

FENLING FENG, Ph.D.¹

E-mail: ffl0731@163.com

WAN LI, Master's Degree Candidate ¹

E-mail: lingleeabd@126.com

QIWEI JIANG, Ph.D.¹

(Corresponding author)

E-mail: jqw@csu.edu.cn

¹ Central South University

School of Traffic and Transportation Engineering

Changsha 410075, China

Safety and Security in Traffic

Original Scientific Paper

Submitted: 17 May 2017

Accepted: 5 Feb. 2018

RAILWAY TRAFFIC ACCIDENT FORECAST BASED ON AN OPTIMIZED DEEP AUTO-ENCODER

ABSTRACT

Safety is the key point of railway transportation, and railway traffic accident prediction is the main content of safety management. There are complex nonlinear relationships between an accident and its relevant indexes. For this reason, triangular gray relational analysis (TGRA) is used for obtaining the indexes related to the accident and the deep auto-encoder (DAE) for finding out the complex relationships between them and then predicting the accident. In addition, a nonlinear weight changing particle swarm optimization algorithm, which has better convergence and global searching ability, is proposed to obtain better DAE structure and parameters, including the number of hidden layers, the number of neurons at each hidden layer and learning rates. The model was used to forecast railway traffic accidents at Shenyang Railway Bureau, Guangzhou Railway Corporation, and Nanchang Railway Bureau. The results of the experiments show that the proposed model achieves the best performance for predicting railway traffic accidents.

KEYWORDS

railway traffic accident; deep auto-encoder; particle swarm optimization algorithm;

1. INTRODUCTION

An accident affecting normal train operation is called a railway traffic accident, such as conflict, derailment, fire, or explosion. Railway is the main artery of Chinese economy related to production development, standard of living, and social welfare. It occupies a very important position in Chinese transportation systems. Given that railway transportation is characterized by high speed, high density, and heavy loads, traffic security is facing new demands and challenges. Transportation enterprises must compensate for losses incurred when goods are lost, short, deteriorated, contaminated, or damaged [1]. Accurate prediction of railway traffic accidents plays a crucial role in railway safety warning systems and reduces the losses of transportation enterprises.

Generally speaking, three kinds of accident prediction methods have been frequently used in accident safety analysis: fault tree analysis, Petri nets, and Bayesian network. However, these methods analyze the accident in the local view, focusing on point-to-point or part-to-part analysis, which are insufficient for complicated railway accidents [2]. MA [2] proposed the use of complex networks to deal with the relationships between factors causing railway accidents. The accident safety analysis methods mentioned above only focus on analyzing the relationships between the causes and accidents or among the causes. Therefore, the number of railway accidents cannot be predicted well by these methods. Besides, these methods are too complex.

At present, there are few studies on railway traffic accident prediction, and research is often concentrated on analyzing accidents in railway and road intersections [3][4][5][6]. In addition, there are also studies focused on analyzing high-speed railway accidents. Wen [7] analyzed train operation conflict prediction in terms of high-speed railway safety. Studies on railway traffic accidents at ordinary speed are scarce, despite the necessity of railway traffic accident prediction.

Many models have been used for prediction: linear regression, the time series method [8], gray system theory [9][10], support vector machine [11][12], system dynamics [13], and artificial neural network [14][15][16]. There are complicated nonlinear relationships between the accident and the influencing factors, meaning that linear regression does not predict accidents accurately. The time series method is good at predicting series with regularity, whereas accidents are uncertain and unexpected, and this method is not suitable for accident prediction. The gray system theory is simple, computes fast, and it can present a good result for short-term forecasting, but it is not ideal for the fluctuation system. The support vector machine features good nonlinear mapping, which transforms high-dimensional data into low-dimensional data, but

this method is difficult to implement for large-scale training samples, and it is not suitable for problems of multiple classification. System dynamics can be used to analyze qualitatively and quantitatively the relationship among the factors in the system and reflect the actual situation. However, it is heavily dependent on builder's understanding of the system movement mechanism. Artificial neural network is characterized by good self-adaptability, self-organization, and self-learning, which overcomes the shortcomings of other forecasting methods in solving nonlinear, uncertain, and time-varying systems, and makes the forecast more accurate.

While artificial neural networks are still being improved, they are suitable for various tasks. Back-propagation neural network (BPNN) is one of the most maturely studied neural network algorithms. It has good self-learning, self-adaptation, robustness, and generalization capacities. However, the back-propagation algorithm has some disadvantages, such as poor rate of convergence, and getting stuck in local minimum easily. Furthermore, the back-propagation algorithm is based on the gradient information of error function. When problems are complex, or the gradient information is hard to obtain, it may be helpless [17]. In order to solve these downsides, Hinton et al. improved the previous shallow structure of neural network and put forward the concept of deep learning and its training strategy, creating the deep auto-encoder (DAE) [18]. DAE eliminates the huge workload of manual extraction, of characteristics from large amounts of data and improves extraction efficiency. It shows a strong capacity to learn the essential features of input data from a few labeled samples and a large number of unlabeled data, and hierarchically represents the characteristics that have been learned. Bo et al. [19], Li et al. [20], and Ong et al. [21] used auto-encoders to predict students' performance, the release of NO_x, and PM_{2.5}, respectively. The students' performance indicates the student final examination score. These studies obtained optimal predictive results. However, in these studies the number of hidden layers, the number of neurons in each layer and learning rates were decided by subjective or multiple experiments, and this affected the ability of extracting data features for auto-encoders. In recent years, Kuremoto et al. [22] and Shao et al. [23] used particle swarm optimization (PSO) to optimize the number of neurons of a deep belief network, which largely reduced the subjectivity. This paper is also an effort to use this method to optimize DAE.

This paper mainly provides the following three innovations: (i) DAE is applied to predict railway traffic accidents. (ii) The improved PSO (IPSO) is used for DAE to decide the number of hidden layers, the number of neurons of each hidden layer, the learning rate of each hidden layer when reconstructing input data during

pre-training and the learning rate of back-propagation algorithm during fine-tuning. (iii) In order to find the appropriate number of layers and neurons of hidden layers quickly, IPSO is proposed, which features good global search ability and fast convergence speed. At Shenyang Railway Bureau, Guangzhou Railway Corporation, and Nanchang Railway Bureau traffic accidents are forecast through IPSO-DAE. Additionally, triangular gray relational analysis (TGRA) [24] is applied to find the indexes related to railway traffic accidents.

The remainder of the paper is organized as follows. Section 2 briefly describes the methods this paper has used, including IPSO. DAE and experimental procedure are described in Section 3. Section 4 reports and discusses the empirical results, followed by conclusions in Section 5.

2. METHODS AND MODEL

2.1 Nonlinear weight-changing particle swarm optimization algorithm

Particle swarm optimization (PSO) is a heuristic algorithm. The PSO algorithm first randomly initializes the position and velocity of a random population of particles. Each particle i is defined by its position vectors X_i^t in the space of the parameters to be optimized and by a random velocity V_i^t . In the following iteration, the particle moves according to its velocity and is evaluated according to the fitness function $f(X)$, which is related to the problem to be optimized. The value of the fitness function is compared with the best previously obtained value. The best value ever obtained for each particle is stored as $pbest_i$, which is actually the personal optimum searching, and the best value among all $pbest_i$ is stored as $gbest$, which is actually the global optimum searching. The velocity of the particle is then updated by:

$$V_i^{t+1} = w \cdot V_i^t + c_1 \cdot rand \cdot (pbest_i - X_i^t) + c_2 \cdot rand \cdot (gbest - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

where w is the inertia weight; $rand$ generates a random number between 0 and 1; c_1, c_2 are the velocity coefficients.

According to Equation 1, it is well established that w controls the convergence and exploration ability effectively. Equation 1 actually determines that the particle velocity changes in a linear way. It has two problems: (i) if particle swarm searches optimum value at the beginning, it is hoped to converge to global optimum quickly, but invariant w decreases the convergence speed of the algorithm; (ii) in late operation of the algorithm, invariant w leads to the decline of local search ability and decrease of particle diversity. Therefore, two ways are proposed to improve them:

$$w = w_{\max} - (w_{\max} - w_{\min}) \cdot \arcsin \frac{t}{t_{\max}} \cdot \frac{2}{\pi} \quad (3)$$

$$\frac{\partial w}{\partial t} = (w_{\max} - w_{\min}) \cdot \frac{2}{\pi} \cdot \frac{1}{t_{\max} \cdot \sqrt{1 - \left(\frac{t}{t_{\max}}\right)^2}} \quad (4)$$

where w_{\max} is the maximum inertia weight, and w_{\min} is the minimum inertia weight; t_{\max} is the maximum iteration time. The values of Equation 4 are shown in Figure 1. In the expression above, at the beginning of operation (Figure 1), w close to w_{\max} guarantees the global search capability of the algorithm. With the increasing of t , nonlinear decrease of w guarantees the local search capability. When operating near the end, w decreases rapidly. Then the balance between global search and local search capability can be adjusted flexibly.

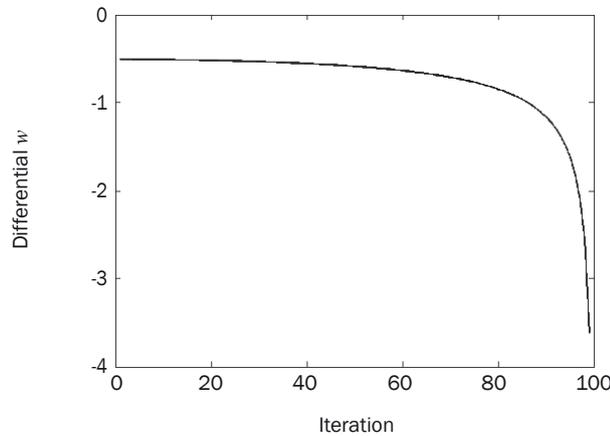


Figure 1 – The differential of w

In order to make the PSO expand search space that is constantly narrow with iteration and help particles jump out of the best position that has been searched, a genetic algorithm mutation method is used here for maintaining particle diversity. After velocity and position of the particle are updated, it may be initialized according to probability p .

2.2 Deep auto-encoder

In 2006, Hinton improved the structure of a previous auto-encoder, and DAE was created [18]. DAE is pre-trained firstly by an unsupervised, layer by layer, and greedy algorithm and then fine-tuned using the back-propagation algorithm to optimize all parameters of a whole neural network. This method improves the performance of a neural network significantly and reduces the probability of easily falling into a local optimum. DAE extracts characteristics from non-labeled and complex high-dimensional data, and then the structure of a deep learning neural network that presents the distributed features of original data is obtained [25].

2.2.1 Basic auto-encoder

It is necessary to introduce the basic auto-encoder before constructing a deep auto-encoder. According to [26], a one-layer auto-encoder is taken as an example which consists of an encoder and a decoder (Figure 2). The mapping function is usually non-linear, and the following is a common form:

$$h_i = f(x_i) = \frac{1}{1 + \exp(-(W_1 x_i + b_1))} \quad (5)$$

where W_1 is the encoding weight; b_1 is the corresponding bias vector. The decoder seeks to reconstruct the input x_i from its hidden representation h_i . The transformation function has a similar formulation:

$$x'_i = g(h_i) = \frac{1}{1 + \exp(-(W_2 h_i + b_2))} \quad (6)$$

where W_2, b_2 are the decoding weight and the decoding bias vector, respectively. The auto-encoder model aims to learn a useful hidden representation by minimizing the reconstruction error. Thus, given N training samples, the parameters $W_1, W_2, b_1,$ and b_2 can be resolved by the following optimization problem:

$$L(x_i, x'_i) = \min \frac{1}{N} \sum_{i=1}^N \|x_i - x'_i\|^2 \quad (7)$$

where L is reconstruction error function. The deep auto-encoder is constructed by stacking multiple one-layer auto-encoders. That is, the hidden representation of the previous one-layer auto-encoder is fed as input of the next one.

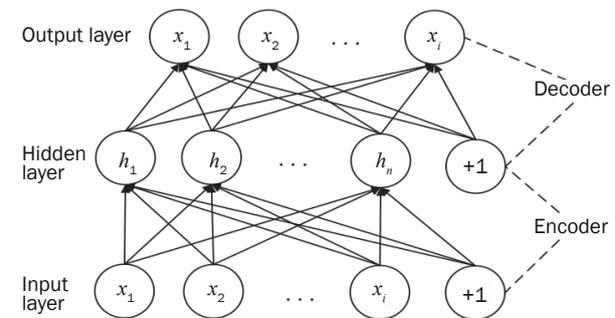


Figure 2 – Basic auto-encoder

2.2.2 Pre-training of DAE

The purpose of pre-training is to increase the performance of initialized weights and bias. The input layer and the hidden layers of DAE are initialized by unsupervised method, then the layer by layer greedy algorithm is used to train each hidden layer into an auto-association unit in order to get the input data reconstructed. The activation function for DAE here is the sigmoid function. The detailed procedure is as follows: **Step 1:** The first layer of the neural network is trained by reconstructing input samples according to Equations 5-7;

Step 2: The output of each hidden layer is taken as the input of the next layer. The next layer is trained by reconstruction of the input, and the error between input and output is controlled in a definite scope;
 Step 3: Repeat Step 2 until all hidden layers are trained;
 Step 4: The output of the last hidden layer is used as the input of the last layer of neural network, and the output of the last layer is sample labels. Then the weights and bias at the last layer of the neural network are initialized.

2.2.3 Fine-tuning of DAE

Building a DAE requires fine-tuning, and the back-propagation algorithm is usually used to accomplish this task. The input layer, the output layer, and all the hidden layers of the encoder are considered as a whole, then a supervised learning algorithm is used to further adjust the trained neural network. The detailed procedure is as follows:

Step 1: The neural network is initialized using the weights and bias that have been obtained by pre-training;
 Step 2: Sample data is used as the input of the neural network, and the back-propagation algorithm is applied to train neural network;
 Step 3: Compute the error between sample label and output of the neural network, then adjust the weights and bias of the neural network according to errors.
 Step 4: Repeat Step 2 and Step 3 until the error meets requirement, or the iteration time is achieved.

It is important to note that the mapping function of the output layer is linear:

$$\hat{y} = W_o h_i + b_o \tag{8}$$

where \hat{y} represents the output of DAE; W_o and b_o represent the weights and bias of output layer. This linear function not only does not affect the fitting ability of the model, but also makes the output of the model

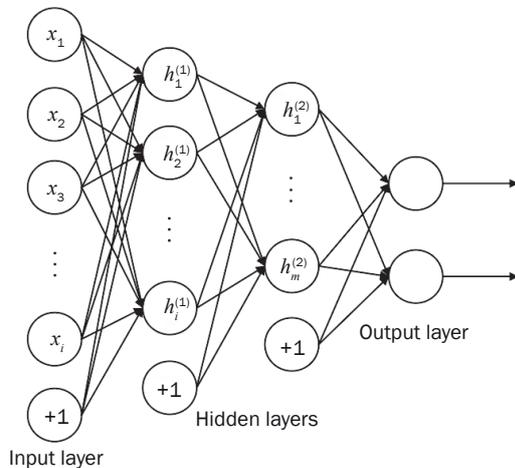


Figure 3 – The structure of a trained DAE

in any range, which enables prediction beyond the expected number of accidents. A DAE, containing two hidden layers and training steps to be completed, is shown in Figure 3.

3. THE PROPOSED MODEL AND EXPERIMENT PROCEDURE

3.1 Optimized deep auto-encoder

In this paper, we use the IPSO mentioned in Section 2.1 to decide three kinds of hyper DAE parameters: (i) the number of neurons of each hidden layer; (ii) the learning rate of each hidden layer when reconstructing input data during pre-training; (iii) the learning rate of back-propagation algorithm during fine-tuning. The number of hidden layers and neurons at each hidden layer has a direct impact on fitting ability and predicting performance of DAE. The learning rate of each hidden layer when reconstructing input data during pre-training affects the performance of input data reconstruction. The learning rate of the back-propagation algorithm during fine-tuning influences the final prediction results of the model. Traditionally, these parameters were decided by multiple experiments or experience, which limited the prediction ability of the model. Therefore, it is necessary to use IPSO to optimize these parameters.

In this paper, the DAE has $j(j=1,2,3,4)$ hidden layer(s), then each particle can be expressed as $X(n_1, n_2, \dots, n_j, \epsilon_1^P, \epsilon_2^P, \dots, \epsilon_j^P, \epsilon^F)$, n_j represents the number of neurons in the j -th hidden layer, ϵ_j^P represents the learning rate of the j -th hidden layer when reconstructing input data during pre-training, ϵ^F represents the learning rate of the back-propagation algorithm during fine-tuning.

Data is divided into training samples, validation samples, and test samples. The fitness function of IPSO is as follows:

$$fitness = 0.5 \cdot \frac{1}{J} \cdot \sum_{j=1}^J \frac{\hat{y}_t^j - y_t^j}{y_t^j} \cdot 100 + 0.5 \cdot \frac{1}{K} \cdot \sum_{k=1}^K \frac{\hat{y}_v^k - y_v^k}{y_v^k} \cdot 100 \tag{9}$$

where y_t^j and y_v^k represent the expected output values of training samples and validation samples, respectively. \hat{y}_t^j and \hat{y}_v^k represent the output values of training samples and validation samples, respectively. Most previous studies only used the fitting error of training samples as fitness function, which may lead to an overfitting model with sub-optimal performance. The validation error of validation samples reflects the predictive performance of the trained model directly, so here the fitness function includes the fitting error of training samples and validation error of validation samples. In this study, the error of training samples and the error of verification samples have the same weight, that is, 0.5, and their sum is used as the fitness function of the model.

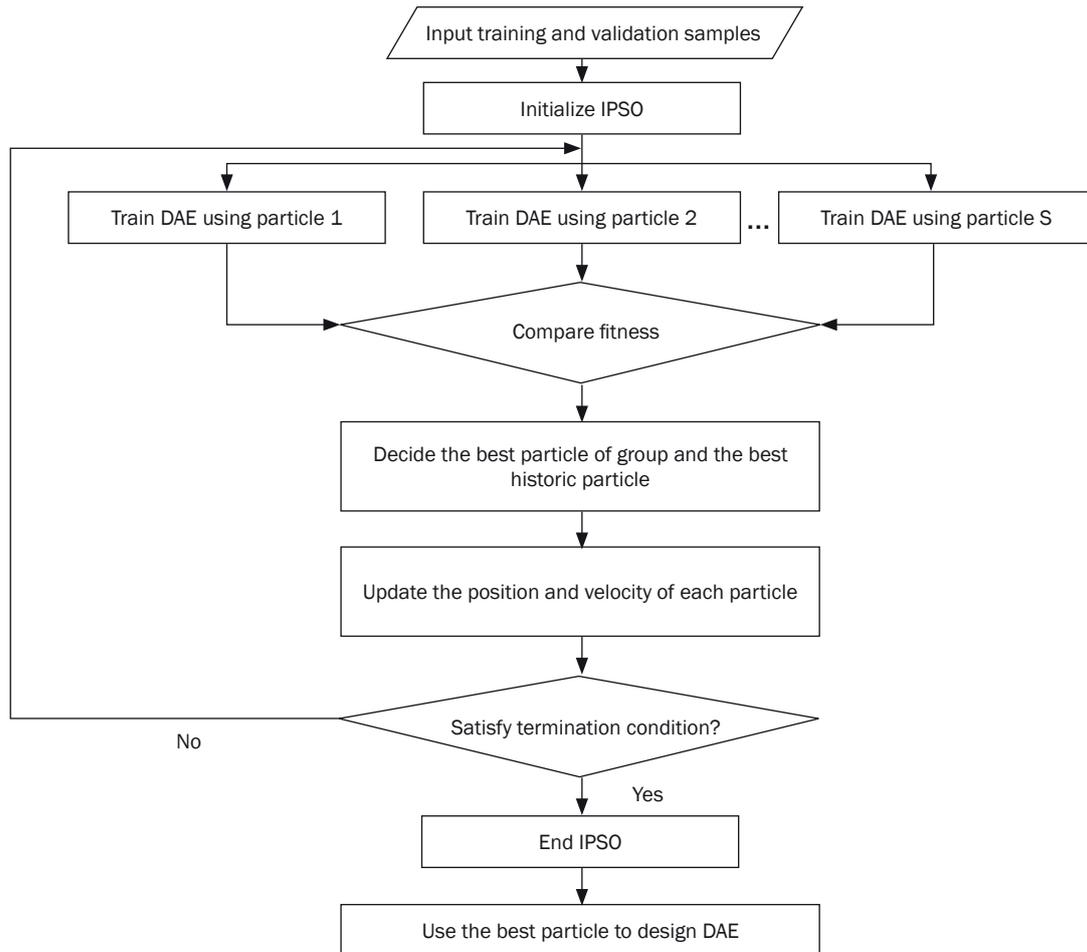


Figure 4 - The flow chart of the optimized DAE using IPSO

3.2 Experiment procedure

The flow chart of the optimization process is shown in Figure 4. A summarized IPSO algorithm used to decide the structure of DAE is shown as follows:

Step 1: Decide the population size of particles and limitation of iteration number.

Step 2: Initialize the start position X_i^0 and the start velocity V_i^0 of each particle.

Step 3: Evaluate each particle using the fitness function (Equation 9) mentioned above, and find the best position of the particle $pbest_i$ from its history, and the best particle position of the swarm $gbest$.

Step 4: Renew positions and velocities of particles by Equation 1 and Equation 2, respectively.

Step 5: If the fitness function converged, or t reaches the maximum value, finish the algorithm. Otherwise, return to Step 3.

4. EXPERIMENTS AND RESULTS ANALYSIS

In this paper, data includes railway traffic accidents recorded at Shenyang Railway Bureaus, Guangzhou Railway Corporation, and Nanchang Railway Bureau from 1999 to 2013. Besides, data also includes

indexes related to railway traffic accidents from 1999 to 2013. The data was found in the China Railway Yearbook and divided into training samples, validation samples, and test samples. The data for the first eleven years were taken as the DAE training samples, the data for the twelfth and thirteenth years as validation samples, and the data for the last two years as test samples.

Railway traffic accidents are divided into four grades: extremely serious, serious, large, and general. Death toll, number of serious injuries, and direct economic loss were selected as the main indexes for accident ranking. Specific classification bases are shown in Table 1. If one of the 3 indexes is achieved, the corresponding grade of the accident is formed.

Because extremely serious and serious accidents are highly random, sudden, and occur infrequently, they are classified as abnormal data and are not included in the total number of accidents. Although large accidents occur much more rarely than general accidents, they do happen a few times every year. General accidents occur often and break railway operation, therefore accidents in this article include only large and general accidents.

Table 1 – Classification bases for railway traffic accidents

Grade	Death toll	Serious injuries	Economic losses (ten thousand yuan)
Extremely serious	>30	>100	>10,000
Serious	10~30	50~100	5,000~10,000
Large	3~10	10~50	1,000~5,000
General	<3	<10	500~1,000

4.1 Data pre-processing

Data normalization

The data, including railway traffic accidents and influencing indexes, are normalized by the following formula:

$$y = (y_{max} - y_{min}) \cdot \frac{x - x_{min}}{x_{max} - x_{min}} + y_{min} \tag{10}$$

where y and x are the normalized value and original value, respectively. x_{max} and x_{min} represent the maximum value and the minimum value of original data series, y_{max} and y_{min} represent the maximum value and the minimum value after normalization. In order to make the model fit better and calculate error conveniently, here $y_{max} = 0.9$, $y_{min} = 0.1$.

Selection of railway bureaus

In 2005, the railway system was reformed, from 15 railway bureaus to 18. Twelve of those railway bureaus did not change their jurisdiction. Therefore, the pre-2006 data for the twelve unchanged railway bureaus can be used. The twelve railway bureaus include Harbin Railway Bureau (HARB), Shenyang Railway Bureau (SYRB), Hohhot Railway Bureau (HORB), Jinan Railway Bureau (JNRB), Shanghai Railway Bureau (SHRB), Nanchang Railway Bureau (NCRB), Guangzhou Railway Corporation (GZRC), Nanning Railway Bureau (NNRB),

Chengdu Railway Bureau (CDRB), Kunming Railway Bureau (KMRB), Lanzhou Railway Bureau (LZRB), and Urumqi Railway Bureau (URB).

Due to limited article length, it is impossible to predict accidents for each railway bureau. Therefore, the accidents of the 12 railway bureaus are clustered, and a railway bureau in each class is randomly selected for accident prediction. After data is normalized according to Equation 10, railway traffic accidents are clustered using the K-means clustering method, in which Euclidean distance is used. In this paper, railway traffic accidents are clustered into 3 groups.

The clustering results are shown in Figure 5. As can be seen, the first class has the most railway bureaus. Its trend shows a large number of accidents in 1999, which decreased from 2000 to 2007, and increased again after 2008. The second class includes 3 railway bureaus. Its trend shows a large number of accidents in 1999, which decreased from 1999 to 2002, and was the least in 2003, then increased gradually. The third class contains two railway bureaus, whose trend shows a large number of accidents in 1999, decreased each year until it grew suddenly in 2003, decreased greatly in 2008, and increased gradually after 2008. On the whole, the number of accidents was relatively large in 1999, which had a certain connection with the running speed increasing for the second time in October 1999. The increased speed required

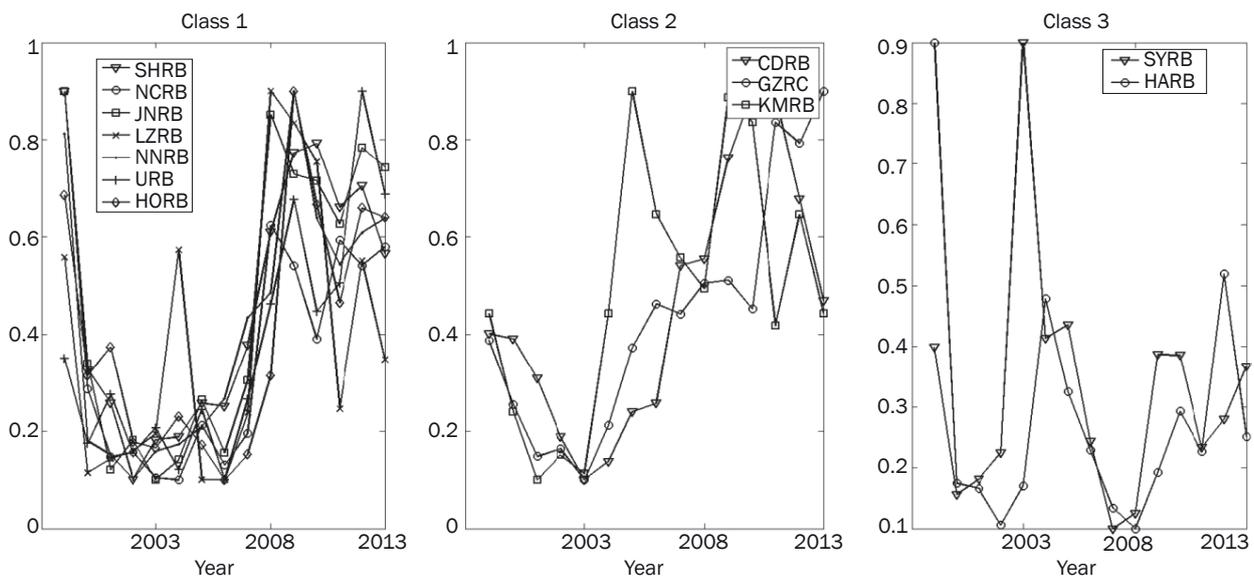


Figure 5 – Clustering results of traffic accidents of 12 railway bureaus

improvements to the infrastructure and stronger safety inspection. After that, the running speed was increased several times (the third time in October 2000, the fourth time in October 2001, and the fifth time in April 2004), but all railway bureaus carried out a series of safety inspection activities, and the number of accidents was reduced effectively. In 2008, a large area in China was affected by freezing rain and snow disaster, which posed a serious threat to the safety of railway operation. In September 2008, the financial crisis caused freight traffic decline in China, until 2009, then freight volumes continued to rise. The increase of passenger volumes and freight volumes led to a railway traffic increase, which had a certain influence on the increase of accidents.

Finally, SYRB, GZRC, and NCRB have been selected randomly from these 3 classes as experimental examples. If the predictive accuracies of the three railway bureaus are high, to a certain extent, the model proposed in this paper is suitable for prediction of traffic accidents in other railway bureaus.

4.2 Selection of related indexes

For the correlation analysis in this paper, TGRA is applied. Compared with Deng's gray relational degree [24], triangular gray relational degree is not only easy to apply but also has a better division to multiple time series.

Based on the gray relation theory, the closer to 1 the triangular relational degree is, the higher is the similarity degree of two sequences; the closer to 0 the triangular relational degree is, the lower the similarity degree of two sequences is.

The indexes affecting railway traffic accidents include: number of passengers dispatched (NPD), passenger turnover volume (PTV), tonnage of freight dispatched (TFD), freight turnover volume (FTV), average daily output of freight locomotive (OFL), average daily number of car loadings (NCL), and operating mileage (OM). The calculation formula of average daily output of freight locomotive is as follows:

$$O = \frac{\sum_{i=1}^I W_i \cdot L_i}{N \cdot T} \tag{11}$$

where O represents average daily output of freight locomotive; W_i and L_i represent the weight and transportation distance of the i -th freight, respectively; N represents the number of freight locomotives; and T represents the number of days.

The average daily number of car loadings refers to the sum of the average daily number of car loadings and the average daily number of car loadings and un-loadings.

The relationship among these 7 indexes is shown in Figure 6. Railway traffic accidents are mainly affected by three indexes, namely, freight turnover volume, passenger turnover volume, and operating mileage. With the continuous increase of passenger and freight traffic volumes and operation mileage growth, the number of locomotives running has been increasing, which directly increases the probability of accidents. In addition, the increase in operating mileage promotes the increase of passenger and freight traffic volumes. In addition, average daily output of freight locomotive determines the capacity of freight transportation.

The correlation degrees between the 7 indexes and traffic accidents of the 3 railway bureaus are shown in Tables 2-4. From the tables, we can see that the

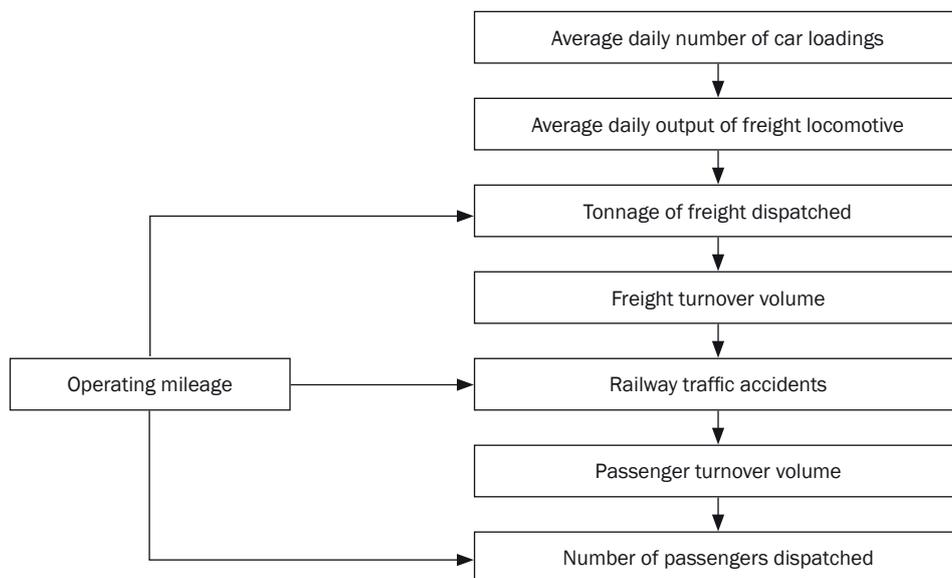


Figure 6 – The relationship among 7 indexes

Table 2 – The correlation degrees between accidents of SYRB and indexes

	SYRB	NPD	PTV	TFD	FTV	OFL	NCL	OM
SYRB	1.00	0.70	0.72	0.72	0.74	0.72	0.72	0.77
NPD	0.70	1.00	0.89	0.86	0.85	0.85	0.86	0.78
VTP	0.72	0.89	1.00	0.98	0.98	0.97	0.98	0.92
TFD	0.72	0.86	0.98	1.00	0.99	0.99	1.00	0.93
VTF	0.74	0.85	0.98	0.99	1.00	0.99	0.99	0.96
OFL	0.72	0.85	0.97	0.99	0.99	1.00	0.99	0.94
NCL	0.72	0.86	0.98	1.00	0.99	0.99	1.00	0.94
OM	0.77	0.78	0.92	0.93	0.96	0.94	0.94	1.00

Table 3 – The correlation degrees between accidents of GZRB and indexes

	GZRB	NPD	PTV	TFD	FTV	OFL	NCL	OM
GZRB	1.00	0.92	0.93	0.88	0.87	0.87	0.89	0.90
NPD	0.92	1.00	0.97	0.95	0.86	0.88	0.96	0.92
VTP	0.93	0.97	1.00	0.96	0.95	0.95	0.97	0.89
TFD	0.88	0.95	0.96	1.00	0.91	0.90	0.98	0.89
VTF	0.87	0.86	0.95	0.91	1.00	0.97	0.90	0.80
OFL	0.87	0.88	0.95	0.90	0.97	1.00	0.89	0.82
NCL	0.89	0.96	0.97	0.98	0.90	0.89	1.00	0.90
OM	0.90	0.92	0.89	0.89	0.80	0.82	0.90	1.00

Table 4 – The correlation degrees between accidents of NCRB and indexes

	NCRB	NPD	PTV	TFD	FTV	OFL	NCL	OM
NCRB	1.00	0.85	0.78	0.70	0.71	0.73	0.65	0.72
NPD	0.85	1.00	0.94	0.86	0.86	0.83	0.75	0.88
VTP	0.78	0.94	1.00	0.96	0.96	0.88	0.87	0.92
TFD	0.70	0.86	0.96	1.00	0.99	0.85	0.93	0.95
VTF	0.71	0.86	0.96	0.99	1.00	0.88	0.92	0.94
OFL	0.73	0.83	0.88	0.85	0.88	1.00	0.81	0.81
NCL	0.65	0.75	0.87	0.93	0.92	0.81	1.00	0.86
OM	0.72	0.88	0.92	0.95	0.94	0.81	0.86	1.00

correlation degrees between the 7 indexes and accidents of SYRB are higher than 0.7; the correlation degrees between the 7 indexes and accidents of GZRC are higher than 0.86; and the correlation degrees between the 7 indexes and accidents of NCRB are higher than 0.65. In addition, there are high correlations between indexes. The correlations between the 7 indexes of SYRB are above 0.78, GZRB above 0.8, and NCRB above 0.75. Although there are high correlations between indexes, the indexes are not exactly the same. It is the difference between indexes that has a different effect on accidents. If only some of the indexes are used for prediction, the accuracy of prediction is likely to be affected. Moreover, these 7 indexes reflect the different aspects of railway transportation. Therefore, all of them are used to forecast railway traffic accidents.

4.3 Test of the proposed IPSO

In order to test the effectiveness of the proposed algorithm, 4 test functions are used to compare convergence and global search capability. The global optimums of these 4 functions are achieved when their variables equal 0, and all global optimums equal 0. Here are the 4 test functions [27]:

$$F0: f(\alpha) = \sum_{i=1}^n \alpha_i^2, -100 < \alpha_i < 100 \tag{12}$$

$$F1: f(\alpha) = \sum_{i=1}^n (100(\alpha_{i+1} - \alpha_i^2)^2 + (\alpha_i - 1)^2), -30 < \alpha_i < 30 \tag{13}$$

$$F2: f(\alpha) = \sum_{i=1}^n (\alpha_i^2 - 10 \cos(2\pi\alpha_i) + 10), -5.12 < \alpha_i < 5.12 \tag{14}$$

$$F3: f(\alpha) = \frac{1}{4000} \sum_{i=1}^n \alpha_i^2 - \prod \cos(2\pi\alpha_i) + 1, \quad (15)$$

$-60 < \alpha_i < 60$

In addition, Shi's linear decrease weight PSO (LDWPSO) [28], Zhang's adaptation weight PSO (APSO) [29], standard PSO, and, as proposed in this paper, IPSO have been tested and compared using the 4 types of test functions. The inertia weight changes for the four types of PSO are shown in Figure 7.

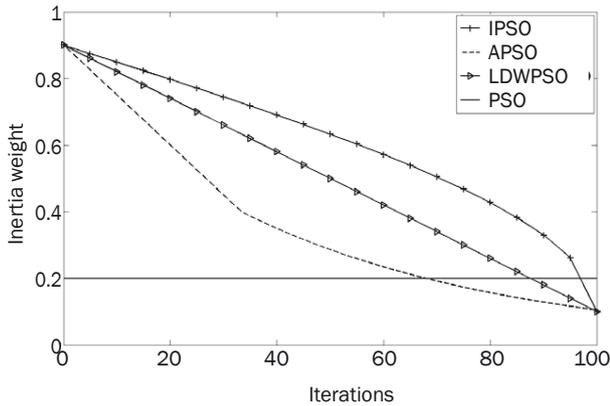


Figure 7 – Inertia weight changes for the 4 PSO types

Table 5 – Parameters of 4 PSO types

	Abbreviation	Value			
		PSO	LDWPSO	APSO	IPSO
Number of particle swarms	-	10	10	10	10
Particle dimension	-	10	10	10	10
Mutation rate	p	-	-	-	0.02
Number of iterations	t_{max}	100	100	100	100
Velocity coefficient	c_1, c_2	2	2	2	2
Inertia weight	w	0.2	-	-	-
Maximum inertia weight	w_{max}	-	0.9	-	0.9
Minimum inertia weight	w_{min}	-	0.1	-	0.1
Adaptation inertia weight	w_0	-	-	0.9	-
Adaptation inertia weight	w_1	-	-	0.5	-
Adaptation coefficient	k	-	-	50	-
Maximum velocity	V_{max}	1	1	1	1

Table 6 – Test results of the 4 PSO types

Function	PSO	LDWPSO	APSO	IPSO
$F0$	889.11	561.18	1,777.5	0.8949
$F1$	336.79	243.11	1,736	0.8931
$F2$	24.38	17.96	28.07	0.0114
$F3$	1.95	2.20	2.02	0.028

Parameters of the 4 PSO types are shown in Table 5. The number of particle swarms, particle dimensions, number of iterations, velocity coefficients, and maximum velocity for all PSO types are the same. What makes the IPSO different from the other PSOs is that it uses nonlinear inertia weights and mutation. Its maximum and minimum inertia weights are 0.9 and 0.1, respectively, which are the same as for LDWPSO, and the mutation rate is 0.02. The inertia weight of the standard PSO is 0.2. The adaptation inertia weights of APSO are 0.9 and 0.5, and its adaptation coefficient is 50, as determined according to [29].

The test results of the 4 PSO types are shown in Table 6. The lower the value in the table is, the better the global searching capability of the algorithm. The proposed IPSO has the lowest test results, i.e., 0.8949, 0.8931, 0.0114, and 0.028, which means its global searching ability is the best compared with APSO, LDWPSO, and PSO. Besides, the $F3$ test results of LDWPSO and APSO are not better than PSO, which indicates that the universality of these two algorithms is not sufficient, and IPSO has higher applicability.

In addition, Figure 8 shows the global optimum trend of the 4 PSOs over iterations. The IPSO has the fastest convergence speed.

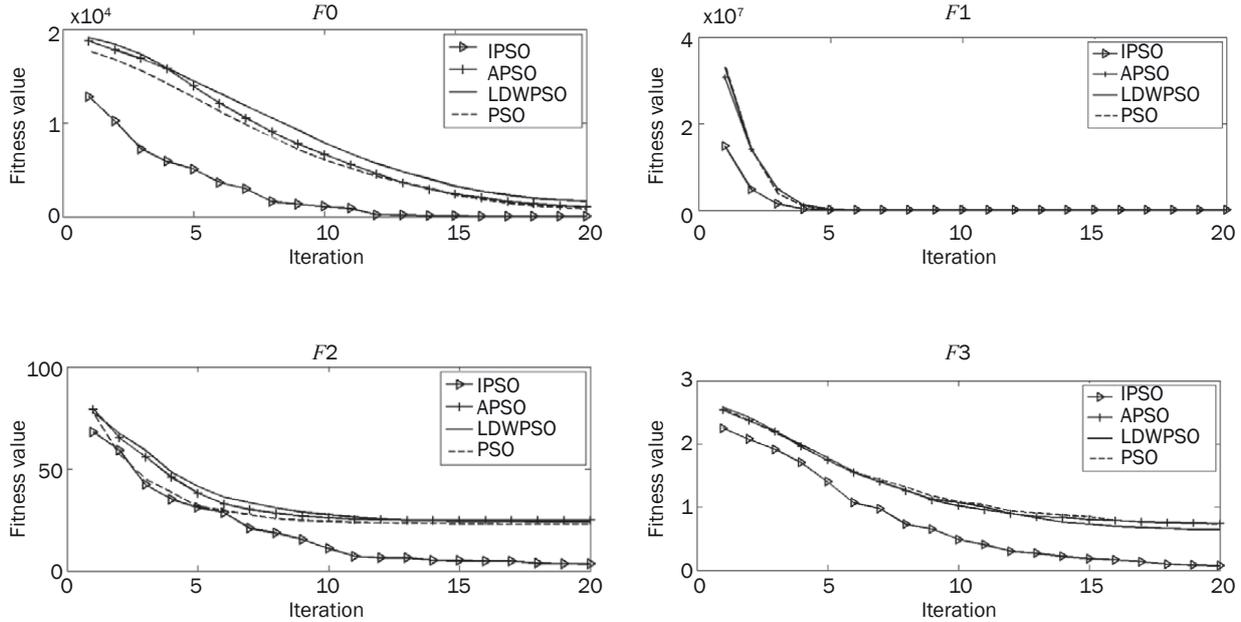


Figure 8 – The global optimum trend for the 4 PSO types over iterations

4.4 Forecast results and discussion

4.4.1 Prediction model parameter setting

All experiments in this paper were operated by MATLAB R2014a. In order to verify the predictive accuracy of the IPSO-DAE proposed in this paper, BPNN, Elman neural network (ELM), and radial basis function neural network (RBF) were used for comparing. The iteration times of BPNN are 1000, learning rate is 0.01, convergence error is 0.0001. The parameters of ELM are the same as BPNN. The convergence goal of RBF is 0. The parameters of IPSO-DAE are shown in Table 7, and some of the IPSO parameters are the same as those in Table 5.

4.4.2 Traffic accident prediction for SYRB

The predictive results for SYRB are shown in Table 8. The structure in the table represents the structure of the neural network. For example, 7-56-39-61-1 represents a neural network with 5 layers, and the number of neurons in the input layer is 7. The number of neurons in the first hidden layer 56, in the second hidden layer 39, in the third hidden layer is 61, and in the output layer it is 1. It should be noted that the experimental process of BPNN and ELM is intended to test the minimum validation error of the number of neurons in the range of 1 to 100, and then use the model with the minimum validation error to predict. The experimental process of RBF is intended to test the minimum validation error of the spread coefficient in the range of 0.01 to 2, and then use the model with the minimum validation error to predict. When the validation error of RBF is minimum, the RBF structure is 7-10-1, and the spread coefficient is 0.06. IPSO-DAE3

Table 7 – IPSO-DAE parameters

	Abbreviation	Value
The number of hidden layers	j	1,2,3,4
The neurons of hidden layers	$n_j(1 \leq n_j \leq 100)$	Given by IPSO
Learning rate of pre-training	ϵ_j^P	Given by IPSO
Iteration times of pre-training	-	100
Learning rate of fine-tuning	ϵ^F	Given by IPSO
Iteration times of fine-tuning	-	1,500
Convergence parameter of DAE	-	0.0001
Convergence value of IPSO fitness	-	5
Population size of IPSO	S	5
Iteration times of IPSO	I	30

Table 8 – Accident prediction results for SYRB

Model	Structure	MAE	MAPE [%]
IPSO-DAE3	7-61-1	16.23	8.10
IPSO-DAE4	7-27-11-1	16.81	7.85
IPSO-DAE5	7-56-39-61-1	9.49	4.08
IPSO-DAE6	7-66-36-86-18-1	16.66	7.35
BPNN	7-70-1	65.98	28.57
ELM	7-5-1	53.39	23.09
RBF	7-10-1	58.40	23.86

means that DAE has 3 layers, which includes a hidden layer. IPSO-DAE6 signifies that DAE has 6 layers. As can be seen from Table 8: (i) Compared with other models, the predictive accuracy of IPSO-DAE5 is the highest (MAE is 9.49, MAPE is 4.08%), which is suitable for SYRB traffic accident prediction. (ii) The prediction performance of IPSO-DAE3 is much better than that of BPNN, which shows that the weights and bias of DAE have been optimized by pre-training. (iii) The performances of all types of IPSO-DAE are better than those of shallow models, including BPNN, ELM, and RBF, which shows that the deep learning model is more suitable for predicting railway traffic accidents that are sudden and random. (iv) The predictive performance of IPSO-DAE6 is not optimal, which indicates that the increase of hidden layers does not mean that the predictive performance will be better. The reason for this phenomenon may be overfitting.

Figure 9 shows the parameter trends of IPSO-DAE5's optimal particle and fitness over iterations. Figures 9a-9c represent the trend of the number of neurons of the first, the second and the third hidden layers, respectively. After several iterations, they are eventually fixed at 56, 39, and 61, respectively. Figures 9d-9f represent the change of learning rates of the first, the second, and the third hidden layers during pre-training, and they are eventually fixed at 0.2062, 0.1045, and 0.1488. Figure 9g shows the change of learning rate of back-propagation algorithm for fine-tuning, which is

finally fixed at 0.1949. Figure 9h shows the change of fitness function value of IPSO-DAE5, and eventually it reaches 12.13% after many iterations. In conclusion, combining DAE with IPSO makes the DAE parameters optimize, and the predictive performance of DAE improves.

Figure 10 shows the change of weights and bias of IPSO-DAE5. Figures 10(a1)-10(a4) denote the weights and bias of the first, the second, and the third hidden layers as well as the output layer before pre-training. Figures 10(b1)-10(b4) represent their values after pre-training. Figures 10(c1)-(c4) show their values after fine-tuning. It can be seen that the weights and bias of the hidden layers changed greatly after pre-training, and the output layer did not change. Pre-training mainly optimizes the weights and bias of the hidden layers. After fine-tuning, all weights and bias change a little, which shows they are optimized by pre-training, and the auto-encoder solves the gradient vanishing of the back-propagation algorithm when there are many hidden layers.

4.4.3 Traffic accident prediction for GZRC

The predictive results for GZRC are shown in Table 9. When the validation error of RBF is minimum, the structure of RBF is 7-10-1, and the spread coefficient is 0.16. As can be seen: (i) Compared with other models, the predictive accuracy of IPSO-DAE4 is the highest (MAE is 8.52, MAPE is 5.33%), which is suitable for GZRC traffic accident prediction. (ii) The pre-

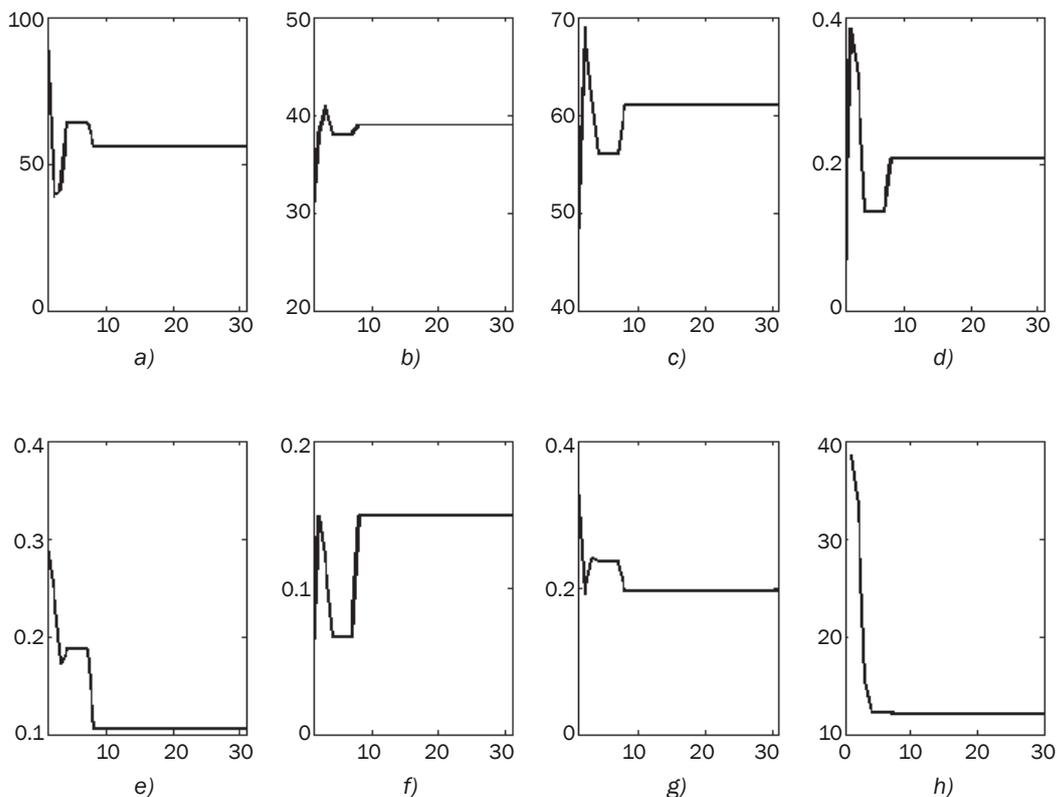


Figure 9 – Parameter trends of IPSO-DAE5's optimal particle and fitness over iterations

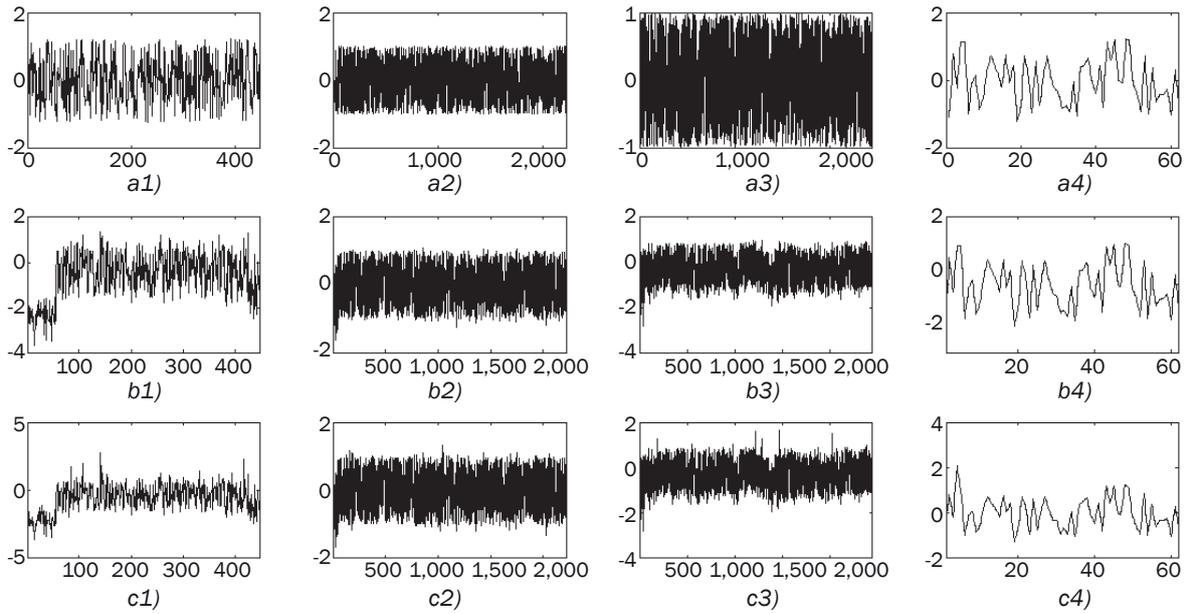


Figure 10 – Parameter trends of IPSO-DAE5

diction performance of IPSO-DAE3 is much better than that of BPNN, which shows that the weights and bias of DAE are optimized by pre-training. (iii) The prediction performance of IPSO-DAE6 is not optimal, which indicates that the increase of hidden layers does not mean that the predictive performance will be better.

Figure 11 shows the parameter trends for IPSO-DAE4's optimal particle and fitness over iterations. Figures 11a and 11b represent the number of neurons of the first and the second hidden layers, respectively. After several iterations, they are eventually fixed at 80 and 23, respectively. Figures 11c and 11d represent the change of learning rates of the first and the second

Table 9 – Accident prediction results for GZRC

Model	Structure	MAE	MAPE [%]
IPSO-DAE3	7-54-1	25.99	16.12
IPSO-DAE4	7-80-23-1	8.52	5.33
IPSO-DAE5	7-47-39-42-1	15.47	9.96
IPSO-DAE6	7-8-95-50-63-1	16.87	10.45
BPNN	7-10-1	26.84	16.59
ELM	7-20-1	11.46	7.00
RBF	7-10-1	62.83	39.34

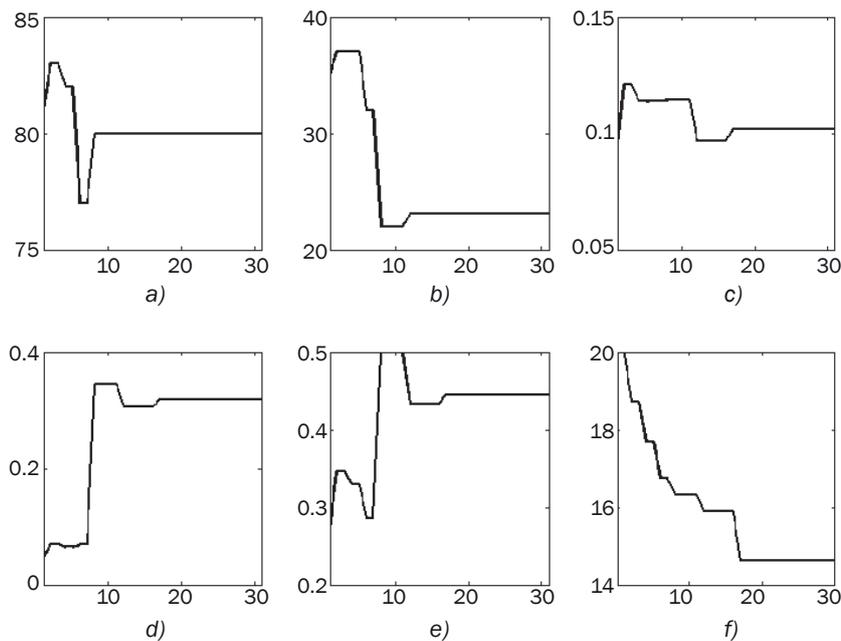


Figure 11 – Parameter trends of IPSO-DAE4's optimal particle and fitness over iterations

hidden layers during pre-training, and they are eventually fixed at 0.1019 and 0.3194. Figure 11e shows the change of learning rate of the back-propagation algorithm for fine-tuning, which is finally fixed at 0.4457. Figure 11f shows the change of the fitness function value of IPSO-DAE4, and eventually it reaches 14.6% after many iterations.

Figure 12 shows the changes of weights and bias of IPSO-DAE4. Figures 12(a1)-(a3) denote the weights and bias of the first, the second hidden layer, and the output layer before pre-training. Figures 12(b1)-(b3) represent their values after pre-training. Figures 12(c1)-(c3) show their values after fine-tuning. The change of weights and bias is the same as in Figure 10, so it is not necessary to describe the phenomenon and analysis in detail.

4.4.4 Traffic accidents predicting of NCRB

The predictive results of NCRB are shown in Table 10. When the validation error of RBF is minimum, the structure of RBF is 7-10-1 and spread coefficient is 0.02. As can be seen: (i) Compared with other mod-

els, the prediction accuracy of IPSO-DAE3 is the highest (MAE is 9.63, MAPE is 7.47%), which is suitable for NCRB traffic accident prediction. (ii) The predictive performance of IPSO-DAE3 is much better than that of BPNN, which shows that the weights and bias of DAE are optimized by pre-training. (iii) The prediction performance of IPSO-DAE6 is not optimal, which indicates that the increase of hidden layers does not mean that the predictive performance is better.

Figure 13 shows the parameter trends for IPSO-DAE3's optimal particle and fitness over iterations. Figure 13a represents the number of neurons of the hidden layer. After several iterations, it is eventually fixed at 52. Figure 13b represents the change of learning rates of the hidden layer during pre-training, and it is eventually fixed at 0.4357. Figure 13c shows the change of learning rate of the back-propagation algorithm for fine-tuning, which is finally fixed at 0.4603. Figure 13d shows the change of the fitness function value of IPSO-DAE3, and eventually it reaches 28.35% after many iterations.

Figure 14 shows the change of weights and bias of IPSO-DAE3. Figures 14(a1) and 14(a2) denote the weights and bias of the hidden layer and the output

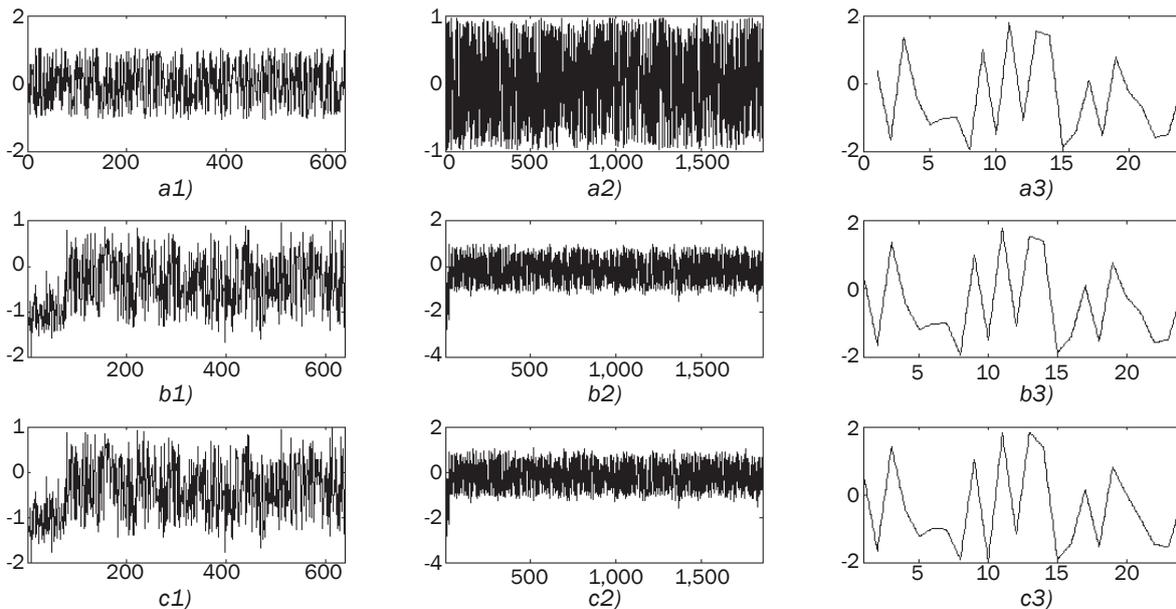


Figure 12 – Parameter trends of IPSO-DAE4

Table 10 – Accident prediction results for NCRB

Model	Structure	MAE	MAPE [%]
IPSO-DAE3	7-52-1	9.63	7.47
IPSO-DAE4	7-65-29-1	11.15	8.36
IPSO-DAE5	7-61-64-63-1	13.81	10.62
IPSO-DAE6	7-62-21-29-46-1	16.66	13.09
BPNN	7-15-1	20.34	15.40
ELM	7-95-1	68.81	52.98
RBF	7-10-1	14.50	11.32

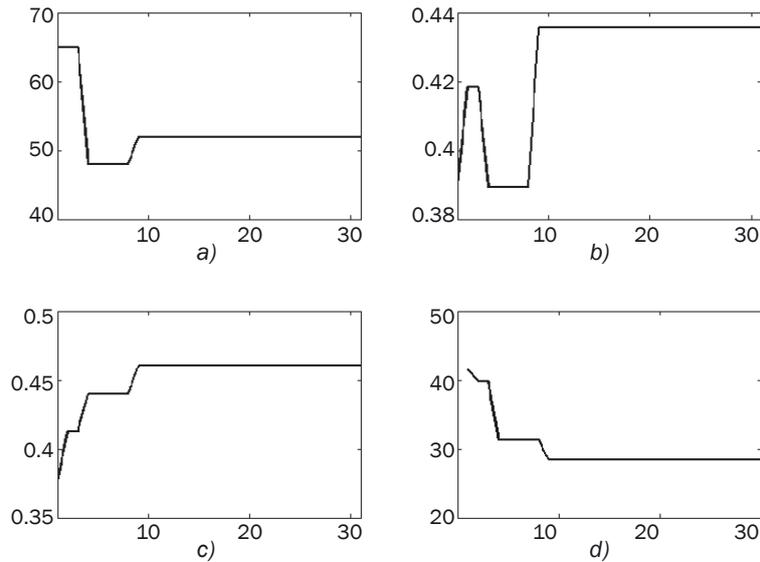


Figure 13 – Parameter trends of IPSO-DAE3's optimal particle and fitness over iterations

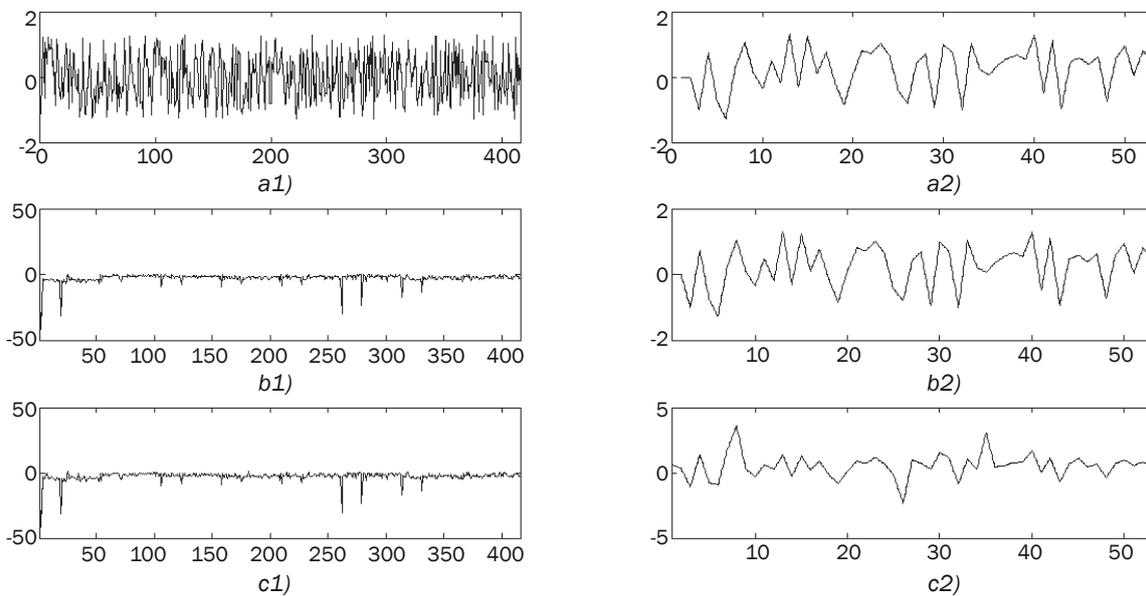


Figure 14 – Parameter trends of IPSO-DAE3

layer before pre-training. Figures 14(b1) and 14(b2) represent their values after pre-training. Figures 14(c1) and 14(c2) show their values after fine-tuning. The change of weights and bias is the same as in Figure 10, therefore it is not necessary to describe the phenomenon and analysis in detail.

The above three experiments show that predicting different classes of rail traffic accidents requires different numbers of layers of the DAE, i.e., class 1 traffic accident forecast uses a 3-layer DAE, class 2 uses a 4-layer DAE, and class 3 uses a 5-layer DAE.

5. CONCLUSIONS

Railway traffic accident prediction is one of the research hotspots in the field of accident prediction. Studies about accident prediction mainly focus on

analyzing relationships between indexes and accidents. It is not sufficient for accident prediction to only analyze these relationships. This paper attempts to reveal these relationships and then predict railway traffic accidents.

In the past, the structure of DAE was obtained by experience or multiple tests, which is time-consuming and laborious. Given this, IPSO is proposed for finding a better DAE structure, including the number of hidden layers and neurons at each hidden layer, the learning rate of each hidden layer when reconstructing input data during pre-training, and the learning rate of the back-propagation algorithm during fine-tuning. There are several main findings after the experiments, as follows: (i) The proposed IPSO has a better global searching ability and higher convergence speed. (ii) The opti-

mized DAE, one of the deep learning models, is more suitable for predicting railway traffic accidents that are sudden and random. (iii) Predicting different classes of rail traffic accidents requires different numbers of DAE layers. (iv) IPSO can optimize the DAE parameters and then improve the DAE's predictive accuracy. (v) The increase of hidden layers in DAE does not mean that the predictive performance will be better.

Although this paper only forecasts accidents for three railway bureaus, they represent three classes of accidents, which means that IPSO-DAE can also be used to predict traffic accidents of other railway bureaus. Moreover, changing the flight range of particles of IPSO and the number of DAE's hidden layers enables applying IPSO-DAE to predictive problems in other research fields. It needs to be pointed out that using more relevant indexes and better deep learning models may lead to better predictive results. This can be done in future research.

ACKNOWLEDGEMENT

This research was funded by the China Railway Science and Technology Research Development Program (No. 2015F024).

冯芬玲, 李万, 蒋琦玮

中南大学 交通运输工程学院, 湖南 长沙 410075

基于优化深度自动编码器的铁路行车事故预测

摘要

安全是铁路运输的重点, 铁路行车事故预测是安全管理的主要内容。事故与相关因素之间有复杂的非线性关系, 因此三角灰色关联分析(TGRA)被用来得到与事故相关的因素, 深度自动编码器(DAE)被用来发现事故与因素之间的复杂关系并用来预测事故。此外, 本文提出了一种具有更快收敛性和更好全局搜索能力的非线性权重变化的粒子群优化算法, 用来获得更优的DAE结构和参数, 包括隐含层数、隐含层神经元数和学习率。该模型对沈阳铁路局、广州铁路公司、南昌铁路局的铁路行车事故进行了预测。实验结果表明提出的模型能较好地预测铁路行车事故。

关键词

铁路行车事故; 深度自动编码器; 粒子群优化算法

REFERENCES

- [1] Feng F, Xu Y, Tang Z. Research on the charge rate of railway value-guaranteed transportation based on competitive and cooperative relationships. *Advances in Mechanical Engineering*. 2018;10(1): 1-11.
- [2] Ma X, Li K, Luo Z, Zhou J. Analyzing the causation of a railway accident based on a complex network. *Chinese Physics B*. 2014;23(2): 1674-1056.
- [3] Ross DA, Jodi LC. An alternative accident prediction model for highway-rail interfaces. *Accident Analysis and Prevention*. 2002;34(1): 31-42.
- [4] Oh J, Washington SP, Nam D. Accident prediction model for railway-highway interfaces. *Accident Analysis and Prevention*. 2006;38(2): 346-356.
- [5] Yan X, Stephen R, Su X. Using hierarchical tree-based regression model to predict train-vehicle crashes at passive highway-rail grade crossings. *Accident Analysis and Prevention*. 2010;42(1): 64-74.
- [6] Lu P, Tolliver D. Accident prediction model for public highway-rail grade crossings. *Accident Analysis and Prevention*. 2016;90: 73-81.
- [7] Wen C. Prediction methods of train operation conflict for high-speed railway. *Journal of Transportation Security*. 2010;3(4): 275-286.
- [8] Akram KG B, Farnoosh N, Viliam M. Highway accident modeling and forecasting in winter. *Transportation Research Part A: Policy and Practice*. 2014;59: 384-396.
- [9] Mao M, Chirwa EC. Application of grey model GM(1, 1) to vehicle fatality risk estimation. *Technological Forecasting and Social Change*. 2006;73(5): 588-605.
- [10] Darin A, Blent A. A neural network (NN) model to predict intersection crashes based upon driver, vehicle and roadway surface characteristics. *Scientific Research and Essays*. 2010;5(19): 2837-2847.
- [11] Seung GK, Young GN, Poong HS. Prediction of severe accident occurrence time using support vector machines. *Nuclear Engineering and Technology*. 2015;47(1): 74-84.
- [12] Qu X, Wang W, Wang W, Liu P. Real-time freeway side-swipe crash prediction by support vector machine. *Intelligent Transport Systems*. 2013;7(4): 445-453.
- [13] Partheeban P, Arunbabu E, Hemamalini RR. Road accident cost prediction model using systems dynamics approach. *Transport*. 2008;23(1): 59-66.
- [14] Muhammed YÇ, Ahmet T. An artificial neural network model for highway accident prediction: a case study of Erzurum, Turkey. *Promet - Traffic & Transportation*. 2015;27(3): 217-225.
- [15] Jafari SA, Jahandideh S, Jahandideh M, Asadabadi EB. Prediction of road traffic death rate using neural networks optimised by genetic algorithm. *International Journal of Injury Control and Safety Promotion*. 2015;22(2): 153-157.
- [16] Zhang J, Duan L, Guo J. A genetic algorithm-based BP neural network method for operational performance assessment of ATC sector. *Promet - Traffic & Transportation*. 2016;28(6): 563-574.
- [17] Ding S, Su C, Yu J. An optimizing BP neural network algorithm based on genetic algorithm. *Artificial Intelligence Review*. 2011;36(2): 153-162.
- [18] Hinton GE, Osinder S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Computation*. 2006;18(7): 1527-1554.
- [19] Bo G, Rui Z, Guang X, Chuangming S, Li Y. Predicting students performance in educational data mining. *Proceedings of the 2015 International Symposium on Educational Technology; 2015 Jul 27-29; Wuhan, China: IEEE; 2015*.
- [20] Li N, Lu G, Li X, Yan Y. Prediction of NOx emissions from a biomass fired combustion process based on flame radical imaging and deep learning techniques. *Combustion Science and Technology*. 2016;188(2): 233-246.
- [21] Ong B, Sugiura K, Zettsu K. Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5. *Neural*

- Computing and Applications*. 2016;27(6): 1553-1566.
- [22] Kuremoto T, Kimura S, Kobayashi K, Obayashi M. Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing*. 2014;137: 47-56.
- [23] Shao H, Jiang H, Zhang X, Niu M. Rolling bearing fault diagnosis using an optimization deep belief network. *Measurement Science and Technology*. 2015;26(11): 115002.
- [24] Cheng G, Yin J, Liu N. Estimation of geophysical properties of sandstone reservoir based on hybrid dimensionality reduction with Elman neural networks. *Applied Mechanics and Materials*. 2014;668-669: 1509-1512.
- [25] Bengio Y, Delalleau O. On the expressive power of deep architectures. *Algorithmic Learning Theory*. 2011;6925: 18-36.
- [26] Song C, Huang Y, Liu F, Wang Z, Wang L. Deep auto-encoder based clustering. *Intelligent Data Analysis*. 2014;18(6): S65-S76.
- [27] Suganthan PN, Hansen N, Liang JJ. *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization*. Singapore: Nanyang Technological University. 2005: 1–50. Available from: <http://decsai.ugr.es/~lozano/AEBs-Continuo/Tech-Report-May-30-05.pdf>
- [28] Shi YH, Eberhart R. A modified particle swarm optimizer. *Proceedings of the IEEE International Conference on Evolutionary Computation; 1998 May 4-9; Anchorage, Alaska, USA*: IEEE; 1999.
- [29] Zhang J, Zhang J, Lok T, Michael RL. A hybrid particle swarm optimization-backpropagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*. 2007;185: 1026-1037.