sample set, and then carries out k-steps Gibbs sampling with alternately sampling visible layer and hidden layer each time. The specific algorithm of RBM Learn using the CD algorithm is as follows.

*Algorithm CD_k_RBMLearn (k, s, n, m, θ, η)*
*//RBM Training by* the CD algorithm
*//input: sampled-data k, sample signal s, Number of neurons in the visible layer n, Number of neurons in the hidden layer m*
*//RBM parameter θ = (b, c, w), learning rate η*
*//output: learned RBM parameter θ*
*for each hidden neuron i*
*sample* $h_{0i}$ *according to* $P(h_{0i}=1|v_0)$
*for each visible neuron j*
*sample* $v_{1j}$ *according to* $P(v_{1j}=1|h_0)$
*calculate dθ with*
$dw = P(h_0 = 1|v_0)\, v_0^T - P(h_1 = 1|v_1)\, v_1^T$
$db = v_0 - v_1$
$dc = P(h_0 = 1|v_0) - P(h_1 = 1|v_1)$
*upgrade θ with*
$w = w + \eta dw,\ b = b + \eta db,\ c = c + \eta dc$

## IV. PARALLEL ALGORITHMS

DBN is a fault diagnosis model with large amount of data and complex calculation. The traditional serial calculation is time-consuming and not practical. Hadoop, a cloud computing platform with MapReduce framework [21]-[24] can provide parallel solution to ensuring the feasibility and practical value of DBN fault diagnosis.

### A. RBM Training Paralleling

The basic idea of RBM training paralleling is shown in Fig. 2. It can be divided into the following steps.

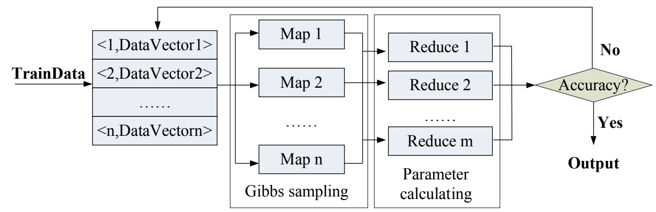


Fig. 2 Mapreduce of RBM training

The 1st Step is for normalization. After normalizing the samples, the training data would be divided into several subsets, with formatting <ID, DataVector>. ID represents the sample number, and DataVector represents the training sample data in vector form. These training data will be distributed on multiple slave servers, and MapReduce can realize data slicing automatically.

The 2nd Step is for map. Gibbs sampling is completed according to the CD algorithm trained for RBM. The input of map is the initial data or iterative data of RBM. After sampled by Gibbs, it will output the change of RBM parameters including the increment of weight value dW, the increment of unit offset in the visible layer db, and the increment of unit offset in the hidden layer dc.

The 3rd Step is for reducing. RBM parameters, W, b and c, would be updated according to dW, db and dc which output in the step of map.

There are two ways to deal with training result. If the output of the 3rd step meets the accuracy requirements, the training results can be output, and the RBM training is finished. Otherwise the result of the reduce step would be output to HDFS for the next round of MapReduce.

For a given training set TrainData, parallel algorithm for RBM training to determine the parameter θ (b, c, w) with MapReduce framework is as follows.

*Algorithm RBM_Mapreduce (TrainData / ReduceData, θ)*
*//MapReduce of RBM Training*
*//input: original dataset TrainData or iterative dataset ReduceData*
*//output: learned RBM parameter θ*
*initialize input data to many epoches*
*for each epoch*
***Map***
*input <ID, DataVector>*
*do Gibbs sanpling*
*calculate approximate gradients for θ (W, b, c)*
*output <θ, θlist>*
***Reduce***
*input <θ, θlist>*
*summary the gradients for θ (W, b, c)*
*calculate θ (W, b, c)*
*output <ID, DataVector>*

*calculate the learned θ (W, b, c)*

### B. BP Fine Tuning Paralleling

The basic idea of BP fine tuning parallelization is shown in Fig. 3. It can be divided into the following steps.

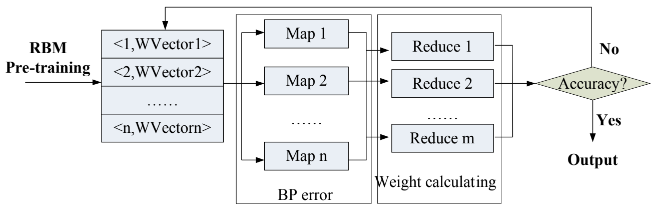


Fig. 3 Mapreduce of BP finetune

The 1st step is for data normalizing. Loading the weights (Wn, Wn-1, …, W1) learned in RBM pre-training, and initializing the weight Wn+1, the input data would be normalized to the format of <ID, wvector>. ID represents the sequence number of data subset. WVector = (Wn + 1, Wn, Wn-1, …, W1).

The 2nd Step is for map. According to the error back propagation algorithm, the weight gradient increment dW = (dWn + 1, dWn, dWn-1, …, dW1) among all layers of DBN is calculated, and then update (Wn+1, Wn, Wn-1, …, W1). The input of map is <'dWi', (dWi, Wi)>, i [1, 2, …, n+1].

The 3rd Step is for reducing. Analyzing the map result data to obtain all the weight and increment information correspondingly, Wi would be calculated.

There are two ways to deal with training result. If the output of the 3rd step meets the accuracy requirements, the training results can be output, and the BP fine tuning is finished. Otherwise the result of the reduce step would be output to HDFS for the next round of MapReduce.

According to the result of RBM pre-training (Wn+1, Wn, Wn-1, …, W1), parallel algorithm for BP fine tuning to determine optimal weight among each layer of DBN with MapReduce framework is as follows.

*Algorithm BPFinetune_Mapreduce (TrainedRBMs, θ)*
*//BP algorithm to fine tune every layer of RBM*
*//input: Pre trained RBM weight of each level or iterative RBM weight of each level*
*//output: Fine-tuned RBM weight of each level*
*load or initialize the learned RBMs W ($W_1$, $W_2$, …, $W_H$) and $W_{H+1}$ for top level to many epoches*
*for each epoch*
***Map***
*Input <ID, WVector>*
*feedforword and back propagation*
*calculate the gradients for every $W_i$*
*update W ($W_1$, $W_2$, …, $W_H$, $W_{H+1}$)*
*output <'d$W_i$', (d$W_i$, $W_i$)>*
***Reduce***
*input <'d$W_i$', (d$W_i$, $W_i$)>*
*classificae and summary according to i of $W_i$*
*calculate and update $W_i$*
*output <ID, WVector>*
*calculate the fine-tuned W ($W_1$, $W_2$, …, $W_H$, $W_{H+1}$)*

### C. Main Function of DBN

After RBM pre-training parallelized by RBM_Mapreduce and BP fine tuning parallelized by BPFinetune_Mapreduce, DBN can be trained accordingly. As DBN_Main used to represent the parallel process of DBN training, and N used to represent the number of layers of RBM in DBN, the specific algorithm is as follows.

*Algorithm DBN_Main (TrainData, DBN)*
*//parallel algorithm to complete DBN fault diagnosis training*
*//input: train set TrainData, initialized DBN*
*//output: trained DBN*
*setup for 1st level RBM*
*initialize the visible level according to TrainData*
*initialize necessary parameters for RBM*
*while precision is not satisfied*
*invoke RBM_Mapreduce ()*
*save the learned $RBM_1$*
*for each other RBM*
*setup for ith level RBM*
*initialize the visible level according to the hide level of the i-1 RBM*
*initialize necessary parameters for RBM*
*while precision is not satisfied*
*invoke RBM_Mapreduce ()*
*save the learned $RBM_i$*
*setup for fine_tune*
*initialize the input data according to TrainData and corresponding lables*
*Pre-trained RBMs*
*initialize necessary parameters for BP*
*while precision is not satisfied*
*invoke BPFinetune_Mapreduce ()*
*save the trained DBN*

## V. SYSTEM EXPERIMENT

### A. Scene Setting

In order to test and verify the effect of DBN fault diagnosis, the communication experiment system is built in the laboratory which includes two parts, the receiver to receive the signals sent by the transmitter and the transmitter used to transmit the communication signals. Since the receiver and the transmitter are in the same laboratory, and the channel is very short, the influence of the channel on the communicating can be ignored in the experiment. To simplify the experiment process, only the output signals of the transmitter and the receiver are measured and collected as the original data for further learning and training. Under the experimental system, the communicating experiments will conduct for several scenarios listed in Table I.

Table I Experimental scene setting

| Scene | Fault Location | Fault Phenomenon |
|---|---|---|
| Scene A | none | none |
| Scene B | receiver | Frequency drift |
| Scene C | Transmitting antenna | Low transmitting power |
| Scene D-1 | transmitter - power amplifier module –unit 1 | Low transmitting power |
| Scene D-2 | transmitter - power amplifier module - unit 2 | Low transmitting power |
| Scene D-3 | transmitter - power | Low transmitting power |

| | | |
|---|---|---|
| Scene D-4 | amplifier module - unit 3<br>transmitter - power amplifier module - unit 4 | Low transmitting power |

In the scenarios listed in the table, there are three phenomena, no fault, frequency drift and low transmission power. The frequency drift fault is set in the transmitter. The fault with low transmission power can be set by transmitting antenna and transmitter power amplifier module. The failure of four power amplifier units in transmitter power amplifier module will lead to low transmission power.

For the seven scenarios listed in the table, random communicating experiments will be done under the experimental system respectively. During the experiment, for each signal, 1024 data points are collected as a data sample. 1500 samples are collected for each scenario. For Scene A, all samples are used for diagnostic testing. For others, 1000 samples are for training and the other 500 samples for diagnostic testing.

## B. Model Training

(1) RBM Setting

The basic parameters of RBM training include the number of neurons in the visible layer, the number of neurons in the hidden layer, the learning rate, and the number of iterations. The learning rate is set to 0.1. In the visible layer, the number of neurons in the lowest RBM layer is the sample vector dimension of input data, that is, the number of sample data points, 1024. The number of neurons in the other RBM layers is the number of hidden layers in the previous RBM layer. The number of neurons in the hidden layer and the number of iterations will directly affect the learning effect of RBM. RBM learning effect can be tested by signal reconstruction. In order to ensure the objectivity and effectiveness of the test, the standard simulation signal instead of communication output signal is used as RBM input signal for RBN training, and the number of training samples is 1000.

After the training, the signal is reconstructed to evaluate the RBM capability by compared the reconstructed signal to the original signal. The evaluation parameter is the square root of the mean square error, which can be calculated by the (8). Here, $s_i^j$ represents the $j^{th}$ data point of the $i^{th}$ signal in the original signal, and $s_i'^j$ represents the $j^{th}$ data point of the $i^{th}$ signal in the reconstructed signal.

$$\varphi = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j}^{m}(s_i^j - s'^j_i)^2}{m*n}} \tag{8}$$

Under MATLAB in single machine, setting the number of iterations to 1, 5, 10, 20, 50, 100 respectively, the experimental results of RBM training and evaluating are shown in Table II.

Table II Experimental result under different iterations

| Iterative number. | $\varphi$ | Training time (s) |
|---|---|---|
| 1 | 0.6456 | 1.0116 |
| 5 | 0.3746 | 4.6767 |
| 10 | 0.3025 | 8.3471 |
| 20 | 0.2288 | 15.5793 |
| 50 | 0.1732 | 34.6245 |
| 100 | 0.1495 | 73.0844 |

Obviously, when the number of iterations increases, the square root of the mean square error of signal decreases, that is, RBM capability increases, while the training time also increases. Considering the time and performance factors, the number of iterations is set with 20.

For the lowest RBM, we set the number of iterations with 20, set the number of neurons in the hidden layer with 1000, 500, 200, 100, 50 respectively, and set the hidden note number with 500 for training and evaluating. The experimental data is shown in Table III.

Table III Experimental result under different neural number

| hidden note number | φ | Training time (s) |
|---|---|---|
| 1000 | 0.2007 | 22.4298 |
| 500 | 0.2288 | 15.5793 |
| 200 | 0.2876 | 10.0726 |
| 100 | 0.3327 | 8.2631 |
| 50 | 0.3876 | 7.2788 |

Obviously, when the hidden note number increases, the square root of the mean square error of signal decreases, that is, RBM ability increases, and training time also increases.

(2) DBN setting

In the DBN diagnosis model, there are multilayer RBMs and a classifier. Obviously, the levels of RBMs will affect the performance of DBN. In order to study the impact of the levels of cascading RBMs on the performance of DBN, the depth of DBN is set to 5, 6, 7 and 8 respectively, which the levels of cascading RBMs is 4, 5, 6 and 7 respectively with the number of neurons in each layer setting to 512-400-300-200, 512-400-300-200-200, 512-400-400-300-300-200-200. Standard signal by simulation is used to train the cascading RBMs with setting the number of iterations as 20, and the learning rate as 0.1. After learning, the signal is reconstructed layered from the top to the bottom. Then, the square root of the mean square error can be calculated by comparing with the original data. The experimental data in the single machine environment is as shown in Table IV.

Table IV Experimental result under different deepth

| DBN depth | RBM levels | RBM numbers | $\varphi$ | Train time (s) |
|---|---|---|---|---|
| 5 | 4 | 3 | 0.1501 | 34.1116 |
| 6 | 5 | 4 | 0.1396 | 41.5737 |
| 7 | 6 | 5 | 0.1221 | 51.4683 |
| 8 | 7 | 6 | 0.1153 | 62.0457 |

Obviously, the deeper the DBN is, the lower the square root of the mean square error of signal is. That is, the DBN capability is increased, and the training time is also increased. Considering the time and performance factors, we set the DBN depth with 5 in the following experiments. It includes 4 layers and 3 cascading RBMs in total.

In Table V, the data are the pre-training results under unsupervised learning. The fine-tuning of BP algorithm with supervised learning is also included in DBN algorithm.

According to the fine-tuning algorithm, all RBM weight values W can be fine-tuned in the 5-layer DBN model. Using adjusted DBN to reconstruct the signal, the square root of the mean square error of signal, can be reduced to 0.08297 under other conditions without changed.

Table V Training scene of DBN train

| DBN Training Model | Input | Output |
|---|---|---|
| DBN-1 | 1000 training samples for scene A and scene B respectively | Receiver frequency drift / fault-free |
| DBN-2 | 1000 training samples for scene A and scene B respectively, 250 training samples for each type of scene D | Transmit antenna failure / Transmitter power amplifier module failure / fault-free |
| DBN-3 | 1000 training samples for scene D-1, scene D-2, scene D-3 and scene D-4 | Unit 1/2/3/4 of transmitter power amplifier module failure |

Determined the structure and parameters of DBN, the model can be trained basing on the requirements of fault diagnosis. According to the experiment scene, experiment type and experiment content, the following types of DBN model training are needed.

### *C. Fault Diagnosis*

Applying training samples generated by the experimental system, after the training of various DBN fault diagnosis models, the fault diagnosis experiment can be carried out. According to the diagnosis requirements, the fault diagnosis experiments listed in Table VI would be processed.

Table VI Experimental type for communication system fault diagnosis

| diagnosis experiment | Type No. | Training Model | Test samples |
|---|---|---|---|
| Diagnose-1 | Type-1 | DBN-1 | 500 test samples for scene A and scene B respectively |
| Diagnose -2 | Type-2 | DBN-2 | 500 test samples for scene A and scene C respectively 500 test samples for scene D randomly |
| Diagnose -3 | Type-3 | DBN-3 | 500 test samples for scene D-1/2/3/4 |

The statistics for diagnose-1 are shown in Table VII.

Table VII Diagnosis data of diagnose-1 experiment

| Test samples | Receiver frequency drift | fault-free | total |
|---|---|---|---|
| scene A: fault-free | 4 (0.8%) | 496 (99.2%) | 500 (100%) |
| scene B: Receiver frequency drift | 498 (99.6%) | 2 (0.4%) | 500 (100%) |
| total | 502 | 498 | 1000 |

According to the statistical data in Table VII, the fault diagnosis indexes of diagnosis-1 experiment can be calculated which accuracy = 99.6%, positive = 0.8% and negative = 0.4%. It can be seen that DBN method can achieve better results for single fault diagnosis of receiver frequency drift.

The statistics for diagnose-2 are shown in Table VIII.

Table VIII Diagnosis data of diagnose-2 experiment

| Test samples | Transmit antenna failure | Transmitter power amplifier module failure | fault-free | total |
|---|---|---|---|---|
| scene A: fault-free | 3 (0.6%) | 5 (1.0%) | 492 (98.4%) | 500 (100%) |
| scene C: Transmit antenna failure | 462 (92.4%) | 32 (6.4%) | 6 (1.2%) | 500 (100%) |
| scene D: Transmitter power amplifier module failure | 27 (5.4%) | 465 (93.0%) | 8 (1.6%) | 500 (100%) |
| total | 492 | 502 | 506 | 1500 |

According to the statistical data in Table VIII, the fault diagnosis indexes of diagnosis-2 experiment can be calculated which shown in Table IX. It can be seen that DBN method can achieve better results for multiple fault diagnosis of transmitter antenna fault and transmitter power amplifier module fault.

Table IX Diagnosis performance index under diagnose-2 experiment

| fault | *Accuracy* | *Positive* | *Negative* |
|---|---|---|---|
| Transmit antenna failure | 92.4% | 3% | 7.6% |
| Transmitter power amplifier module failure | 93.0% | 3.7% | 7% |
| average | 92.7% | 3.35% | 7.3% |

The statistics for diagnose-3 are shown in Table X.

Table X Diagnosis data of diagnose-3 experiment

| Test samples | Unit 1 fault | Unit 2 fault | Unit 3 fault | Unit 4 fault | total |
|---|---|---|---|---|---|
| scene D-1: Unit 1 fault | 402 (80.4%) | 40 (8.0%) | 31 (6.2%) | 27 (5.4%) | 500 (100%) |
| scene D-2: Unit 2 fault | 28 (5.6%) | 423 (84.6%) | 30 (6%) | 19 (3.8%) | 500 (100%) |
| scene D-3: Unit 3 fault | 22 (4.4%) | 32 (6.4%) | 416 (83.2%) | 30 (6%) | 500 (100%) |
| scene D-4: Unit 4 fault | 28 (5.6%) | 37 (7.4%) | 44 (8.8%) | 391 (78.2%) | 500 (100%) |
| total | 480 | 532 | 521 | 467 | 2000 |

According to the statistical data in Table X, the fault diagnosis indexes of diagnosis-3 experiment can be calculated which shown in Table XI. It can be seen that DBN method can achieve better results for nuances fault diagnosis of different unit fault in transmitter power amplifier module.

Table XI Diagnosis performance index under diagnose-3 experiment

| fault | *Accuracy* | *Positive* | *Negative* |
|---|---|---|---|
| Unit 1 fault | 80.4% | 5.2% | 19.6% |
| Unit 2 fault | 84.6% | 7.27% | 15.4% |
| Unit 3 fault | 83.2% | 7% | 16.8% |
| Unit 4 fault | 78.2% | 5.07% | 21.8% |
| average | 81.6% | 6.14% | 18.4% |

## VI. RESULT ANALYSIS

With traditional fault diagnosis methods, communication system fault diagnosis maybe intricacies as following.

(1) Different signals need to be collected for different faults. For example, the output signal of the receiver should be collected for diagnosis-1, the output signal of the transmitter should be collected for diagnosis-2, and the output signal of the transmitter power amplifier module should be collected for diagnosis-3.

(2) After signal collecting, it generally needs to be analyzed to obtain the relevant index parameters which are necessary for supporting the fault diagnosis. For example, signal spectrum distribution parameters should be obtained for diagnosis-1 and needs to obtain the signal spectrum distribution parameters, and the signal power parameters are needed for diagnosis-2 and diagnosis-3.

(3) Additional installation may be needed as required. For example, a dummy load between the transmitter and the antenna should be added for diagnosis-2. It is used to check the transmitter's working condition when connecting the dummy load, so as to further distinguish the transmitter antenna fault and the transmitter power amplifier module fault.

Compared with the traditional fault diagnosis methods, the following characteristics of DBN fault diagnosis can be found from the results of the above experiments in this paper.

(1) DBN model has high accuracy in fault diagnosis. According to the experimental results of diagnosis-1 and diagnosis-2, we can find that DBN fault diagnosis has high accuracy, low false alarm rate and false alarm rate. It has the highest level of the existing diagnosis methods.

(2) DBN model can distinguish subtle faults. The experimental results of diagnosis-2 and diagnosis-3 show that DBN fault diagnosis can further distinguish the different fault causes of the same fault phenomenon. With high reliability, it fills in the gap of communication system fault diagnosis to some extent. Diagnosis-3 experiments show that DBN model can distinguish the specific fault units of the transmitter power amplifier module, whereas the existed diagnosis methods cannot complete such subtle fault location.

(3) The process of DBN fault diagnosis is simple and friendly. Unlike the complexity of the algorithm, the process of DBN fault diagnosis is very simple. Although the experiments of diagnosis-1, diagnosis-2 and diagnosis-3 are aimed at different types and vary parts of faults, they all only need to collect the output signals of corresponding equipment.

Due to the limitations of traditional fault diagnosis methods, it is impossible to realize real-time diagnosis and online diagnosis for the communication system at present. It also cannot identify similar faults, and cannot directly locate faults to module units. DBN fault diagnosis is not bound by these limitations. It can realize real-time diagnosis and online diagnosis, can identify similar faults, and can locate faults in the unit of module. Following the development of GPU, parallel technology and deep learning, DBN fault diagnosis technology is becoming more and more mature. It will replace the traditional fault diagnosis method to become the mainstream of communication system fault diagnosis.

## VII. CONCLUSION

Based on the fault analysis of the communication system, the limitations of traditional fault diagnosis algorithm are proposed. Using deep confidence network deep learning method, the DBN fault diagnosis framework for communication system is constructed. On this basis, the DBN fault diagnosis algorithm is designed and realized, and parallelization is carried out based on MapReduce. Finally, the effectiveness and advancement of the DBN fault diagnosis method are verified by the experimental system.

Further, the research will be applicated to the health management of communication equipment, and it will be optimized during application. It will provide advanced technical support and software program for the health of communication equipment and provide strong support for enhancing the communication support capability of navy.

## ACKNOWLEDGMENT

This research was supported by a grant from the Doctoral Research Starting Fund of Wuchang Institute of Technology (No. 2018BSJ04).

## REFERENCES

[1] S. Haykin, "Communication Systems," Beijing, China: Publishing House of Electronics Industry, 2018, pp. 34- 123.

[2] X. H. Tong and B. Zhao, "Military Communication System," Beijing, China: Publishing House of Electronics Industry, 2020, pp. 67-145.

[3] C. P. Mu, "Military Communication Network Technology," Beijing, China: Beijing Institute of Teechnology Press, 2019, pp. 89-162.

[4] K. Yan, Y. Zhang, Y. Yan, C. Xu and S. Zhang, "Fault diagnosis method of sensors in building structural health monitoring system based on communication load optimization," Computer Communications, vol. 159, pp. 310-316, June 2020.

[5] S. A. Naseem, R. Uddin, A. S. Alghamdi, M. H. Uddin and A. A. Bilal, "Ethernet-based fault diagnosis and control in smart grid: a stochastic analysis via Markovian model checking," Journal of Electrical Engineering & Technology, vol. 14, no. 6, pp. 2289-2300, October 2019.

[6] J. L. Li, J. C. Xie, Z. J. Yang and J. Li, "Fault diagnosis method for a mine hoist in the internet of things environment," Sensors (Basel, Switzerland), vol. 18, no. 6, pp. 1920, 2018.

[7] Q. K. Zhu, "A new methodology and CAD programs to detect two serious faults in VLSI design: HV/LV connection faults and floating gate faults," WSEAS Transactions on Circuits and Systems, vol. 18, no. 33, pp. 220-235, 2019.

[8] D. Y. Tu'uau, T. Marica and M. H. Assaf, "Electric power system fault analysis", WSEAS Transactions on Circuits and Systems, vol 19, no. 3, pp. 19-27, 2020.

[9] S. A. Herve, Y. T. Aurelien, H. S. I. Judith and N. N. Leandre, "Networked iterative learning fault diagnosis algorithm for systems with sensor random packet losses, time-varying delays, limited communication and actuator failure: application to the hydroturbine governor system," WSEAS Transactions on Systems and Control, Vol. 16, no. 20, pp. 244-252, 2021.

[10] I. Goodfellow, Y. Bengio and A. Courville, "Deep Learning," Beijing, China: Posts & Telecom Press, Beijing: 2017, pp. 66-142.

[11] V. Peiris, N. Sharon, N. Sukhorukova and J. Ugon, "Generalised rational approximation and its application to improve deep learning classifiers," Applied Mathematics and Computation, vol. 389, no. 1, January 2021.
[12] H. Ren, N. Li, Y. Chai, J. F. Qu, Q. Tang and L. Huang, "The input pattern problem on deep learning applied to signal analysis and processing to achieve fault diagnosis," Science China (Information Sciences), vol. 62, no. 12, pp. 269-271, 2019.
[13] S. Zhang, Y. Chen, W. Y. Zhang and R. J. Feng, "A novel ensemble deep learning model with dynamic error correction and multiobjective ensemble pruning for time series forecasting," Information Sciences, vol. 544, no. 1, pp. 427-445, January 2021.
[14] C. L. Zhang, Y. G. He, L. F. Yuan and S. Xiang, "Analog circuit incipient fault diagnosis method using dbn based features extraction," Ieee Access, vol. 6, pp. 23053- 23064, April 2018.
[15] M. Fatahi, M. Shahsavari, M. Ahmadi, A. Ahmadi, P. Boulet, P. Devienne, "Rate-coded DBN: An online strategy for spike-based deep belief networks," Biologically Inspired Cognitive Architectures, vol. 24, pp. 59-69, April 2018 .
[16] C. H. Hu, H. Pei, X. S. Si, D. B. Du, Z. N. Pang and X. Wang, "A prognostic model based on DBN and diffusion process for degrading bearing," IEEE Transactions on Industrial Electronics, vol. 67, no. 10, pp. 8767-8777, October 2020.
[17] T. F. Zhang, Z. Li and Z. H. Deng, "Hybrid data fusion DBN for intelligent fault diagnosis of vehicle reducers," Sensors, vol. 19, no. 11, pp. 2504, April 2018.
[18] G. M. Wang, J. F. Qiao, J. Bi, Q. S. Jia and M. C. Zhou, "An adaptive deep belief network with sparse restricted Boltzmann machines," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 10, pp. 4217-4228, 2019.
[19] S. Pirmoradi, M. Teshnehlab, N. Zarghami and A. Sharifi, "The self-organizing restricted boltzmann machine for deep representation with the application on classification problems," Expert Systems with Applications, vol. 149, no. 2, pp. 113286, July 2020.
[20] N. Zhang, S. F. Ding, J. Zhang and Y. Xue, "An overview on restricted Boltzmann machines," Neurocomputing, vol. 275, pp. 1186-1199, January 2018.
[21] D. L. Chiang, S. K. Wang, Y. Y. Wang, Y. N. Lin and H. W. Ho, "Modeling and analysis of hadoop mapreduce systems for big data using petri nets," Applied Artificial Intelligence, vol. 35, no. 1, pp. 80-104, 2020.
[22] M. T. Wu, "MapReduce analytics-based fast search algorithms," Journal of the Chinese Institute of Engineers, vol. 43, no. 8, pp. 831-837, 2020.
[23] H. I. Abaker Targio, A. N. Badrul, M. Mohsen, E. Ahmed, H. Chiroma, A. Firdaus, et al., "MapReduce scheduling algorithms: A review," Journal of Supercomputing, vol. 76, no. 7, pp. 4915-4945, December 2018.
[24] W. T. Wu and S. C. Xu, "Application of MapReduce parallel association mining on IDS in cloud computing environment," Journal of Intelligent & Fuzzy Systems, vol. 39, no. 2, pp. 1915-1923, 2020. Juan Li was born in Huaian China in January 1977. She got a doctorate in communication and information systems in 2015 in Naval University of Engineering locating in Wuhan China. The major direction of research is Computer science and technology. Bin Chen was born in Nanjing China in December 1975. He got a doctorate in communication and information systems in 2017 in Naval University of Engineering locating in Wuhan China. The major direction of research is Communication information system.

Juan Li was born in Huaian China in January 1977. She got a doctorate in communication and information systems in 2015 in Naval University of Engineering locating in Wuhan China. The major direction of research is Computer science and technology.
Bin Chen was born in Nanjing China in December 1975. He got a doctorate in communication and information systems in 2017 in Naval University of Engineering locating in Wuhan China. The major direction of research is Communication information system.

## Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Juan Li propsed DBN model for communication system fault diagnosis and then designed the corresponding algorithms and provided parallel solutions.
Bin Chen built the experimental system and carried out the simulation and statistics.