

Research on Reliability Prediction Methods for Dynamic Networks Based On Deep Neural Networks

Shiyun Zhang, Jianhua Wei*

School of Economics and Management, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

* Corresponding author: Jianhua Wei (Email: 905227339@qq.com)

Abstract: In reality many networks, such as information storage networks, can be modelled as a multi-state stochastic flow network. The reliability of an information storage network is the probability that the network is able to deliver the required amount of information data to the specified server, and its an important metric to assess the performance of an information storage network. Most existing multi-state network reliability assessment algorithms are calculated using very small capacity vectors. However, calculating the reliability of such a network requires reuse of algorithms and is inefficient when the network size, demand, etc., changes rapidly, resulting in managers not making timely decisions accordingly. In order to obtain reliability metrics quickly, this paper proposes a dynamic network reliability prediction model based on deep neural networks (DNNs), which allows the corresponding reliability to be obtained quickly even when the network is changing rapidly. Finally, a local information storage network is used as an example for validation to illustrate the feasibility of the prediction model.

Keywords: Deep Neural Networks, Network Reliability, Dynamic Networks.

1. Introduction

Currently, people's daily lives are intricately linked to network systems, such as electronic information networks [1-3], transportation networks [4, 5], and power distribution networks [6, 7], which ensure the normal operation of our lives. These real-world networks can be described as multi-state stochastic flow networks. Reliability, as a critical indicator for network performance evaluation, plays an essential role in the processes of network construction, operation, and maintenance. In multi-state network systems, reliability is defined as the ability to complete predetermined tasks of material flow, information flow, and energy flow transmission under specific conditions [8-10].

In the field of operations research, one of the methods for evaluating network systems is to model them as a multi-state stochastic flow network. Computing the reliability of a network is an NP-hard problem, and numerous scholars have proposed different methods for assessing network reliability. Generally, there are two approaches: the minimal path method and the minimal cut method [11-25]. Both methods search for all minimal paths or cuts while satisfying requirements and then calculate the reliability through these paths or cuts. In addition, the universal generating function method [26, 27] and simulation method [28, 29] also demonstrate good efficiency in evaluating reliability. However, the above-mentioned reliability assessment methods are based on the evaluation of the reliability of a single network. When the capacity state of the network changes rapidly with time, these methods cannot quickly calculate the network's reliability. To solve this problem, many researchers have used deep neural networks to predict network reliability [30-32]. Deep neural networks have powerful predictive functions and can be used for various prediction tasks, such as image classification [33-34], speech recognition [35-36], natural language processing [37-38], and time series prediction. Huang [30] transformed the capacity state and demand information of the network into a one-dimensional array,

which was used as input information for the deep neural network to achieve the reliability prediction of multistate stochastic cloud/edge-based network. In addition, Huang [31] evaluated the reliability of a manufacturing network with rework probability through a neural network. Huang [32] using artificial neural networks (ANN) to predict network reliability of random capacitated-flow networks under different network connectivity scenarios. However, the above-mentioned models based on neural network predictions all make the assumption that the topological structure of the multi-state stochastic flow network is fixed.

Taking a dynamic information storage network (MDISN) as an example, whose topology and capacity state change rapidly with time. The model proposed by Huang et al [30-32]. Is clearly not suitable for such situations. Therefore, this paper proposes a fast reliability prediction model based on deep neural network for dynamic networks. Achieving reliability prediction requires converting network information into the required input format for the neural network, which includes the network topology, arc capacity and capacity probability, demand, etc. When characterizing a multi-state stochastic flow network, the adjacency matrix is used to describe the topology, and then each piece of information is converted into a one-dimensional array and arranged sequentially. For data with different dimensions, the method of complementing zeros is used to supplement them so that each data has the same dimension.

The arrangement of this article is as follows: Section 2 explains the method and detailed steps for evaluating network reliability based on the minimal path, while introducing the basic concepts and models of neural networks. Section 3 describes the process of generating training and testing data required for the model, as well as the structure and parameters of the deep learning network. Section 4 uses a dynamic MISN as a case study to demonstrate the feasibility of the proposed prediction model in detail. Finally, the conclusion is presented in Section 5.

Table 1. Notations

$G = (V, E, W)$	a multi-state stochastic flow network
V	$V = \{s, 1, \dots, n-2, t\}$ is the set of nodes.
E	arc set.
W	maximum capacity vector.
w_i	largest capacity of e_i .
m	number of arcs.
e_i	the i -th arc in E .
s	the source.
MP_j	the j -th MP.
d	demand level.
X	a vector $X = (x_1, x_2, \dots, x_m)$.
F	$F = \{f_1, f_2, \dots, f_g\}$ is the flow vector
t	the sink.
n	number of nodes.
$\lceil \bullet \rceil$	minimum integer larger than or equal to \bullet .
$\lfloor \bullet \rfloor$	maximum integer smaller than or equal to \bullet .

2. Network Reliability Evaluation for the MDISN

2.1. Multi-state Network Reliability Evaluation

For assessing the reliability of a multi-state information storage network, a static information storage network is modeled as a multi-state stochastic flow network, defined as $G = (V, E, W)$, where $V = \{s, 1, \dots, n-2, t\}$ denotes the set of all nodes (containing source s and sink t) in the network; $E = \{e_i \mid 1 \leq i \leq m\}$ denotes the set of arcs in the network, where e_i denotes the i -th arc in the network and m denotes the number of arcs in the network; $W = \{w_i \mid 1 \leq i \leq m\}$, where w_i denotes the maximum capacity of e_i and satisfies $0 \leq w_i$. $X = \{x_i \mid 1 \leq i \leq m\}$ denotes the capacity state vector of arcs, and $0 \leq x_i \leq w_i$; Assume that there are g MPs in the network, and MP (minimal path) denotes the set of all arcs in a path from s to t .

The following assumptions also need to be satisfied in the assessment of network reliability.

(1) All nodes and arcs are completely reliable and satisfy the law of flow conservation.

(2) The capacity state distribution of the arcs are statistically independent.

(3) The flow through the minimal path to reach the sink t is a non-negative integer.

Suppose f_j ($1 \leq j \leq g$) represent the flow from the j -th minimal path MP_j to the sink. The demand is d . Let $F = \{f_1, f_2, \dots, f_g\}$, named the flow vector, then $F = \{f_1, f_2, \dots, f_g\}$ exactly satisfies the demand d if it satisfies the following conditions:

$$\sum_{j=1}^g f_j = d \quad (1)$$

For the flow vector $F = \{f_1, f_2, \dots, f_g\}$, f_j also satisfies the following condition:

$$f_j \leq \min \{w_i \mid e_i \in MP_j\} \quad (2)$$

In the network, the flow from different minimal paths entering the arc e_i is restricted by the maximum capacity bound w_i , therefore the following constraints should be satisfied:

$$\sum_{e_i \in MP_j} f_j \leq w_i \quad (3)$$

Meanwhile, any F is feasible under $X = (x_1, x_2, \dots, x_m)$ if and only if it meets the following condition:

$$\sum_{e_i \in MP_j} f_j \leq x_i \quad (4)$$

The F which is obtained by filtering the above conditions is called the feasible flow vector, and capacity state vector X can be obtained by the following equation:

$$xi = \lceil \sum_{e_i \in MP_j} f_j \rceil \quad (5)$$

The details of the proof process can be found in [39-44].

Definition 1: $X \leq Y$: $(x_1, x_2, \dots, x_m) \leq (y_1, y_2, \dots, y_m)$ if and only if $x_i \leq y_i$ for $i = 1, 2, \dots, m$.

Definition 2: $X < Y$: $(x_1, x_2, \dots, x_m) < (y_1, y_2, \dots, y_m)$ if and only if $x_i \leq y_i$ for $i = 1, 2, \dots, m$, and at least one j ($1 \leq j \leq m$) exists such that $x_j < y_j$.

The X obtained by Eq. (5) is not always the minimal capacity vector (d -MCV), and the d -MCV can be obtained by the comparison method. The detailed steps of the comparison method [11, 12] are as follows:

(1) $H = \emptyset$ (H is to store the indexes of d -MCVs, and initially, H is empty).

(2) For $i = 1$ to σ & $i \notin H$.

(3) For $j = i + 1$ to σ & $j \notin H$.

(4) If $X_j < X_i$, then X_i is not a d -MCV, $H = H \cup \{i\}$, and go to step (7);

Else if $X_j \geq X_i$, then X_j is not a d -MCV and $H = H \cup \{j\}$.

(5) Next j .

(6) X_i is a d -MCV.

(7) Next i .

Let X_1, X_2, \dots, X_u be all d -MCVs, and network reliability

$R_d = \Pr \left\{ \bigcup_{i=1}^u \{X \mid X \geq X_i\} \right\}$ that can be efficiently calculated

using the recursive sum of disjoint product (RSDP) technique [45]. With the above elaboration, the steps of the traditional method of network reliability assessment can be summarized as Algorithm 1.

Algorithm 1

Input: $G = (V, E, W)$, d .

Step 1. Find all MPs.

Step 2. Find all flow vectors $F = \{f_1, f_2, \dots, f_g\}$ that satisfy Equation(1)(2)(3).

Step 3. Convert each F derived from **Step 2** into (d -MCV candidates) according to Equation (5).

Step 4. Utilize the comparative method to find all d -MCVs.

Step 5. Assume that X_1, X_2, \dots, X_u are all d -MCVs from **Step 4**, then utilize the RSDP technique to compute $R_d = \Pr$

$\left\{ \bigcup_{i=1}^u \{X \mid X \geq X_i\} \right\}$.

Output: R_d .

2.2. DNN Model

Deep neural network is the infrastructure of deep learning. The main purpose of deep learning is to build a deep convolutional neural network (DNN) and use a large amount of data samples as input to generate a model with powerful analysis and identification ability, which is widely used in computer vision, language recognition, natural language processing.

Suppose a DNN model with several inputs and one output is schematized as Figure 1:

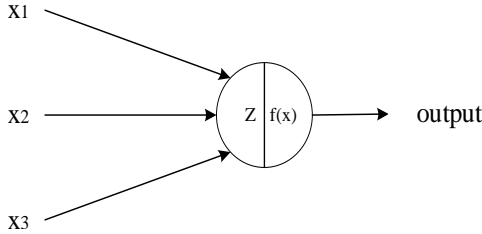


Figure 1. The basic operating principles of neurons

A linear relationship between input and output is obtained by learning, and the intermediate output results is z :

$$z = \sum_{k=1}^v w_k x_k + b \quad (6)$$

Suppose a neuron has v inputs, w_k ($1 < k < v$) is the input x_k with the weight of the neuron, and b denotes the bias of the neuron.

The intermediate output result z is used as the input of the activation function $f(x)$ that yields the output of this neuron. The common activation functions are Sigmoid, Tanh, ReLU, LReLU, PReLU, Swish, etc. Take sigmoid as an example, sigmoid is a highly frequent activation function, also called logistic function, for calculating the output of hidden layer neurons, and takes the value range of $(0, 1)$, and the expression of the function is as follows.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

The internal layers of DNN can be divided into three categories, input layer, hidden layer and output layer. Take Figure 2 as an example, the first layer is the input layer, the last layer is the output layer, and the middle layers are all hidden layers. The layers are fully connected, i.e., any neuron in layer i must be connected to any neuron in layer $i+1$.

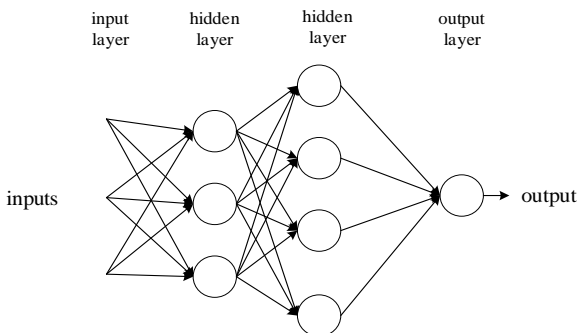


Figure 2. The structure of a deep learning network

The training of deep neural networks usually requires the following steps:

- 1) Weight initialization.
- 2) Forward propagation.

- 3) Compute cost.
- 4) Backward propagation.
- 5) Update parameter.

3. Reliability Prediction

Aiming to make the reliability of the network can be evaluated quickly, this paper makes predictions based on the DNN model with four steps:

Step 1: The topology of the network, maximum capacity of the edges, probability distribution, demand etc. are described by a matrix or vector.

Step 2: The reliability is obtained by using Algorithm 1 with the known information from **Step 1**. Take the known information as input and reliability as output to get a data sample. Repeat **Steps 1, 2** to create a data set with a large number of samples and divide it into a training set and a test set.

Step 3: Set the parameters of the DNN model, such as the number of training times, initial weights, number of hidden layers and hidden layer neurons, learning rate, and number of iterations.

Step 4: The training set is fed into the model for training, and the test set was used for prediction.

3.1. Data Generation

The required information to calculate the reliability of a multi-state network includes the topology, the maximum capacity of each edge and the corresponding state probability distribution. This information is transformed into data that can be recognized by the DNN model through the following operations:

Suppose there are two topologies as shown in Figure 3. The topology, the maximum capacity of the arc and the state probability distribution are known.

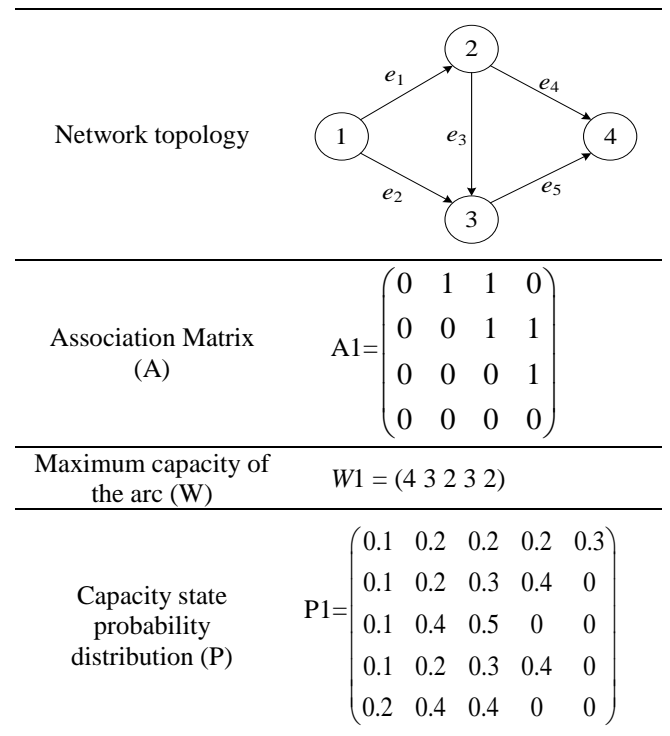


Figure 3. Network topology

For the association matrix A , the internal element $a_{ij} = 1$ means that the i -th node is connected to the j -th node. For the

maximum capacity vector $W = (w_1, w_2, \dots, w_m)$, where w_i denotes the maximum capacity of arc e_i . For a state probability distribution P , the i -th row ($0 \leq i \leq m$) in P denotes the probability of the capacity distribution corresponding to arc e_i . For example, in the first network, the maximum capacity of e_1 is 4, then the probabilities corresponding to the capacities 0, 1, 2, 3, 4 of e_1 are 0.1, 0.2, 0.2, 0.2, 0.3.

3.2. Training Set and Test Set

To obtain real-time network reliability when the network is varying rapidly, a deep learning model is used for reliability prediction. In order to predict reliability index more precisely, a massive number of samples are needed for training. In this study, 10,000 training samples were generated with matlab, and the steps to generate a training sample are shown below:

(1) Suppose the number of network nodes is n , and the association matrix A with $n \times n$ is generated.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

Where $a_{ij} = 0$ or 1, $a_{ij} = 1$ means flow from the i -th node to the j -th node, and $a_{ij} = 0$ means the i -th node is unconnected to the j -th node. The matrix A satisfies

$$m = \sum_{i=1}^n \sum_{j=1}^n a_{ij}.$$

(2) First generate the maximum capacity vector $W = (w_1, w_2, \dots, w_m)$ randomly, let $v = \max(w_1, w_2, \dots, w_m) + 1$, and then generate the probability distribution matrix P of $m \times v$.

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1v} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mv} \end{pmatrix}$$

Where $p_{ij} = 0$ when $w_i + 1 < j \leq v$.

(3) Generate the demand as d , calculate R_d with Algorithm 1.

(4) Expand the above information into a one-dimensional array:

$$\Omega_A = [a_{11}, a_{12}, \dots, a_{1n}, \dots, a_{n1}, a_{n2}, \dots, a_{nn}]$$

$$W = [w_1, w_2, \dots, w_m]$$

$$\Omega_P = [p_{11}, p_{12}, \dots, p_{1v}, \dots, p_{m1}, p_{m2}, \dots, p_{mv}]$$

(5) Combine all information into one data sample: $\Omega_\mu = [\Omega_A, W, \Omega_P, d, R_d]$.

The above steps (1)-(5) are repeated until a sufficient number of datasets are generated. In the process of generating the dataset, the correlation matrix describing the topology and the dimensionality of the state distribution probability matrix will be inconsistent before and after the change of the network, therefore, the operation of filling in the zeros is applied to make the dimensionality consistent. Take the two network structures in the above table as an example. The two network structures in Table 1 above are used as examples to illustrate.

In Figure 3:

$$\Omega_{A1} = [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$$

$$W_1 = [4 \ 3 \ 2 \ 3 \ 2]$$

$$\Omega_{P1} = [0.1 \ 0.2 \ 0.2 \ 0.2 \ 0.3 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0 \ 0.1 \ 0.4 \ 0.5 \ 0 \ 0 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0 \ 0.2 \ 0.4 \ 0.4 \ 0 \ 0]$$

The above Ω_{A1} with 16 elements, W_1 with 5 elements, and Ω_{P1} with 25 elements; but Ω_{A2} with 25 elements, W_2 with 7 elements, and Ω_{P2} with 35 elements. Therefore, 9 (25-16) zeros need to be added after Ω_{A1} , 2 (7-5) zeros after W_1 , and 10 (35-25) zeros after Ω_{P1} in such a way that the two different structured networks correspond to the same dimensionality of the data set Ω_μ .

Therefore, $\Omega_\mu 1 = [\Omega_{A1}, 0, 0, 0, 0, 0, 0, 0, 0, 0, W_1, 0, 0, \Omega_{P1}, 0, 0, 0, 0, 0, 0, 0, 0, 0, d_1, R_{d1}]$.

With each new dataset generated, the dimensionality is compared with all the datasets already generated before, and the datasets with smaller dimensionality are subjected to a zero-fill operation according to the above rules.

All the generated data are divided into a training set and a test set. The training set is fed into the DNN model for training, and the test set is used as a validation to evaluate the performance of the DNN model.

3.3. Network Structure and Parameters of DNN

(1) Network structure

In general, the more the number of hidden layers, the better the learning ability of the model will be, but excessive hidden layers can cause problems such as training difficulties and gradient disappearance.

(2) Learning Rate

The smaller the learning rate, the slower the change rate of the loss function, leading to overfitting. However, a large learning rate leads to gradient explosion, making it difficult to converge the model.

(3) Epoch and batch size

In the process of model training, the complete process of running the model to complete a forward and backward propagation of all the data is called an epoch. In the process of gradient descent model training, the neural network gradually moves from an unfitted state to an optimally fitted state, and then enters an overfitted state after reaching the optimal state. Therefore, the epoch is not as large as possible, but generally between 50 and 200. The more diverse the data, the larger the epoch. When we deal with larger data sets, feeding the network with all the data at once does not give good training results. The number of samples in each batch is called the sample size batchsize. Batchsize is one of the important neural network hyperparameters and is generally set at around 50-400.

(4) Forward propagation algorithm

The forward propagation algorithm uses a number of weight coefficients matrix W , bias vector b to perform a series of linear operations and activation operations with the input value vector X . Starting from the input layer, the algorithm computes backward layer by layer until it reaches the output layer and obtains the output result.

(5) Backward propagation algorithm

The DNN model needs to find the appropriate linear coefficient matrix W and bias vector b for all the hidden layers and output layers, such that the output computed from all the input test data is as similar as possible to the real output value.

(6) Loss function

The process of optimizing the loss function for extreme value solving is most commonly done iteratively by gradient

descent, but it can also be done by other iterative methods such as Newton's method and Newton-like method. Before performing the DNN backpropagation algorithm, we need to choose a loss function to measure the loss between the output calculated from the training samples and the real training sample output. Then this loss function is optimized to minimize the extreme value process, and the backpropagation is continuously updated for a series of linear coefficient matrices W , bias vector b .

The training set is fed into the DNN model for training, and then the test set is used for validation. The predicted values are compared with the true values to calculate the validation metrics. The commonly used validation metrics are Root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and in this paper, RMSE is used as the validation index, and the calculation formula is

$$E = \sqrt{\frac{1}{k} \sum_{i=1}^k (R_i - \hat{R}_i)^2} \quad (8)$$

Where k is the number of test set data, R_i is the predicted value, and \hat{R}_i is the true value.

The process of prediction with a DNN is shown in Figure 5:

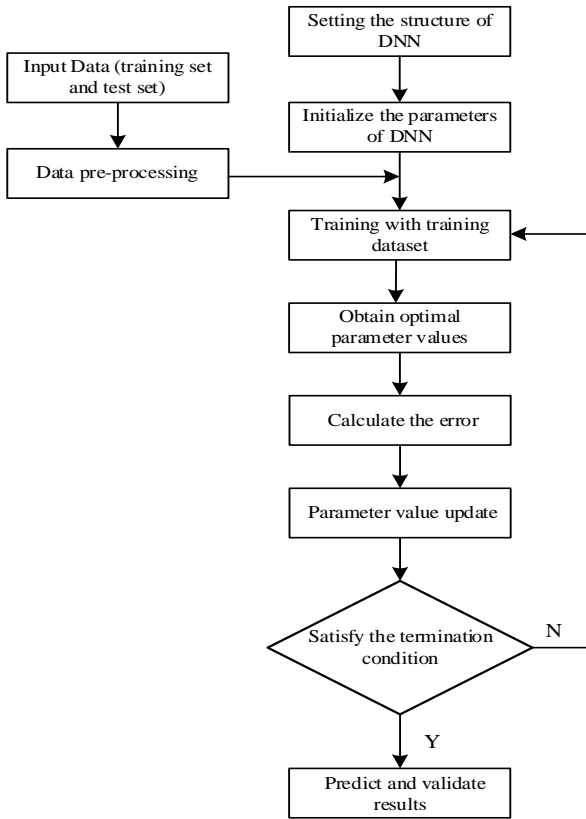


Figure 4. Process of prediction.

4. A numerical Example

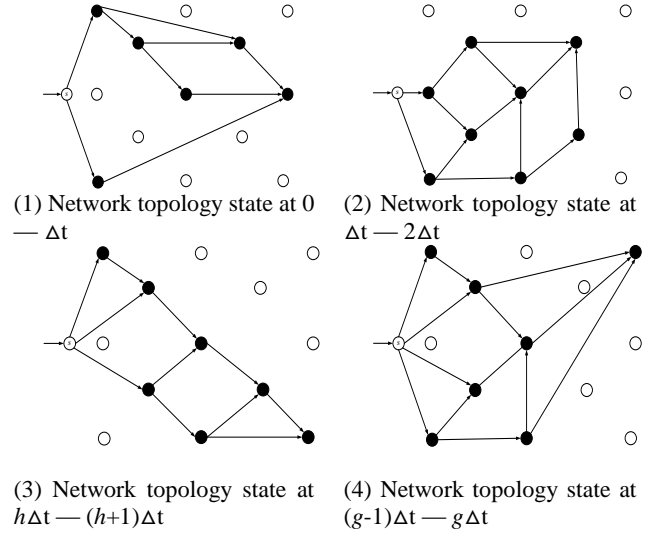
4.1. Description of MDISN

To illustrate the proposed reliability prediction method, A dynamic information storage network (MDISN) is used as an example. In an MDISN network, all nodes (except the source and sink nodes) have two states: busy or idle. When a node is busy, it does not participate in information transmission,

while an idle node participates in information transmission. The source node in the network represents the information server, which receives information and identifies the destination of the transmission. All intermediate nodes can serve as storage or forwarding nodes. During each information transmission service, any two adjacent idle nodes form a transmission arc, with the capacity and distribution being random and independent.

Assuming that the interval between each transmission of the information server (source) is Δt and the whole transmission process is divided into g intervals and the whole topological dynamics of an MDISN evolves as Table 2:

Table 2. Evolution of dynamic networks in terms of time



Where the ● indicates that the state of the node is 1 (idle state) and the ○ indicates that the state of the node is 0 (busy state), the information reception point is random and is determined by the demand place information segment within the information. The above changes represent the random change process of a dynamic information storage network, and the topology of the network before and after the change does not affect each other.

4.2. Parameter Setting

In this research, the DNN structure is set up with one input layer one output layer and two hidden layers with 246 neurons in the input layer, one neuron in the output layer and 8 neurons in both hidden layers.

4.3. Training and Prediction

To ensure the speed and accuracy of the prediction, 3000 training data were generated in this study. Thirty data were selected as the test set for reliability prediction and error verification.

The results obtained by using DNNs for reliability prediction are shown in Table 3:

Table 3. The prediction results

	#	**		#	**
1	0.39388	0.396555	16	0.597544	0.562538
2	0.489242	0.499341	17	0.722112	0.688588
3	0.719897	0.700775	18	0.799615	0.849029
4	0.904888	0.914287	19	0.59636	0.536434
5	0.836764	0.850825	20	0.930725	0.920649
6	0.915486	0.907676	21	0.891704	0.881599
7	0.633266	0.670436	22	0.598828	0.607837
8	0.869711	0.859501	23	0.787892	0.795234
9	0.479199	0.438547	24	0.626878	0.626939
10	0.597251	0.615595	25	0.88529	0.913345
11	0.446727	0.435551	26	0.550347	0.55157
12	0.470692	0.504608	27	0.601518	0.597296
13	0.809594	0.777553	28	0.803686	0.817463
14	0.771331	0.765723	29	0.47761	0.476347
15	0.412146	0.352877	30	0.498263	0.511465
RMSE			0.0255843		
Note: # is the actual values of reliability, and ** is the values of reliability predicted by the DNN model.					

5. Conclusion

The reliability of a network is an important metric needed to measure the efficiency of the network and the decisions of managers. The assessment of the reliability of a dynamic information storage network, whose topology and capacity state change rapidly over time, can be solved using existing algorithms, but when the network structure changes, the algorithm needs to be reused for calculation, which leads to low efficiency in calculating the reliability.

In order to quickly obtain the reliability of dynamic information storage networks within a time period and facilitate rapid decision making by managers, a reliability prediction model based on the DNN model is proposed to achieve efficient prediction of reliability when dynamic information storage networks change. First, 10,000 pieces of data are generated through the steps in Section 3, each piece of data containing the necessary information such as the topology of the network, the capacity state and probability distribution, the demand, and the actual reliability values. Next, a deep neural network was built, with the actual reliability value of each data item as the output and the rest of the information as the input, using sigmoid as the activation function and RMSE as the loss function to train the deep neural network. Finally, 30 pieces of data were selected as the test set and input to the trained network for prediction, and the error between the actual reliability value and the predicted value was compared. The prediction result showed that the RMSE between the actual value and the predicted value was 0.0255843. Overall, the contribution of this paper is based on a deep neural network that inscribes the topology of a multi-state stochastic network through an association matrix to quickly predicting the network reliability in each temporal state.

References

- [1] Y.K. Lin, F. Lance, and P.C. Chang, "Quantifying the impact of correlated failures on system reliability by a simulation approach," *Reliability Engineering & System Safety*, vol.109, no.1, pp.32–41, 2013.
- [2] Y.K. Lin, P.C. Chang, and F. Lance, "Quantifying the Impact of Correlated Failures on Stochastic Flow Network Reliability," *IEEE Transaction on Reliability*, vol.61, no.3, pp.692-701, 2015.
- [3] C.F. Huang, D.H. Huang, and Y.K. Lin, "Network reliability evaluation for a distributed network with edge computing," *Computers & Industrial Engineering*, vol.147, 106429, 2020.
- [4] W.W. Wu, A. Ning, and X.X. Ning, "Evaluation of the reliability of transport networks based on the stochastic flow of moving objects," *Reliability Engineering & System Safety*, vol.93, no.6, pp.838–844, 2008.
- [5] C.T. Yeh, Y.K. Lin, L.C.L. Yeng, and P.T. Huang, "Reliability evaluation of a multistate railway transportation network from the perspective of a travel agent," *Reliability Engineering & System Safety*, vol.214, 107757, 2021.
- [6] H.P. Jia, D.N. Liu, Y.B. Li, Y. Ding, M.G. Liu, and R. Peng, "Reliability evaluation of power systems with multi-state warm standby and multi-state performance sharing mechanism," *Reliability Engineering & System Safety*, vol.204, 107139, 2020.
- [7] Y.K. Lin, C.T. Yeh, "Maximal network reliability for a stochastic power transmission network," *Reliability Engineering & System Safety*, vol.96, no.10, pp.1332-1339, 2011.
- [8] A. Lisnianski, G. Levitin, "Multi-state system reliability: Assessment, optimization and applications," Singapore: World Scientific, 2003.
- [9] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering & System Safety*, vol. 94, pp.125-141, 2009.
- [10] T. Aven, "The reliability science: its foundation and link to risk science and other sciences," *Reliability Engineering & System Safety*, vol.215, 107863, 2021.
- [11] Y.K. Lin, "A simple algorithm for reliability evaluation of a stochastic-flow network with node failure," *Computers & Operations Research*, vol.28, pp.1277-1285, 2001.
- [12] Y.K. Lin, "Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs," *Reliability Engineering & System Safety*, vol.75, no.1, pp.41-46, 2002.
- [13] W.C. Yeh, "A novel method for the network reliability in terms of capacitated minimum-paths without knowing minimum-paths in advance," *Journal of the Operational Research Society*, vol.56, pp.1235-1240, 2005.
- [14] W.C. Yeh, "A fast algorithm for searching all multi-state minimal cuts," *IEEE Transactions on Reliability*, vol.57, no.4, pp.581-588, 2008.
- [15] W.C. Yeh, "A novel node-based sequential implicit enumeration method for finding all d-MPs in a multistate flow network," *Information Sciences*, vol.297, pp.283–292, 2015.
- [16] G.H. Bai, M. Zuo and Z.G. Tian, "Search for all d-MPs for all d levels in multistate two-terminal networks," *Reliability Engineering & System Safety*, vol.142, pp.300-309, 2015.
- [17] G.H. Bai, T. Liu, Y.A. Zhang, and J.Y. Tao, "An improved method for reliability evaluation of two-terminal multistate networks based on state space decomposition," *IEEE Transactions on Reliability*, vol.70, no.3, pp.1084-1095, 2021.
- [18] T. Liu, G. Bai, J. Tao, et al., "Modeling and evaluation method for resilience analysis of multi-state networks," *Reliability Engineering & System Safety*, vol.226, 108663, 2022.

- [19] T. Liu, G.H. Bai, J.Y. Tao, Y.A. Zhang, and Y.N. Fang, "An improved bounding algorithm for approximating multistate network reliability based on state-space decomposition method," *Reliability Engineering & System Safety*, vol.210, 107500, 2021.
- [20] M. Forghani-elahabad, and L.H. Bonani, "Finding all the lower boundary points in a multistate two-terminal network," *IEEE Transactions on Reliability*, vol.66, no.3, pp.677- 688, 2017.
- [21] M. Forghani-elahabad, N. Kagan, and N. Mahdavi-Amiri, "An MP-based approximation algorithm on reliability evaluation of multistate flow networks," *Reliability Engineering & System Safety*, vol.191, 106566, 2019.
- [22] Z.Y. Zhang, and F.M. Shao. "A Diameter-Constrained Approximation Algorithm of Multistate Two-Terminal Reliability," *IEEE Transactions on Reliability*, vol.67, no.3, pp.1249-1260, 2018.
- [23] Y.F. Niu, and X.Z. Xu, "A new solution algorithm for the multi-state minimal cut problem," *IEEE Transactions on Reliability*, vol.69, no.3, pp.1064-1076, 2020.
- [24] Y.F. Niu, Z.Y. Gao and W.H.K. Lam, "A new efficient algorithm for finding d-minimal cuts in multi-state networks," *Reliability Engineering & System Safety*, vol.66, pp.151-163, 2017.
- [25] Y.F. Niu, X.Y. Wan, X.Z. Xu, and D. Ding, "Finding all multi-state minimal paths of a multi-state flow network via feasible circulations," *Reliability Engineering & System Safety*, vol.204, 107188, 2020.
- [26] G. Levitin, "A universal generating function approach for the analysis of multi-state systems with dependent elements," *Reliability Engineering & System Safety*, vol.84, no.3, pp.285-292, 2004.
- [27] G. Levitin, *The universal generating function in reliability analysis and optimization*. London: Springer-Verlag; 2005.
- [28] J.E. Ramirez-Marquez, D.W. Coit, "A Monte-Carlo simulation approach for approximating multi-state two-terminal reliability," *Reliability Engineering & System Safety*, vol.87, no.2, pp. 253-264, 2005.
- [29] George-Williams, H., Patelli, E., "A hybrid load flow and event driven simulation approach to multi-state system reliability evaluation," *Reliability Engineering & System Safety*, vol. 152, pp.351-367, 2016.
- [30] Huang, DH., Huang, CF. & Lin, YK. A reliability prediction model for a multistate cloud/edge-based network based on a deep neural network. *Ann Oper Res* (2022).
- [31] Huang, CF., Huang, DH., Lin, YK. et al. Network reliability evaluation of manufacturing systems by using a deep learning approach. *Ann Oper Res* (2022).
- [32] Huang, CF., Huang, DH., Lin, YK. Network reliability prediction for random capacitated-flow networks via an artificial neural network, *Reliability Engineering & System Safety*, vol.237,109378,2023.
- [33] Gui, Y., Zeng, G. Joint learning of visual and spatial features for edit propagation from a single image. *Vis Comput* 36, 469–482 (2020).
- [34] Ye, Z., George, A., Wu, A.T., Niu, X., Lin, J., Adusumilli, G., Naismith, R.T., Cross, A.H., Sun, P. and Song, S.-K. (2020), Deep learning with diffusion basis spectrum imaging for classification of multiple sclerosis lesions. *Ann Clin Transl Neurol*, 7: 695-706.
- [35] Phapatanaburi, K., Wang, L., Sakagami, R. et al. Distant-talking accent recognition by combining GMM and DNN. *Multimed Tools Appl* 75, 5109–5124 (2016).
- [36] Zoughi, T., Homayounpour, M.M. A Gender-Aware Deep Neural Network Structure for Speech Recognition. *Iran J Sci Technol Trans Electr Eng* 43, 635–644 (2019).
- [37] S. Guo, J. Liu, Z. Yang, W. You and R. Zhang, "Linguistic Steganalysis Merging Semantic and Statistical Features," in *IEEE Signal Processing Letters*, vol. 29, pp. 2128-2132, 2022.
- [38] Su, Y, Wang, Z, Jin, S, Shen, W, Ren, J, Eden, MR. An architecture of deep learning in QSPR modeling for the prediction of critical properties using molecular signatures. *AIChE J*. 2019; 65:e16678.
- [39] W.C. Yeh, "A new approach to evaluate reliability of multistate networks under the cost constraint," *Omega*, vol.33, no.3, pp. 203-209, 2005.
- [40] W.C. Yeh, "Multistate network reliability evaluation under the maintenance cost constraint. *International Journal of Production Economics*," vol.88, pp.73-83, 2004.
- [41] Y.K. Lin, and P.C. Chang, "Evaluation of system reliabilities for a maintainable stochastic-flow network," *IEEE Transactions on Reliability*, vol.61, no.2, pp.398-409, 2012.
- [42] C.F. Huang, Y.K. Lin, and L.C.L. Yeng, "Routing scheme of a multi-state computer network employing a retransmission mechanism within a time threshold," *Information sciences*, vol.340, pp.321-336, 2016.
- [43] Y.K. Lin, C.T. Yeh, C.F. Huang, "Reliability assessment of a multistate freight network for perishable merchandise with multiple suppliers and buyers," *International Journal of Systems Science*, vol.48, no.1, pp.74-83, 2017.
- [44] Y.K. Lin, C.T. Yeh, C.F. Huang, "Reliability evaluation of a stochastic-flow distribution network with delivery spoilage," *Computers & Industrial Engineering*, vol.66, no.2, pp.352-359, 2013.
- [45] M. Zuo, Z.G. Tian, and H.Z. Huang, "An efficient method for reliability evaluation of multistate networks given all minimal path vectors," *IIE Transactions*, vol.39, pp.811–817, 2007.