Received October 22, 2018, accepted November 14, 2018, date of publication November 19, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881890

A Fault Prediction Algorithm Based on Rough Sets and Back Propagation Neural Network for Vehicular Networks

RONG GENG[1], (Member, IEEE), XIAOJIE WANG[2], NING YE[1], AND JUN LIU[1]
[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[2]School of Software, Dalian University of Technology, Dalian 116620, China

Corresponding author: Jun Liu (liujun@cse.neu.edu.cn)

This work was supported in part by the National Natural Science Funds of China under Grants 61701100, 61501038, 61671141, 61501105, 61871107, and 201706085051, and in part by the Fundamental Research Funds for the Central Universities under Grant N161613001.

ABSTRACT Vehicular networks (VNs) have become a feasible solution to solve network related problems in an intelligent transportation system. Due to their wide use in services, VNs are extremely vulnerable to interference that leads to frequent faults; therefore, their security challenges are widely recognized and urgently needed. Before a fault occurs, there are often some characteristic signals or data development trends that indicate a fault. If these data can be collected and judged effectively, fault prediction can be achieved. Therefore, it is of great significance to guarantee network security, reliability and continuous operation. In this paper, we propose a new fault prediction method for VNs. The fault prediction algorithm that uses rough sets and backpropagation (BP) neural network for VNs is divided into three parts, namely, the data acquisition module, the data prediction module, and the fault prediction module. The data acquisition module collects the data from the network to build the decision table using the rough sets and establishes the discernible matrix decision tables to reduce the data. The data prediction module combines the advantages of gray theory and BP neural network into a gray BP neural network to predict the data. The fault prediction module uses the normal data and fault data as input data and uses the fault type as the output to train the error BP neural network to obtain the appropriate weights, and then the predicted data is entered into the trained BP neural network to realize fault prediction. The fault prediction algorithm is simulated and analyzed using NS-2 and MATLAB, respectively, and the results show that the proposed algorithm can accurately diagnose and predict faults using the predicted data.

INDEX TERMS Vehicular networks, fault prediction, rough sets, BP neural network

I. INTRODUCTION

Vehicular networks (VNs) are becoming one of the most promising research topics since they provide comfort and safety information to drivers and passengers [1]–[5]. In VNs, networks can be formed and information can be relayed among vehicles, where vehicles are connected to each other and can connect to the Internet. VNs are one of the most valuable types of MANETs, which spontaneously create wireless networks for data exchange [6]. In Vehicular Networks, nodes move freely, which means that there are no constraints on their movement. Nodes communicate with each other via a single hop or multiple hops. VNs that transmit enormous amounts of data are vulnerable to being attacked, which leads to a decline in the network performance in the operating environment [7]–[10].

With secure VNs, valuable information cannot be modified or deleted by the attacker. However, security in VNs also demonstrates the ability to determine driver responsibility while maintaining driver privacy. The information between the vehicles must be exchanged securely and, more importantly, timely. The delay of message exchanges can lead to disastrous consequences such as vehicle collisions. Therefore, security in VNs is a major concern because human lives are always at risk. Conversely, in traditional networks, the major security concerns include confidentiality, integrity, and availability, all of which do not involve the potential loss of life. However, the deployment of a comprehensive security system for VNs is very challenging in practice. The security vulnerabilities of VNs are often critical and dangerous. In addition, the nature of the vehicular networks is highly dynamic, with frequent and instantaneous arrivals and departures of vehicles and short connection durations. In addition to its dynamic characteristics and high mobility, the use of wireless media also makes VNs vulnerable to attacks due to the openness and broadcasting nature of wireless communications [11]–[14].

Because the networks gradually become heterogeneous, dynamic and intelligent, those changes make it increasingly more difficult to predict faults [15]. Therefore, fault prediction in VNs becomes a hot topic. If we can predict the faults accurately and rapidly, it has great significance for network security.

Aiming at the problems mentioned above, we study the existing self-organizing networks, sensor networks, optical networks and so on and propose a novel fault prediction algorithm based on rough sets and BP neural network for VNs in order to handle the transmission delays, the changeable network topology wireless communications, etc Our main contributions of this paper are demonstrated as follows:

- We propose a fault prediction algorithm based on rough sets and BP neural network for VNs aiming at predicting the faults. We divide the algorithm into three modules: the data acquisition module, the data prediction module and the fault prediction module.
- We use the data acquisition module to collect data from the network. The data prediction module combines the advantages of gray theory and BP neural network into a gray BP neural network to predict the data. We use the fault prediction module to obtain the appropriate weights.
- We conduct evaluations using NS-2 and MATLAB, Results show that the proposed algorithm can accurately diagnose and predict faults using the predicted data.

The rest of this paper is organized as follows. In Section II, we illustrate the related works of the existing security policies including fault prediction. In Section III, the fault prediction algorithm is proposed for VNs, which consists of a data acquisition module based on rough sets theory, a data forecasting module based on the gray model, and a fault prediction module based on BP neural networks. Section IV demonstrates the simulation results, and our work is concluded in Section V

II. RELATED WORK

During the development of fault prediction, the expert knowledge system is the first one to be applied to fault diagnosis. However, the expert system has difficulties in knowledge acquisition, knowledge maintenance, narrow knowledge applications and an inappropriate reasoning ability. This technology cannot be accurately applied in the case of new faults or current information. For the fuzzy fault features, it is difficult to obtain accurate diagnosis results, and the system will have matching conflicts The Bayesian network [16] is an effective derivation theory for an artificial intelligence system, and it is especially suitable for probability and uncertainty. In the face of large networks, how to learn the conditional probability of each point is also a difficult problem. The precise inferences of the Bayesian network and some approximate methods have been proved to be an NP (non-deterministic polynomial) problem. If you use the network, you must undertake much careful analysis and preparation. Fault diagnosis based on graph theory [17] uses the graph theory model for the network modeling and studies fault localization on this basis. Initially, a fault diagnosis method for complex multiprocessor systems, the system-level diagnosis theory [18], [19] corresponds to a circuit level diagnosis. The basic principle is to set up test tasks for each processor node in the system according to a certain strategy, test the nodes, and analyze the failures of the nodes with the obtained test structure.

The idea of rough sets theory [20] is that "knowledge embodies an ability to classify objects." It can solve problems such as the discovery of the relevance of knowledge, the evaluation of data, the acquisition of decision-making algorithms and the reduction of knowledge in information systems. The idea of an attribute reduction algorithm based on the Skowron discernibility matrix [21] is to give a corresponding discernibility matrix for each given information system, and the specific knowledge in the information system is represented by discernibility matrixes. The main idea of the attribute reduction algorithm based on multiforked trees is to use the "tree" learned in the data structure, which is composed of a set and a specific relation defined on that set [22]. Time series prediction [23] refers to finding the change characteristics, developmental process and laws that can predict future developments from the previous time series data. The hidden Markov model [24], [25] is a kind of Markov chain, but its state is uncertain or invisible. It is manifested through the random process of a sequence of observations. The observed events do not correspond to the state but are connected through a set of probability distributions.

Currently, some solutions have been proposed for fault diagnosis in mobile ad hoc networks and wireless sensor networks, but most of the solutions are for a single diagnosis of nodes or links and have not achieved consistent results. These schemes are not fully applicable to VNs. To predict faults in VNs, a fault prediction algorithm must be proposed according to the characteristics of VNs to ensure the secure and reliable operations of the network.

III. FAULT PREDICTION ALGORITHM BASED ON ROUGH SETS AND BP NEURAL NETWORK FOR VNs

The fault prediction algorithm based on rough sets and BP neural network for VNs is divided into three parts, namely, the data acquisition module, the data prediction module and the fault prediction module.

As shown in Fig. 1, the data acquisition module collects data from the network to build decision tables using rough sets and establishes the discernible matrix decision tables to reduce the data. The data prediction module combines the advantages of gray theory and BP neural network into a gray BP neural network to predict the data. The fault prediction module uses the normal data and fault data as input data

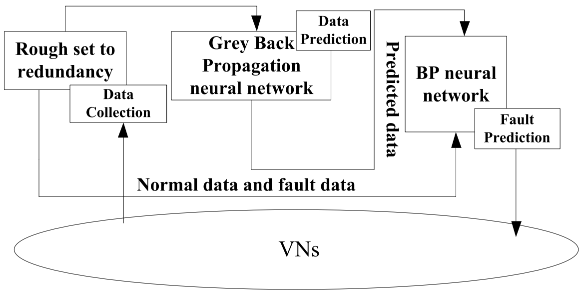


FIGURE 1. Fault prediction algorithm based on rough sets and BP neural network for VNs.

and uses the fault type as the output to train the BP neural network to obtain the appropriate weights. Then, it enters the predicted data into the trained BP neural network to realize the prediction of faults.

A. DATA ACQUISITION BASED ON ROUGH SETS THEORY

The data acquisition module collects data from the network, and the data are then divided into normal data and fault data. The data will be chosen to represent the fault types of the network. However, since general data is redundant, the rough sets are used to construct the decision table and the resolution matrix reduces the data. The relatively accurate data will be obtained through the calculation of the importance degree of each attribute to rule out the objective factors.

1) DECISION TABLE, DISCERNIBILITY MATRIX AND ATTRIBUTE SIGNIFICANCE OBJECTIVE

Definition 1: $S = (U, R)$ is the expression of the knowledge systems, and R can be divided into several conditional attribute sets C The decision attribute set is D, and $C \cup D = R, C \cap D = \emptyset$ If there is a decision attribute in the knowledge representation system it can be taken from the decision table, and the decision table is recorded as $T = (U, R, C, D)$ C and D denote the components of the decision table and $Ind(C)$, $Ind(D)$ are called the classes and decisions, respectively. In the decision table, the data in the system can be assigned to different decision attributes and decision tables in order to express the attributes.

In the knowledge representation system, the contributions of each attribute are not equal In the rough sets, data can be obtained directly from the importance of the decision table's objects. The method of determining the significance of attributes is as follows If there is one or some attributes that are changed in the decision table, it means that this property is more important, and vice versa. $r_c(D)$ can be used to describe the significance of the attributes. $r_c(D)$ is as follows:

$$r_c(D) = Card(pos_c(D))/Card(U) \tag{1}$$

The classification of the importance $i \subseteq C$ is inferred by the Attributes D and can be described by the difference in its dependence It is as follows:

$$r_c(D) - r_{c-i}(D) \tag{2}$$

U/D shows the influence of the system after deleting property i from the collection C, and the objective importance of the property is as follows:

$$w_i = \frac{r_c(D) - r_{c-i}(D)}{\sum_{i=1}^{n} [r_c(D) - r_{c-i}(D)]} \tag{3}$$

Thus, the obtained data weighting vector is as follows: $W = \{w_1, w_2, \ldots w_n\}$

Meanwhile, we compress the information related to the attributes to transform the decision table into a matrix by using the resolution matrix rough sets It may reduce the attributes in the decision table.

In the decision table $T = (U, R, C, D)$, for the field $U = \{x_1, x_2, \ldots, x_n\}$ where $R = C \cup D$ the condition Attributes set is $C = \{a, i = 1, 2, \ldots, m\}D = \{d_j, j = 1, 2, \ldots, l\}$ is the decision attribute set, and $a_k(x_j)$ represents the value of x_j that comes from the samples for the property a_k. Here, the difference matrix is defined as being an $n \times n$ matrix i is the first row in the n-th column, and the element is as follows:

$$m_{ij} = \begin{cases} a_k \in C, a_k(x_i) \neq a_k(x_j) \wedge D(x_i) \neq D(x_j) \\ \emptyset D(x_i) = D(x_j) \end{cases} \tag{4}$$

From (4), we can see that the discernibility matrix m_{ij} is a set composed of all the attributes of different objects x_i and x_j. For example, if x_i and x_j exist in the same Decision class, then m_{ij} in the resolution matrix element is the empty set. By definition, $M(T) = [m_{ij}]_{n \times n}$ is a symmetric matrix and its main diagonal elements are empty. Therefore, only the upper or upper triangular part of a matrix should be considered in order to achieve the purpose of simplification.

2) NETWORK FAULT RULES AND OBJECTIVE ATTRIBUTES EVALUATION

The rough sets decision table can represent the different types of network faults, and the discernibility matrix can simplify the decision table. It is more convenient for extracting decision rules. The data reduction can eliminate the unimportant information and keep the valuable information. This can accelerate the speed of fault diagnosis. Furthermore, it can also create a problem in which the reduction of attributes decreases the efficiency of fault diagnosis since some of the more important attributes are missing. Filling in these properties can largely improve the diagnosis effect, but choosing a conditional attribute has certain subjectivity, which is one of its great disadvantages. Therefore, the attributes in the decision table should be evaluated objectively to improve the accuracy of fault diagnosis.

Definition 2: If $R' \subset R$ is a simplified $T_{R'}$, the ability to distinguish between the decision table $T_R = (U_1, R, C, D)$ and $T_{R'} = (U_2, R', C', D')$ is the same After the reduction of T_R into $T_{R'}$, we have $|U_1| \leq |U_2|$

As seen from Definition 2, the length of the decision rules can be reduced, and the number of rules can be accordingly reduced. Therefore, the minimum conditional attribute can be used to determine the fault type, which can increase the practicability of network fault diagnosis. According to the correspondence between the resolution matrix and the decision table, we can obtain the algorithm for $T = (U, R, C, D)$ to reduce $Red(T)$ as follows:

a) Calculate the resolution matrix of $T = (U, R, C, D)$;

b) Select all the values of the non-empty set $m_{ij}(m_{ij} \neq \emptyset)$ in the resolution matrix and construct the corresponding logic expressions to extract $L_{ij}(L_{ij} = \vee m_{ij})$;

c) Make all L_{ij}s interact to obtain the CNF $L = \wedge L_{ij}$;

d) Transform the CNF paradigm into the disjunctive form, by which we can obtain $L' = \wedge L'_{ij}$; and

e) The attribute reduction results show that in the disjunctive the conjunctions of each item should be the same as the attribute reduction results Each one of the attributes included in the conjunctive condition constitutes a set of attributes after the reduction.

When making decisions, the importance of each attribute is different. In the past, many fault diagnosis methods were influenced by human subjective factors, which is a drawback of fault diagnosis. Therefore, the fault accuracy decreases. The rough sets evaluate each attribute, excavates the deep knowledge, and can accurately determine the objective importance of the attribute. The reduction algorithm of the attribute importance is as follows:

a) Obtain the positive region $Pos_c(D)$ of $T = (U, R, C, D)$, where C is called the conditional attribute set, D is called the decision attribute set, and both C and D are disjointed;

b) Use the equation $r_c(D) = Card(Pos_c(D))/Card(U)$ to obtain the dependence of attributes C and D, and calculate the importance of the attributes using $r_c(D) - r_{c-i}(D)$; and

c) According to the equation $w_i = \frac{r_c(D)-r_{c-i}(D)}{\sum_{i=1}^{n}[r_c(D)-r_{c-i}(D)]}$, the attribute importance is normalized to obtain the relative importance of each attribute, which is used for network fault diagnosis.

B. DATA PREDICTION MODULE BASED ON THE GRAY MODEL

The basic principle of the gray model is to accumulate the historical data to generate a new sequence, which is used to establish the gray differential equation. However, the original system will be affected by other factors, and the historical data will appear disordered, which is the gray sequence or gray process.

The data prediction module combines the advantages of gray theory and BP neural network into a gray BP neural network to predict the data.

The gray model is a kind of mathematical model that reflects both the difference and differential properties through the combination of the difference and differential equations. At present, the GM (1, 1) model is the most studied gray prediction model. The modeling process and algorithmic steps are as follows:

Set the original nonnegative sequence as $X^{(0)} = \{x^{(1)}(k), k = 1, 2, \ldots n\}$, and accumulate and convert the sequence individually The new data sequence is as follows:

$$X^{(0)} = \{x^{(1)}(k), x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), k = 1, 2, \ldots n\} \tag{5}$$

Then based on the gray modeling establish the differential equation, which is the following:

$$x^{(0)}(k) + az^{(1)}(k) = u \tag{6}$$

$z^{(1)}(k) = 0.5(x^{(1)}(k) + x^{(1)}(k-1)), k = 2, 3, \ldots n$ where the parameters a and n are respectively the development coefficient and the ash effect amount of the GM (1, 1) model. These correspond to the white differential equation, which is as follows:

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = u \tag{7}$$

The least square method of (7) could determine the parameters a and u, which are calculated as follows:

$$L = \begin{pmatrix} -z^{(1)}(2)\,1 \\ -z^{(1)}(3)\,1 \\ \vdots \\ -z^{(1)}(n)\,1 \end{pmatrix}, \quad Y = \begin{pmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{pmatrix} \tag{8}$$

In the initial condition of $x^{(1)}(1) = x^{(0)}(1)$, we use the first-order linear differential equation solution method to obtain the differential (8) solution, which is as follows:

$$\hat{x}(t)^{(1)} = \left(x^{(0)}(1) - \frac{\hat{a}}{\hat{u}}\right) e^{-a(\hat{t}-1)} + \frac{\hat{a}}{\hat{u}} \tag{9}$$

Typically (9) is called the continuous-time response function in the GM (1, 1) model.

Then, by using these results individually, (4) is restored to obtain the fitting solution of $X^{(0)}$:

$$\hat{x}(t)^{(0)} = \hat{x}(t)^{(1)} - \hat{x}(t-1)^{(1)} = \left(1 - e^{\hat{a}}\right) \left| x^{(0)}(1) - \frac{\hat{a}}{\hat{u}} \right| e^{-a(\hat{t}-1)} \tag{10}$$

Finally, (10) can be dispersed as follows:

$$\hat{x}(k)^{(0)} = \left(1 - e^{\hat{a}}\right) \left(x^{(0)}(1) - \frac{\hat{a}}{\hat{u}}\right) e^{-a(\hat{t}-1)}, \quad k = 2, 3, \ldots \tag{11}$$

Accordingly, (11) is called the discrete-time corresponding sequences of the GM (1, 1) model.

C. FAULT PREDICTION BASED ON BP NEURAL NETWORKS

The three-layer or even multilayer neural network structure using BP is called the BP neural network. The input layer, hidden layer and output layer constitute the structure of the neural network. A hidden layer can be one or more layers. The learning process of the BP network can be divided into

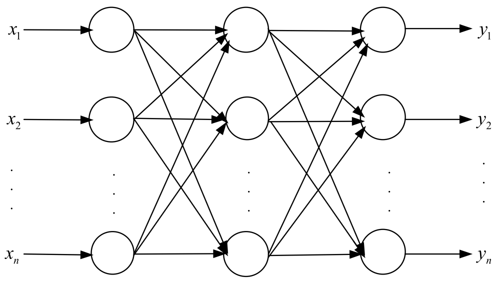


FIGURE 2. Feed forward neural network architecture.

two parts: forward propagation and reverse propagation. In the process of the forward transmission of data, information is propagated from the input layer to the hidden layer and then spread to the output layer after processing. The state of each neuron in each layer only affects the state of the next neuron and has no effect on other neurons. If the output layer does not obtain the expected value after calculating the output, then the process of BP of the error will occur. In the process of back propagation, the connection weights between each pair of neurons are modified layer by layer, and this process repeats continuously until the error reaches the set accuracy.

Fig. 2 shows a basic model of the neurons, which has n inputs, an input layer and a hidden layer, and the connections between the hidden layer and the output layer are represented by the weight w_i. The output function is $y = f(\alpha, \theta)$, where α is the sum of the weighted values, $\alpha = \sum w_i x_i$, and the activation function f has two inputs (α and the threshold θ). In Fig. 2, the right value and the input matrix can be written as follows:

$$\begin{aligned} W &= [w_1, w_2, \ldots w_{n0}] \\ X &= [x_1, x_2, \ldots x_{n0}] \end{aligned} \tag{12}$$

The output neuron equation is as follows:

$$y = f(W \bullet X + \theta) = f(\sum_{i=1}^{n_0} w_i x_i + \theta) \tag{13}$$

In Fig. 2, the threshold as an input of the activation function, namely, the deviation, is crucial to the design of the network. It is the key to controlling the translation of the activation function from left to right, and thus provides the direction for solving this problem. As a result, the threshold value can be added to the right value, which is represented using a fixed constant 1 (if $x_0 = 1$), and then its weight w_0 is the threshold θ.

D. DPREDICTION MODEL BASED ON BP NEURAL NETWORKS

Since the data predicted by the gray prediction model has a certain deviation from the actual value, the neural network and the gray system are similar to each other to some extent. If they are fused, their advantages can be neutralized. First, the neural network can make use of the relationship between the input and output to make the actual output value close to the expected value. According to the definition of gray theory, the output value of the BP neural network is actually a fuzzy value, that is, a gray number. Therefore, we can use the neural network to correct the values from the gray model.

1) GRAY NEURAL NETWORK MODEL

In the gray system, the original sequence of numbers is summed up once, which results in x. The first original series $x^{(0)}(k)$ is summed up once and the series $x^{(0)}(k)$ can be generated, which presents the law of exponential growth. Therefore, the gray differential equation is established as follows:

$$\frac{dx^{(1)}(k)}{dt} + ax^{(1)}(k) = u \tag{14}$$

a, u are the determined parameters. The gray differential (15) can obtain the discrete response function as follows:

$$\hat{x}_1^{(1)}(k+1) = (x_1^{(1)}(0) - \frac{u}{a})e^{-ak} + \frac{u}{a} \tag{15}$$

The parameters of the gray differential (14) for the basic process of whitening are as follows. The time response function (15) of (14) is mapped to a BP network, and then it trains the BP network. When the network converges, the corresponding equation coefficients are extracted from the BP network after training to obtain the white differential equation. Then, the system can be further studied using this white differential equation, and the solution of this differential equation can be obtained.

$$\begin{aligned} \hat{x}_1(k+1) &= ((x_1^{(1)}(0) - \frac{u}{a})\frac{e^{-ak}}{1+e^{-ak}} + \frac{u}{a} \\ &\quad \times \frac{1}{1+e^{-ak}})(1+e^{-ak}) \\ &= ((x_1^{(1)}(0) - \frac{u}{a})(1 - \frac{1}{1+e^{-ak}}) + \frac{u}{a} \\ &\quad \times \frac{1}{1+e^{-ak}})(1+e^{-ak}) \\ &= ((x_1^{(1)}(0) - \frac{u}{a}) - x_1^{(1)}(0) \times \frac{1}{1+e^{-ak}} + 2 \\ &\quad \times \frac{u}{a} \times \frac{1}{1+e^{-ak}})(1+e^{-ak}) \end{aligned} \tag{16}$$

After transformation, (17) can be mapped to the BP network. The BP network weights can be assigned as $\frac{u}{a} = b$, and the threshold of y_1 is set to the following:

$$\begin{cases} w_{11} = a \quad w_{21} = -x_1^{(1)}(0) \\ w_{12} = a \quad w_{22} = 2b \quad w_{13} = 1 + e^{-ak} \end{cases} \tag{17}$$

By (17), the LB layer neuron activation function is taken as the Sigmoid function, and the others are linear functions. After (16)'s assignment and the BP network activation function are determined, each node in the network can be calculated as follows:

$$b_1 = f(ak) = \frac{1}{1+e^{-ak}} = b_2 \tag{18}$$

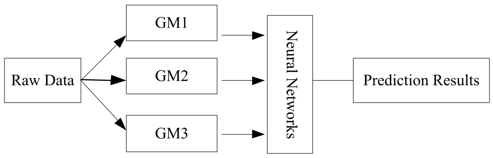


FIGURE 3. Gray series neural network model.

$$\begin{aligned} y_1 &= b_1 w_{21} + b_2 w_{22} + \theta_{y_1} \\ &= -x_1^{(1)}(0) \times \frac{1}{1+e^{-ak}} + 2 \\ &\quad \times \frac{u}{a} \times \frac{1}{1+e^{-ak}} - (b - x_1^{(1)}(0)) \end{aligned} \tag{19}$$

$$y = \hat{x}_1^{(1)}(k+1) = (1+e^{-ak}) \times y_1 \tag{20}$$

The LD layer is only one node whose function is only to amplify y_1 in order to make it consistent with (16).

The time response function of the gray differential equation is mapped to a BP network. The weights in this BP network correspond to the parameters of the gray differential equation. The parameters of the gray differential equation are gray. Therefore, the gray property exists in the BP network, which is a gray BP neural network.

2) FAULT PREDICTION FOR VNS

By combining the features of VNs and the advantages of the gray model and BP neural network, the tandem type models will be used. The tandem type model is shown in Fig. 3

The training process of the tandem neural network is as follows.

A series of predictions are obtained by using the selected gray model to predict the original data series. However, there is a deviation between the data predicted by the gray model and the actual values. Therefore, the BP neural network is used to correct these deviations. The data predicted by the gray model is used as the network input, the actual value is used as the network output, and the weights and thresholds of the neural network are trained. Finally, the predicted value of the gray model is input into the neural network, and the predicted value can be obtained in the future. Therefore, the GM (1,1) gray model was used to predict the faults of VNs.

Suppose that you have a set of k historical data Then the gray BP neural network model is used for the predictions as follows

a) The GM (1,1) model is established with the historical data.
b) The predicted data obtained by the model are superimposed to obtain a new data series P.
c) Take the data of several data series as T.
d) The two sequences of P and T are normalized to make them conform to the input conditions of the network.
e) Input P into the input layer of the neural network, input T into the output layer of the neural network, and initially determine the weight and threshold.
f) Start the network learning and obtain the weight and threshold of the error precision.
g) Finally, the GM (1, 1) model established in step (1) is used to predict the value at a future time (such as k + 1). The predicted value is used as the input of the neural network, and the output is the expected predicted result.

TABLE 1. Decision table of rough sets reductions.

Successful delivery ratio	Packet loss ratio	Total cost	Mean latency	Normal	Abnormal	Damage	Congestion
1	0	4.011	0.02	1	0	0	0
0.573	0.427	5.495	0.978	0	1	0	0
0	1	0	0	0	0	1	0
...	...	...	...	...	...	...	...
0.422	0.578	7.747	1.850	0	0	0	1

IV. PERFORMANCE ANALYSIS

A. SIMULATION ENVIRONMENT AND EXTRACTION FAULT DATA

In the NS2, the two-ray ground wireless signal transmission model is used to model the wireless signal transmissions. Nodal traffic patterns such as CBR are used. The MAC layer uses the 802.11 protocol, the routing layer uses the AODV protocol, and the simulation time is set to be 30s. When facing network congestion, many abnormal, damaged nodes can be instantly accessed, and cascading faults would be produced.

By calculating the importance of each attribute using (3), it is concluded that for the specific faults, the network's mean latency, successful delivery ratio, packet loss ratio and total cost are the most important attributes for the fault type. These samples are used as the input samples for the MATLAB simulation.

The latter four items in the table represent the fault types simulated by the network and encode the fault types in the network: 1 means a failure, and 0 means no failure.

B. TRAIN FAULT PREDICTION ALGORITHM

The design of the BP network generally comes from the layers of the network, the number of hidden layer neurons, the selection of the weights' initial values and the learning rate selection. In Matlabr2012a, after 222 iterations, the training value and test value had basically converged and the training accuracy reached the initialized settings, which means that it is complete.

The results of the training, where the actual output of the neural network will be a transpose matrix, are shown in Tab. 2

Tab. 3 shows the desired output of the neural network, where 1 means a failure, and 0 means no failure.

By comparing Tab. 2 and Tab. 3, we found that the actual value, desired output and the network conditions matched,

TABLE 2. Output of the neural network after transposition.

0.99069	0.012137	0.00032675	0.0019752
0.0049663	0.99055	0.0080746	0.025689
0.0017935	0.014324	0.98725	0.00015758
0.0019411	0.047866	7.8849e-06	0.98853

TABLE 3. The expected output of neural network.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

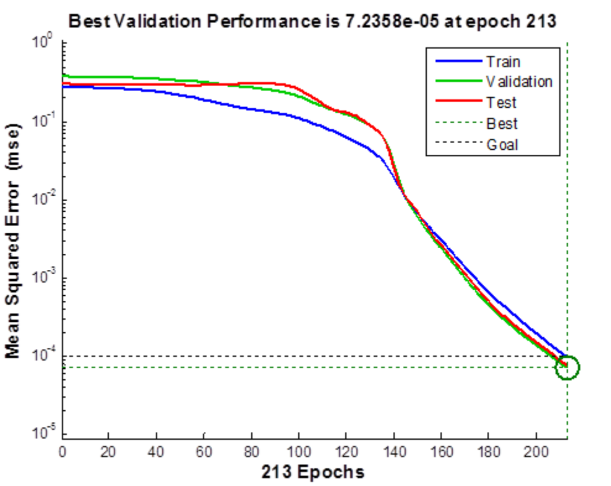


FIGURE 4. The error of the gray neural network.

which indicates that the BP neural network algorithm can predict faults.

C. DATA PREDICTION MODULE BASED ON GRAY ERROR NEURAL NETWORK

The forecast model output is the input of the neural network, and the actual value is the output of the neural network. The Matlabr2012a simulation results are as follows.

The red line represents the error of the testing data, the blue line represents the error of the training data, and the green line represents the error in the ideal case. The goal represents the error accuracy. From Fig. 4, after 213 iterations, the error has basically reached the initial designed accuracy.

Fig. 5, Fig. 6, Fig. 7 and Fig. 8 show the performance of successful delivery ratio, packet loss ratio, total cost and mean latency under different group, respectively. As these

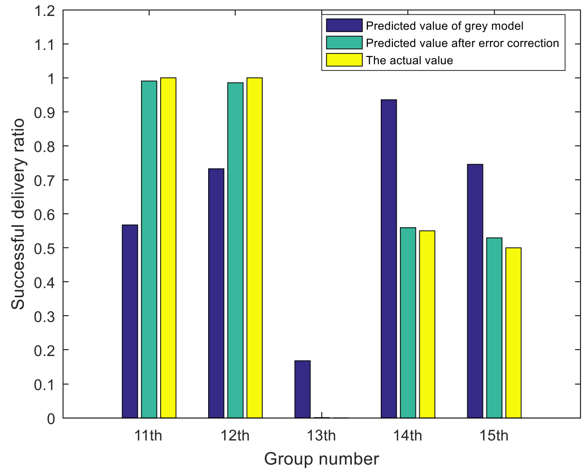


FIGURE 5. Successful delivery ratio under different group.

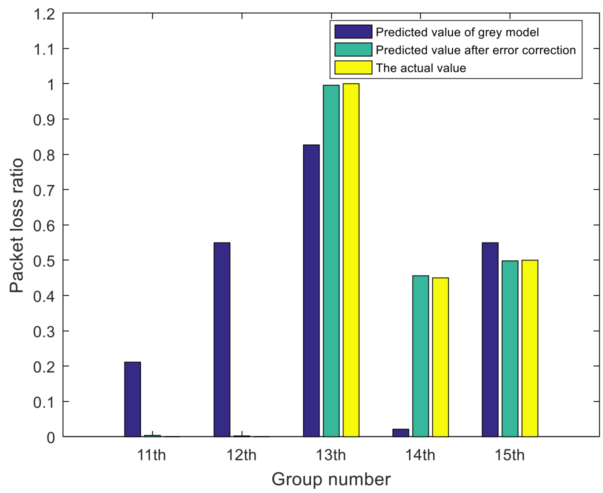


FIGURE 6. Packet loss ratio under different group.

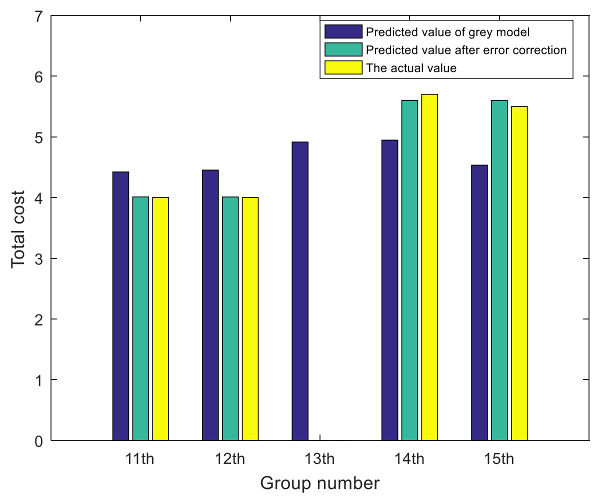


FIGURE 7. Total cost under different group.

figures shown, the difference between the predicted value of gray theoretical model and the actual value relatively large, but the predicted value after the error correction of the reverse

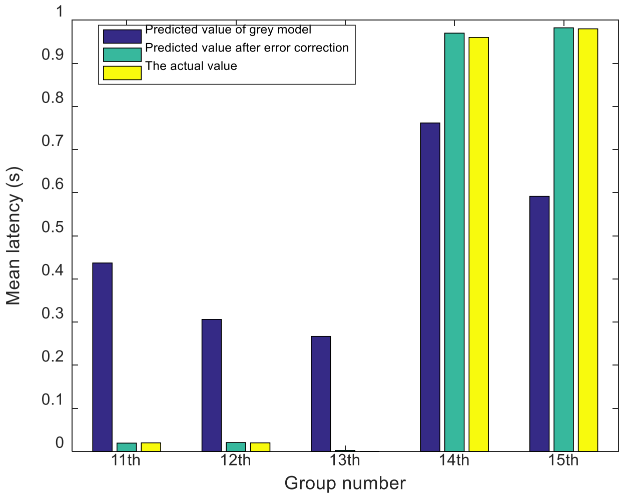


FIGURE 8. Mean latency under different group.

TABLE 4. The fault output of the network prediction.

	Normal	Abnormal	Damage	Congestion
Expected	0	0	0	1
Actual	0.000124	0.00435	0.06347	0.99062

neural network is basically consistent with the actual value of the network. Although there is some error, it is within the allowable error range of the network, because the predicted value cannot be completely coincident with the actual value. It is proved that the reverse neural network has good error correction ability. But in the gray theoretical model, the data of group 11 is obtained through the first 10 groups of data values through one-time accumulation, and by establishing the first-order differential equation. It is found that the error between actual value and predicted value is large. The data of group 12 is by removing the data of group 1, because the correlation between the data of group 1 and the prediction of group 12 is already very small. So, we need to get rid of that. At this point, the actual value of group 11 has occurred. So, we've reconstructed 10 sets of data. Through these 10 sets of data, the data of group 12 is predicted again. Data from groups 13, 14, and 15 repeat the process. The predicted values of groups 13, 14 and 15 were also significantly different from the actual values, indicating that the prediction model itself had a large error. but the predicted value after the error correction of the reverse neural network is basically consistent with the actual value of the network. Although there is some error, it is within the allowable error range of the network, because the predicted value cannot be completely coincident with the actual value. It is proved that the reverse neural network has good error correction ability.

Tab. 4 shows the results of the neural network after training is completed. The test data is input to the trained neural network through the count of the network from the input to output and is used in the calculation of the back propagation-propagation process to modify the weights. After repeated iterations, the output is calculated by the fault prediction algorithm. Then, it predicts the type of network fault abnormal nodes, the nodal damage, the network congestion, and the cascading failures.

V. SUMMARY

According to the characteristics of VNs, we combine the statuses and wireless networks to propose the fault prediction algorithm of rough sets and BP neural network for VNs, and we describe the algorithm in detail. The algorithm is divided into the three modules of data processing, data prediction and fault diagnosis. We use the NS2 network simulation software and MATLAB to analyze the proposed algorithm and its performance. The simulation results show that the proposed fault prediction algorithm can effectively predict the faults The proposed fault prediction algorithm is aimed at an ideal VNs network model, while the actual network environment of VNs is relatively complex. The learning and memory of the network is unstable. In other words, if the number of learning samples increases, the trained neural network will relearn, while the weight of the trained ones is not remembered, which greatly limits the expansion of the network Therefore, the algorithm needs further improvement in these aspects.

REFERENCES

[1] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[2] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2868616.2018.

[3] L. Li *et al.*, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.

[4] Z. Ning, X. Kong, and F. Xia, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Commun. Mag.*, to be published, doi: 10.1109/MCOM.2018.1700895.

[5] X. Wang *et al.*, "A privacy-preserving message forwarding framework for opportunistic cloud of things," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2864782.

[6] Y. Takefuji, "Connected vehicle security vulnerabilities [commentary]," *IEEE Technol. Soc. Mag.*, vol. 37, no. 1, pp. 15–18, Mar. 2018.

[7] X. Wang, Z. Ning, and L. Wang, "Offloading in Internet of vehicles: A fog-enabled real-time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.

[8] L. Gao, F. Li, X. Xu, and Y. Liu, "Intrusion detection system using SOEKS and deep learning for in-vehicle security," *Cluster Comput.*, vol. 3, pp. 1–9, Apr. 2018.

[9] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Commun.*, to be published, doi: 10.1109/MWC.2018.1700441.

[10] L. Guo, Z. Ning, W. Hou, B. Hu, and P. Guo, "Quick answer for big data in sharing economy: Innovative computer architecture design facilitating optimal service-demand matching," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1494–1506, Oct. 2018.

[11] W. Hou *et al.*, "On-chip hardware accelerator for automated diagnosis through human-machine interactions in healthcare delivery," *IEEE Trans. Automat. Sci. Eng.*, to be published.

[12] Z. Ning *et al.*, "A cooperative quality-aware service access system for social Internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2506–2517, Aug. 2018.

[13] X. Li, Y. Yu, G. Sun, and K. Chen, "Connected vehicles' security from the perspective of the in-vehicle network," *IEEE Netw.*, vol. 32, no. 3, pp. 58–63, May/Jun. 2018.

[14] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular social networks: Enabling smart mobility," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 49–55, May 2017.

[15] G. O. Young, "Synthetic structure of industrial plastics," in *Plastics*, vol. 3, 2nd ed., J. Peters, Ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.

[16] Z.-Q. Li, L. Cheng, X.-S. Qiu, and Y.-G. Zeng, "Fault diagnosis for large-scale IP networks based on dynamic Bayesian model," in *Proc. IEEE 5th Int. Conf. Natural Comput.*, Aug. 2009, pp. 67–71.

[17] M. Kokawa, S. Miyazaki, and S. Shingai, "Fault location using digraph and inverse direction search with application," *Automatica*, vol. 19, vol. 6, pp. 729–735, 1983.

[18] M. Elhadef and S. Grira, "Partial syndrome-based system-level fault diagnosis using game theory," *Int. J. Parallel Emergent Distrib. Syst.*, vol. 33, no. 1, pp. 69–86, 2018.

[19] S. Chessa and P. Santi, "Comparison-based system-level fault diagnosis in ad hoc networks," in *Proc. 20th IEEE Symp. Reliable Distrib. Syst.*, Oct. 2001, pp. 257–266.

[20] M. J. Peng *et al.*, "An intelligent hybrid methodology of on-line system-level fault diagnosis for nuclear power plant," *Nucl. Eng. Technol.*, vol. 50, no. 3, pp. 396–410, 2018.

[21] H. Yang, W. Li, C. Zhang, and J. Li, "A new definition of intuitionistic fuzzy rough set and its similarity measure," in *Proc. IEEE Int. Conf. Comput. Sci. Service Syst.*, Aug. 2012, pp. 1852–1855.

[22] B. K. Tripathy and K. G. Rajulu, "On covering based pessimistic multi-granular rough sets," in *Proc. IEEE Int. Conf. Comput. Intell. Commun. Netw.*, Nov. 2014, pp. 708–713.

[23] Y. Liu and W. Zhu, "On three types of covering-based rough sets via definable sets," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jul. 2014, pp. 1226–1233.

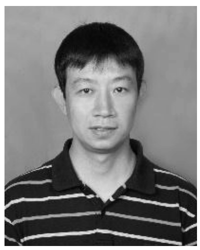
[24] A. Pawlowski *et al.*, "Application of time-series methods to disturbance estimation in predictive control problems," in *Proc. IEEE Int. Symp. Ind. Electron.*, Jul. 2010, pp. 409–414.

[25] G. Manogaran, V. Vijayakumar, R. Varatharajan, P. M. Kumar, R. Sundarasekar, and C.-H. Hsu, "Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering," *Wireless Pers. Commun.*, vol. 102, no. 3, pp. 2099–2116, 2018.

RONG GENG received the B.S. degree in communication engineering, and the M.S. and Ph.D. degrees in communication and information system from Northeastern University, Shenyang, China, in 2001, 2004, and 2010, respectively. She is currently a Lecturer with Northeastern University. Her research interests are space networks, vehicular networks, network security, and routing.

XIAOJIE WANG, photograph and biography not available at the time of publication.

NING YE received the B.S. degree in communication engineering and the M.S. and Ph.D. degrees in communication and information system from Northeastern University, Shenyang, China, in 2005, 2008, and 2014, respectively. She is currently a Lecturer with Northeastern University. Her research interests are topology control, routing and security in mobile ad hoc networks, space information networks, and 5G networks.



JUN LIU received the degree in communication and information system from Northeastern University, Shenyang, China, in 2008. He is currently an Associate Professor with Northeastern University. His research interests include space information network, self-organizing network, and information security.