CHAPTER 3 PROPOSED METHODOLOGIES

This chapter proposes algorithms for freeway incident detection. Section 3.1 introduces the construction of fuzzy neural network, including approach-based and lane-based fuzzy neural network, and back-propagation training method. Section 3.2 presents the rolling-trained procedure, which can update the network parameters in response to the prevailing traffic conditions. Section 3.3 details the chaotic diagnosis which examining the abnormal change in traffic dynamics due to the incident. Section 3.4 illustrates the definitions of incident detection performance.

According the literature review, early AID algorithms focused on pattern recognition approaches and statistical approaches. To improve the incident detection performance and to achieve real-time detection, more advanced approaches have been introduced, such as artificial intelligent based approaches, fuzzy set theory approaches, and image processing based approaches. The detection performance in terms of detection rate and false alarm could be sensitive to the chosen traffic parameters, their designated criteria for judging the incident occurrence, and the detection locations. It can also be sensitive to the changes in prevailing traffic conditions. In practice, the complexity of traffic dynamics is characterized with uncertain and nonlinear nature. Most previous AID algorithms, however, subjectively set the parameters and use crisp criteria in distinguishing the abnormal traffic (incident data) from the normal one (incident-free data), thus they may result in poor detection performance as the traffic conditions alter drastically.

In dealing with the subjective selection of the thresholds and uncertain contexts (unclear input-output relationships or imprecise input values), both neural networks (NN) and fuzzy systems (FS) have been proven as powerful tools. NN generally represents a complex system with precise inputs and outputs used for training the generic model to formulate a good approximation of the unclear relationship. FS, in contrast, addresses the imprecision of the input and output variables (often defined with fuzzy numbers) but their interrelationships take the form of well-defined if-then rules (Tsoukalas and Uhrig, 1997). Each of these two tools has its own advantages and disadvantages. For instance, the NN approaches have the advantages of learning capability to avoid subjectively setting of the parameters and possessing high fault tolerance due to the distributed memory of parameters separately stored on each link of the network. However, NN approaches usually require long training time, especially when such network parameters as training rate, momentum and initial weights are not appropriately chosen (Wasserman, 1993; Shepherd, 1997). This may preclude the online training procedure for some advanced applications where real-time adjustments are required. The crisp criteria to judge for any event occurrence may be too sensitive, leading the NN approaches to misjudge easily.

Taking traffic incident detection as an example, the distributed memory of parameters separately stored on each link of a NN will have the advantage of high fault tolerance. Consequently, reducing the number of input nodes or poor quality of few input data will not remarkably influence the output results. However, the crisp criteria for judging the incident occurrence used in NN approaches may easily cause misjudgment due to too sensitive to the crisp criteria. If incorporating fuzzy inference into the NN (called FNN), we can avoid the too sensitive problem but still possess self-learning capability with high fault tolerance. Consequently, FNN approaches have been commonly employed in traffic engineering, ranging from pavement diagnosis (Lan and Chiou, 1997), vehicular count and classification (Lan and Kuo, 2002; Lan, *et al.*, 2003a) to traffic prediction (Abdulhai, *et al.*, 2002; Yin, *et al.*, 2002). More recently, Lan, *et al.* (2004) developed incident detection algorithms with various FNN structures. Off-line tests have validated that their proposed FNN system was capable of detecting the freeway incidents with rather high accuracy. Sensitivity analysis further showed that alternating the FNN structures by reducing the number of detectors or number of input traffic parameters only slightly deteriorated the detection performance, providing strong evidences of high fault tolerance of the FNN incident detection system. Based on the self-learning capacity and high fault tolerance, the FNN approach is established as the incident detection algorithm using the traffic data directly collected by detectors.

However, their FNN approach did not adaptively adjust the network parameters in response to the prevailing traffic conditions, hence there may have some room for the improvement. To capture the change in traffic dynamics through the network training, Yin, *et al.* (2002) developed a FNN-type model with online rolling-trained procedure to predict the traffic flows in an urban street network. Their FNN model consists of two modules: a gate network and an expert network. The gate network classifies the inputs into several clusters using a fuzzy approach and the expert network specifies the input-output relationship as in a conventional NN approach. Both simulation and real observation data demonstrated that the prediction power can be enhanced through the online rolling-trained procedure in response to the prevailing traffic conditions. Inspired by Yin’s *et al*. (2002) work, this research presumes that the rolling-trained procedure in FNN might be imperative in augmenting the incident detection performance. Thus, this research attempts to develop a rolling-trained fuzzy neural network (RTFNN) approach for freeway incident detection. Its underlying logic is to establish a proper fuzzy neural network and then adaptively adjust the network parameters using the most up-to-date traffic data in response to the prevailing traffic conditions so as to improve the detection performance over the conventional FNN approach.

Most of the conventional incident detection algorithms, including pattern recognition, statistical approach, catastrophe theory, artificial intelligent based approach and fuzzy set theory, are mainly based on the change in some traffic parameters such as flow, speed, occupancy rate and density. These algorithms often have difficulties in distinguishing in traffic data during incidents from similar patterns (Cheu and Ritchie, 1995). Sheu and Ritchie (1998) have proposed a new methodology which is capable of detecting incidents promptly as well as characterizing incidents in terms of time-varying lane changing fractions and queue lengths in blocked lanes, lanes blocked due to incidents, and incident duration. This research attempts to use the change in chaotic traffic parameters, including embedding dimension, fractal dimension, correlation dimension, Lyapunov exponent, BDS exponent, complexity, entropy and delay time, to examine the existence of traffic incident. Takens' embedding theorem is used to reconstruct the phase spaces of both normal and incident traffic flow time series. If an incident occurs, some of the above-mentioned chaotic traffic parameters may change. The chaotic-based approach, which is selecting the chaotic parameters with significant change and designate appropriate threshold values to discriminate the incident traffic from the normal flows, is proposed in this research.

3.1 The Structure of Fuzzy Neural Network

The framework of fuzzy neural network (FNN) and training process are shown in this section. The approach-based fuzzy neural network (see Figure 3-1), which using the station averages across all lanes regardless of the number of lanes, is built for the transferability potential, which means that separate algorithms would need not to be developed and trained for a freeway facility with 3, 4, 5 or more lanes. Furthermore, it is worthy to construct the lane-based fuzzy neural network (see Figure 3-2), which using lane-specific traffic data, to enhance the accuracy of the incident detection algorithms.

3.1.1 Layers of Fuzzy Neural Network

The FNN structure of the proposed models is established with four layers. The first layer is the input layer, which processes all of the traffic flow information. The second layer is the membership layer, which processes the original traffic flow data through the corresponding relationship of membership functions and calculates its fuzzy membership. The third layer is the rule layer composed of three categories of fuzzy inference rules: time-specific, lane-specific and space-specific. The fourth layer is the output layer. The components of each layer and their relationships are detailed as follows.

(1) The First Layer

<u>Approach-based</u>

Twelve nodes are designed in this layer to input the approach traffic parameters at upstream and downstream detectors. These nodes represent the average speeds of previous time step (S_{u0}^1) and current time step (S_{u1}^1), the aggregated flows (F_{u0}^1 and F_{u1}^1) and the average densities (D_{u0}^1 and D_{u1}^1) of the upstream detector, and the average speeds (S_{d0}^1, S_{d1}^1), the aggregated flows (F_{d0}^1, F_{d1}^1) and the average densities (D_{d0}^1, D_{d1}^1) of the downstream detector. Note that the above densities are not directly measured from the detectors, but indirectly calculated from the detected flows and speeds. Because of the concept that any stationary detector cannot readily obtain the density data from the field and the surrogate measure for density is to use the detector occupancy (% OCC). Additionally, the density is difficult to measure in the field, a surrogate of it, such as percent occupancy that can be readily provided by the field detectors, should be used in the future work.

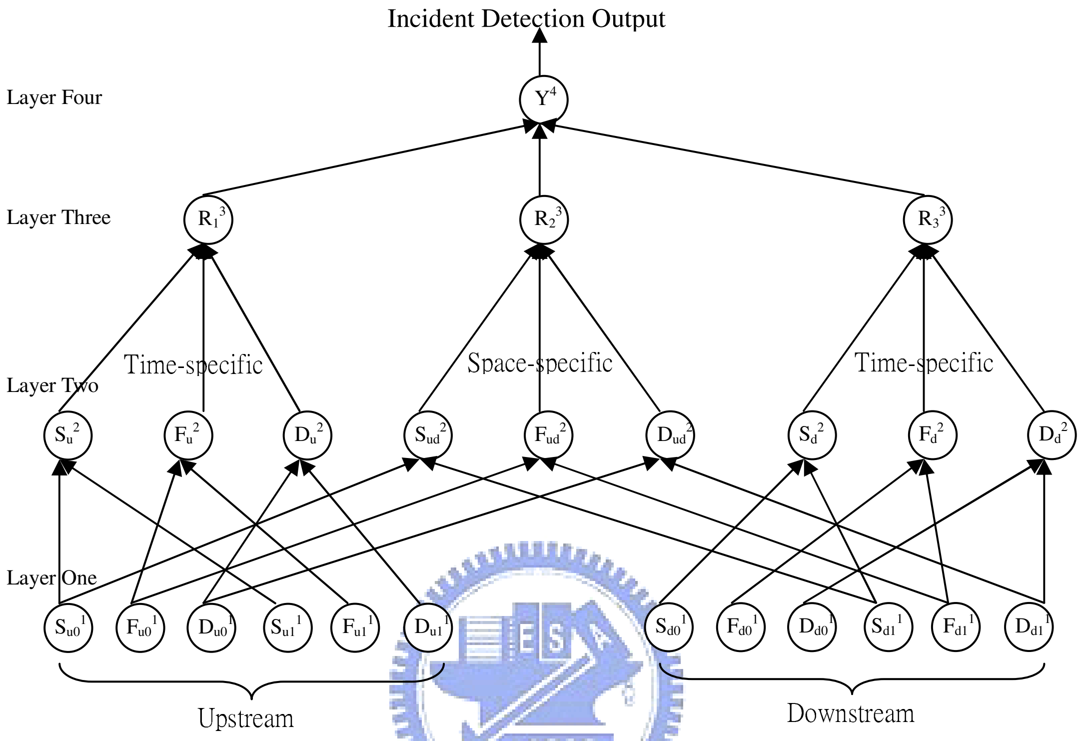

Figure 3-1 The approach-based fuzzy neural network structure

Lane-based

Twenty-four nodes are designed in this layer to input the lane-specific traffic parameters at upstream and downstream detectors. These nodes represent the speeds of previous time step (S_{u10}^1) and current time step (S_{u11}^1), flows (F_{u10}^1 and F_{u11}^1) and densities (D_{u10}^1 and D_{u11}^1) of the upstream inner lane, speeds (S_{u20}^1, S_{u21}^1), flows (F_{u20}^1, F_{u21}^1) and densities (D_{u20}^1, D_{u21}^1) of the upstream outer lane, speeds (S_{d10}^1, S_{d11}^1), flows (F_{d10}^1, F_{d11}^1) and densities (D_{d10}^1, D_{d11}^1) of the downstream inner lane, and speeds (S_{d20}^1, S_{d21}^1), flows (F_{d20}^1, F_{d21}^1) and densities (D_{d20}^1, D_{d21}^1) of the downstream outer lane. Note that the above densities are not directly measured from the detectors, but indirectly calculated from the detected flows and speeds.

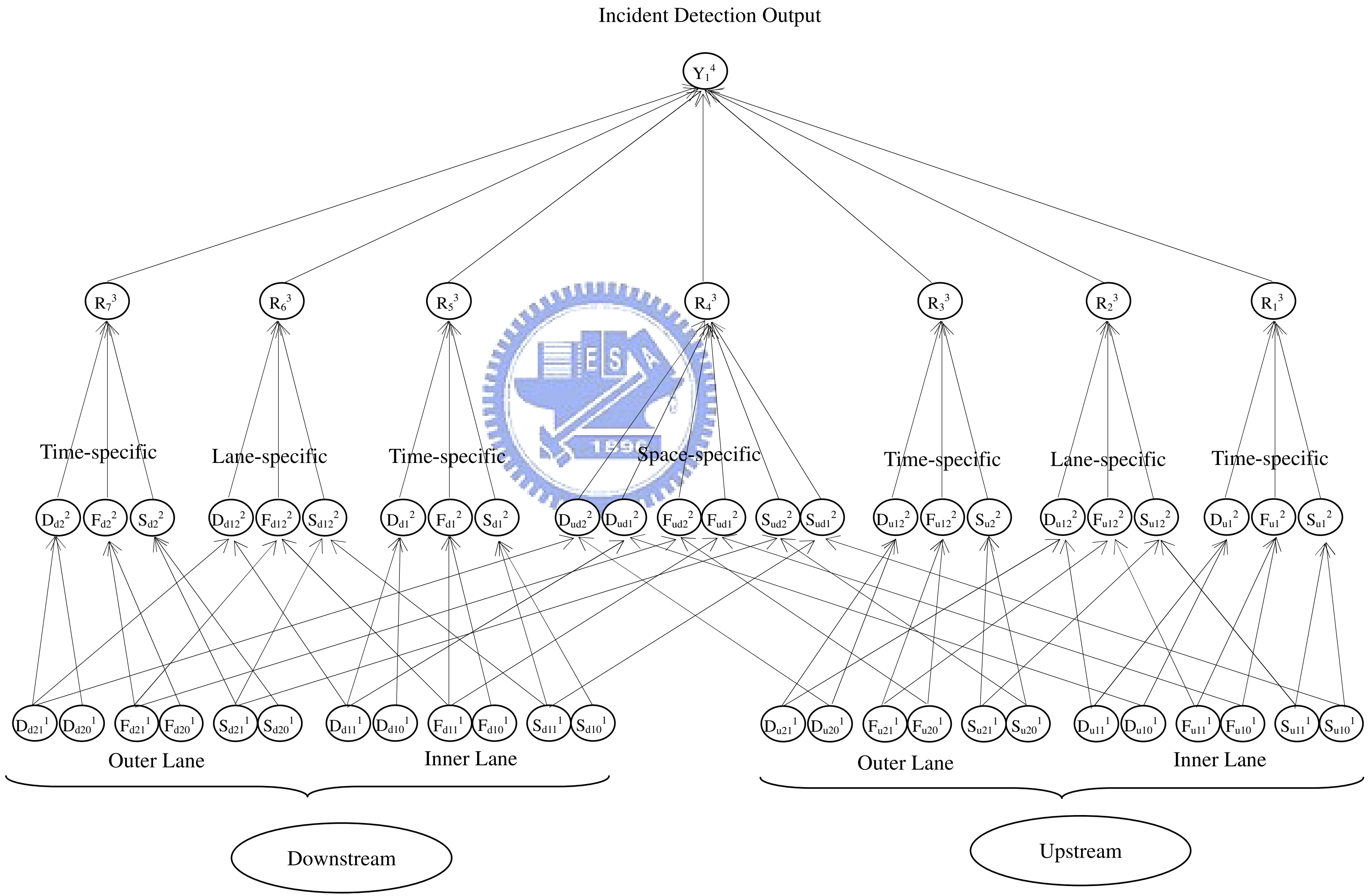

Figure 3-2 The lane-based fuzzy neural network structure

The weighted values w_i in this layer are set equal to one and there is no need for adjustment. The output values o_i are expressed as:

$$o_i = f(u_i) = u_i \quad i=1\sim12 \text{ (approach-based) or } 1\sim24 \text{ (lane-based)} \tag{3-1}$$

$$o_i = x_i \cdot w_i = x_i = u_i \tag{3-2}$$

where u_i are the input values.

(2) The Second Layer

Approach-based

A trapezoid membership function as shown in Figure 3-3 is used. The nodes in the second layer fall into two categories. The time-specific category (6 nodes) compares the upstream and downstream approach-based speeds, flows and densities at present time with those at the previous time step (upstream: S_u^2, F_u^2, D_u^2; downstream: S_d^2, F_d^2, D_d^2). The space-specific category (3 nodes) compares the flows, speeds and densities between upstream at time t and downstream at time $t+\tau$, where τ is the time lag measured by the time for vehicles traveling from upstream detecting point to downstream detecting point. These space-specific nodes calculate the membership degrees of the difference of speeds, flows and densities between upstream and downstream (S_{ud}^2, F_{ud}^2, D_{ud}^2).

Lane-based

A trapezoid membership function as shown in Figure 3-3 is used. The nodes in the second layer fall into three categories. The time-specific category (12 nodes) compares the upstream and downstream lane speeds, flows and densities at present time with those at the previous time step (upstream: S_{u1}^2, F_{u1}^2, D_{u1}^2, S_{u2}^2, F_{u2}^2, D_{u2}^2; downstream: S_{d1}^2, F_{d1}^2, D_{d1}^2, S_{d2}^2, F_{d2}^2, D_{d2}^2). The lane-specific category (6 nodes) calculates the membership degrees of the difference of speeds, flows and densities between upstream lanes (S_{u12}^2, F_{u12}^2, D_{u12}^2) and downstream lanes (S_{d12}^2, F_{d12}^2, D_{d12}^2). The space-specific category (6 nodes) compares the flows, speeds and densities between upstream at time t and downstream at time $t+\tau$, where τ is the time lag measured by the time for vehicles traveling from upstream detecting point to downstream detecting point. These space-specific nodes calculate the membership degrees of the difference of speeds, flows and densities between upstream and downstream (inner lane: S_{ud1}^2, F_{ud1}^2, D_{ud1}^2; outer lane: S_{ud2}^2, F_{ud2}^2, D_{ud2}^2).

The weighted values in the second layer w_{ij} are also set equal to one and there is no need for further adjustment. Both a_j and b_j are parameters of the trapezoid membership function, whose output values o_j can be written as:

$$o_j = f_j(u_j) = \mu_j(x_{ij}) = \begin{cases} 0 & \text{for } x_{ij} \le a_j \\ \dfrac{x_{ij} - a_j}{b_j - a_j} & \text{for } a_j < x_{ij} \le b_j \\ 1 & \text{for } x_{ij} > b_j \end{cases} \tag{3-3}$$

In the time-specific category $x_{ij} = |u_i - u_{i+1}|$, where *i=1, 3, 5* and *j=1~3* for upstream inner lane, *i=7, 9, 11* and *j=7~9* for upstream outer lane, *i=13, 15, 17* and *j=16~18* for downstream inner lane, and *i=19, 21, 23* and *j=22~24* for downstream outer lane. In the lane-specific category $x_{ij} = |u_i - u_{i+6}|$, where *i=2, 4, 6* and *j=4~6* for upstream, *i=14, 16, 18* and *j=19~21* for downstream. In the space-specific category $x_{ij} = |u_i - u_{i+13}|$, where *i=1, 3, 5* and *j=10, 12, 14* for inner lane, *i=7, 9, 11* and *j=11, 13, 15* for outer lane.

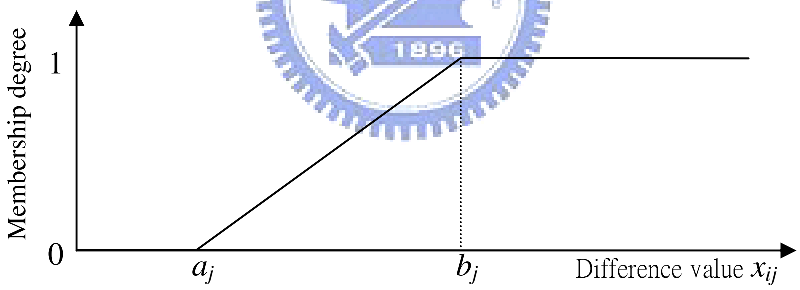

Figure 3-3 The membership function

(3) The Third Layer

<u>Approach-based</u>

The rules in the time-specific category are stated as: **IF** there is a remarkable difference of speeds, flows or densities between the present time step and the previous one, upstream or downstream **THEN** an incident occurrence is inferred with membership degrees R_1^3 (upstream) and R_3^3 (downstream). The output values o_k can be expressed as:

$$o_k = f(u_k) = (w_{jk} \cdot x_{jk}) * (w_{(j+1)k} \cdot x_{(j+1)k}) * (w_{(j+2)k} \cdot x_{(j+2)k}) \quad (3\text{-}4)$$

where $j=1$ and $k=1$ for upstream inner lane, $j=7$ and $k=3$ for upstream outer lane, $j=16$ and $k=5$ for downstream inner lane, $j=22$ and $k=7$ for downstream outer lane.

The rules in the space-specific category are stated as: **IF** there is a remarkable difference of speeds, flows or densities between the upstream at time t and downstream at time $t+\tau$, inner lane or outer lane, **THEN** an incident occurrence is inferred with membership degrees R_4^3. The output values o_k can be expressed as:

$$o_k = f(u_k) = (w_{jk} \cdot x_{jk}) * (w_{(j+1)k} \cdot x_{(j+1)k}) * (w_{(j+2)k} \cdot x_{(j+2)k}) * (w_{(j+3)k} \cdot x_{(j+3)k}) * (w_{(j+4)k} \cdot x_{(j+4)k}) * (w_{(j+5)k} \cdot x_{(j+5)k}) \quad (3\text{-}5)$$

for $j=10$ and $k=4$.

Lane-based

The rules in the time-specific category are stated as: **IF** there is a remarkable difference of speeds, flows or densities between the present time step and the previous one, upstream or downstream, inner lane or outer lane, **THEN** an incident occurrence is inferred with membership degrees R_1^3 (upstream inner lane), R_3^3 (upstream outer lane), R_5^3 (downstream inner lane), and R_7^3 (downstream outer lane). The output values o_k can be expressed as:

$$o_k = f(u_k) = (w_{jk} \cdot x_{jk}) * (w_{(j+1)k} \cdot x_{(j+1)k}) * (w_{(j+2)k} \cdot x_{(j+2)k}) \quad (3\text{-}6)$$

where $j=1$ and $k=1$ for upstream inner lane, $j=7$ and $k=3$ for upstream outer lane, $j=16$ and $k=5$ for downstream inner lane, $j=22$ and $k=7$ for downstream outer lane.

The rules in the lane-specific category are stated as: **IF** there is a remarkable difference of speeds, flows or densities between the inner lane and the outer lane, upstream or downstream, **THEN** an incident occurrence is inferred with membership degrees R_2^3 (upstream) and R_6^3 (downstream). The output values o_k are represented as:

$$o_k = f(u_k) = (w_{jk} \cdot x_{jk}) * (w_{(j+1)k} \cdot x_{(j+1)k}) * (w_{(j+2)k} \cdot x_{(j+2)k}) \quad (3\text{-}7)$$

for upstream: $j=4$ and $k=2$; downstream: $j=19$ and $k=6$.

The rules in the space-specific category are stated as: **IF** there is a remarkable difference of speeds, flows or densities between the upstream at time t and downstream at time $t+\tau$, inner lane or outer lane, **THEN** an incident occurrence is inferred with membership degrees R_4^3. The output values o_k can be expressed as:

$$o_k = f(u_k) = (w_{jk} \cdot x_{jk})*(w_{(j+1)k} \cdot x_{(j+1)k})*(w_{(j+2)k} \cdot x_{(j+2)k})*(w_{(j+3)k} \cdot x_{(j+3)k})*(w_{(j+4)k} \cdot x_{(j+4)k})*(w_{(j+5)k} \cdot x_{(j+5)k}) \tag{3-8}$$

for $j=10$ and $k=4$.

(4) The Fourth Layer

The fourth layer is the output layer, which contains one node Y_1^4 for this two-lane freeway. The center of area method is employed to defuzzify the fuzzy number to a crisp binary value ($Y_1^4=0$ indicates incident-free; $Y_1^4=1$ represents incident-occurrence). It is essential to set the initial weighted values for this layer w_{km} and then adjust them through the network training. The output values o_m are:

$$o_m = f(u_m) = \sum_{k=1}^{3} w_{km} \cdot x_{km} \tag{3-9}$$

3.1.2 Back-propagation Training Algorithm

The structure of FNN would be introduced. In this section, the learning process, including supervised learning (back-propagation, perception, and counter-propagation), unsupervised learning (self-organizing map and adaptive resonance theory) and associate learning (hopfield and bi-directional associative memory), to train the proposed FNN would be discussed. The back-propagation learning process is the most popular training algorithm in the training of artificial neural networks. The learning process, called the generalized delta rule, involves the presentation of a set of pairs of input patterns and target output patterns. The system of neural network with randomly (or otherwise) initialized weights uses the given input vector to produce its own output vector and compares this with the target output pattern. When there is a difference, the rule will change weights appropriately. There are some parameters to be adjusted in the

FNN. However, the link weighted values of first layer to second layer and second layer to third layer are set equal to one and there is no need for adjustment. Because that first layer is input layer, which distributed the original traffic flow data to FNN, and second layer is membership function, which directly carried the fuzzy membership degree into fuzzy inference layer. Only the parameters of the membership function in the second layer, fuzzy inference rules in the third layer, and the weighted values between the third and fourth layers need to adjust by training process. After parameters are initialized, a gradient descent-based back-propagation algorithm is used to train the proposed FNN approaches. By learning the training patterns from training data set, the parameters in the FNN approaches could be updated. The main goal of supervised learning is to minimize the total error function. To show that the learning method is based on back-propagation algorithm, the inference due to error for each layer should be computed. Because, this procedure can be started from fourth layer since the error is feed-backward from output layer.

A back-propagation approach, minimizing the total error function with the gradient steepest descent method, is used for the training process. The process is described as follows.

Step 1: Initialize the network parameters, including weighted value (w_{km}), membership function parameters (a_j, b_j), momentum term (α), and learning rate (η) presents the number of current training epochs. Learning rate (η) would decrease as the number of training cycles (n) increases. Initially, the network parameters, including momentum parameter (α), a_j, b_j are set as 0.5, 0, 1 respectively.

Step 2: Obtain the output value (o_m^4) with the above-mentioned FNN by inputting a training sample using the existing network parameters. A training sample is composed of an input vector (flow, speed, and density of the approach-based or lane-based data) and output vector (binary incident occurrence information). The output value is calculated as formula 3-9 shown.

Step 3: Calculate the error of the fourth layer (δ_m^4):

$$\delta_m^4 = d_m^4 - o_m^4 \quad (3\text{-}10)$$

where d_m^4 is the observed output value of training sample.

Step 4: Update the weighted values between the third and fourth layers (w_{km}):

$$w_{km}(t+1) = w_{km}(t) + \eta \cdot \delta_m^4 \cdot x_{km} + \alpha \cdot [w_{km}(t) - w_{km}(t-1)] \quad (3\text{-}11)$$

Step 5: Calculate the errors of the third and second layers (δ_k^3, δ_j^2):

$$\delta_k^3 = \delta_m^4 \cdot w_{km}(t+1) \quad (3\text{-}12)$$

$$\delta_j^2 = \delta_k^3 \cdot w_{jk} \cdot x_{jk} \, . \quad (3\text{-}13)$$

Step 6: Adjust the parameters of the membership function in the second layer (a_j, b_j):

$$a_j(t+1) = a_j(t) + \eta \cdot \delta_j^2 \cdot x_k \cdot \frac{x_j - b_j(t)}{[b_j(t) - a_j(t)]^2} + \alpha \cdot [a_j(t) - a_j(t-1)] \quad (3\text{-}14)$$

$$b_j(t+1) = b_j(t) + \eta \cdot \delta_j^2 \cdot x_k \cdot \frac{a_j(t) - x_j}{[b_j(t) - a_j(t)]^2} + \alpha \cdot [b_j(t) - b_j(t-1)] \cdot \quad (3\text{-}15)$$

Step 7: Repeat Steps 2 through 6 until all training samples have been inputted. Each routine is finished called an epoch.

Step 8: Calculate the value of total error function of i^{th} epoch (TE_i). In each epoch, the total error function is calculated as follow:

$$TE_i = \frac{1}{2}\sum_{t=1}^{N}[d_m(t) - o_m(t)]^2 \quad (3\text{-}16)$$

where N is the total number of training samples.

Step 9: Test the stop condition.

Training can be terminated when the predetermined number of training epochs reaches or the total error function converges; otherwise, go to step 2. In this paper, the later condition is used. Namely,

$$|TE_n - TE_{n-1}| \le \varepsilon \quad (3\text{-}17)$$

where ε is an arbitrary small number. We stop the network training as TE_i decreases smoothly.

3.2 The Rolling-trained Procedure

A back-propagation technique that minimizes the total error function with gradient steepest descent method is used for the network training. To capture the fluctuations of traffic, we further develop a rolling-trained procedure. The most up-to-date flow parameters are used to distinguish the traffic characteristics of one time interval from another. The proposed rolling-trained procedure is depicted in Figure 3-4 and detailed as follows.

Step 1: Gather the initial training data.

Step 2: Train and update the network parameters by back-propagation algorithm.

The back-propagation algorithm (see 3.1.2) may not have appropriate initial values, thus its network parameters need to be updated through the initial training process.

Step 3: Collect new training data.

Gather input data and conduct incident detection through the FNN algorithm. In the meantime, save both input data and output results in the training sample dataset for the follow-up training.

Step 4: Verify the incident by persistence tests.

As an incident can last for a while, it is necessary to conduct the persistence tests to avoid including the incorrect (misjudged) data in the training dataset. The underlying philosophy of a persistence test is that if there are no continuous detections of an incident occurrence, then that detected incident should be considered as a false alarm. In this case we should discard the training sample and go back to Step 3.

Step 5: Update the training dataset.

Put into the training dataset the data that have passed the persistence tests in step 4. Replace the most distant data with the most updated data so that the training dataset can be maintained at a predetermined training sample size. Go back to step 2 every fixed time interval (e.g., every hour; hereafter, rolling horizon) to keep renewing the network parameters to the latest traffic conditions. The idea of RTFNN is to capture the change in traffic dynamics through rolling-trained procedure so as to adaptively adjust the network parameters. Therefore, the potential advantage for fixing the training sample size is to avoid too many obsolete traffic data which may significantly differ from the prevailing traffic conditions.

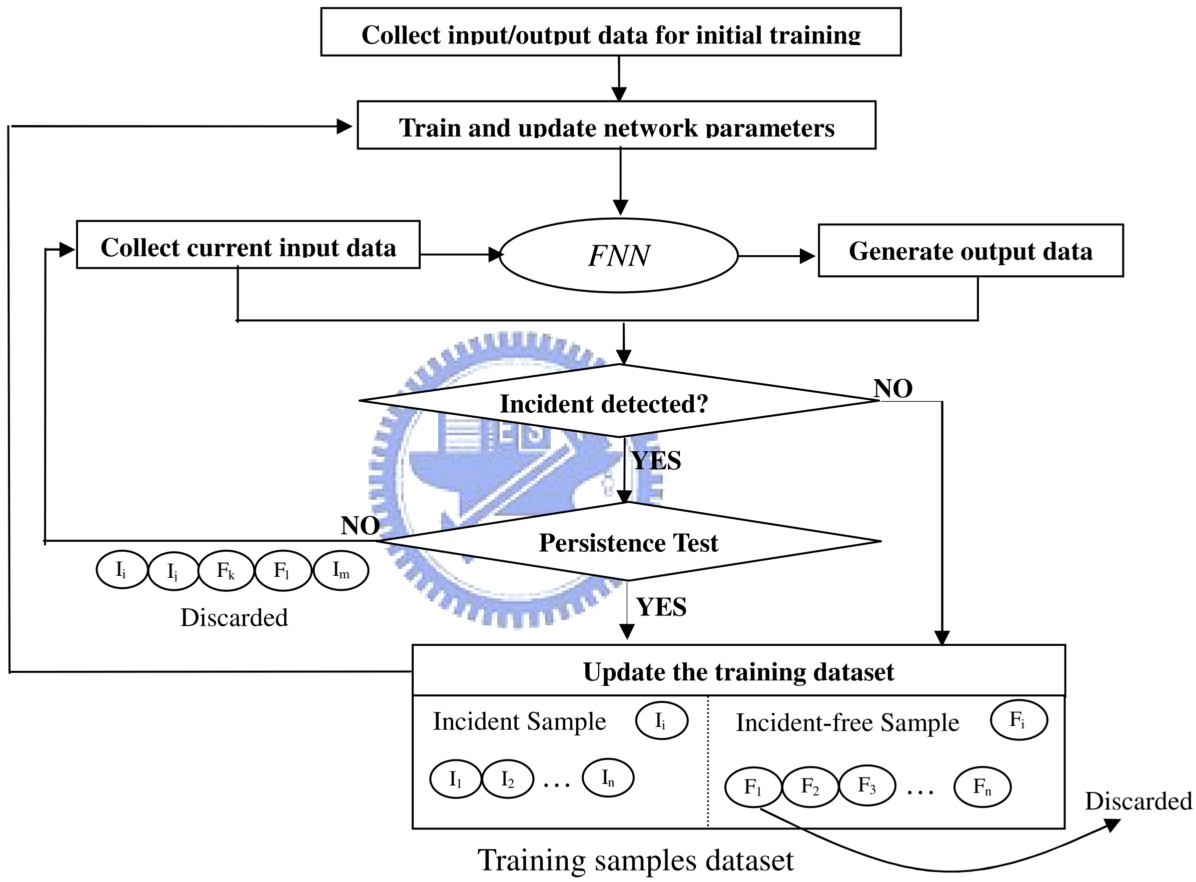

Figure 3-4 The rolling-trained procedure

3.3 Chaotic Diagnosis

The philosophy of chaotic diagnosis for freeway incidents is to use the change in appropriate chaotic parameters to examine the abnormal change in traffic dynamics due to the incident. Figure 3-5 presents the framework for our proposed chaotic diagnosis.

Firstly, we generate a real incident on the freeway mainline. The observed traffic flow time series data are tested for the existence of chaos. Next, the variations of some selected chaotic parameters before and after the traffic incident are compared and those with significant changes are selected as measures for abnormality diagnosis. Then, we use a traffic simulator (Paramics) to generate sufficient flow data with different incident scenarios. The values of the significant chaotic parameters are calculated and their corresponding threshold values are determined so as to distinguish the abnormal traffic dynamics from the normal one. Finally, off-line tests are carried out to evaluate the performance of this chaotic diagnosis.

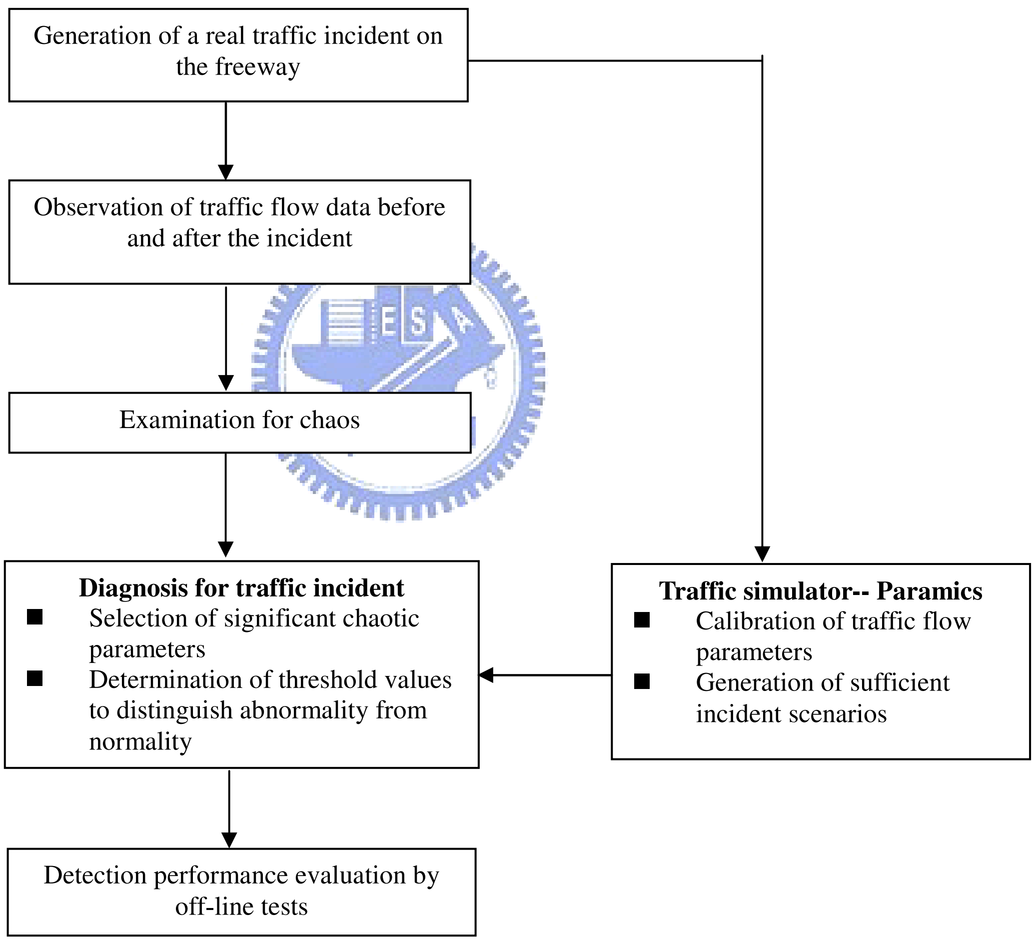

Figure 3-5 Framework of chaotic diagnosis for incident detection

In this section, we briefly introduce the examination for chaos, including largest Lyapunov exponent, capacity dimension, correlation dimension, relative Lz complexity, Kolmogorov entropy, delay time and Hurst exponent. Details of these indexes can be found in Sprott and Rowlands (1995).

(1) Largest Lyapunov exponent

Perhaps the simplest way of testing for chaos is to compute the dominant Lyapunov exponent. Testing for a positive value for Lyapunov exponent is equivalent to testing for the SDIC property of chaos. The Lyapunov exponents for a dynamical system are measures of the average rate of divergence or convergence of typical trajectories in the phase space. A positive Lyapunov exponent is a measure of the average exponential divergence of two nearby trajectories. If a discrete nonlinear system is dissipative, a positive Lyapunov exponent is an indication that the system is chaotic (Gencay, 1996). The definition of Lyapunov exponent is expressed as follows:

$$\lambda = \frac{1}{n}\ln\frac{d_n}{d_0} \qquad (3\text{-}18)$$

where

$x_i = i^{th}$ value from the time series data

$x_j = j^{th}$ value that is close to x_i

$d_0 = |x_j - x_i|$

$d_n = |x_{j+n} - x_{i+n}|$

Chaotic orbits should have at least one positive Lyapunov exponent. For periodic orbits, all Lyapunov exponents are negative. Hence, testing for positive largest Lyapunov exponent produces a direct examination for chaos.

(2) Capacity dimension

It is an estimate of the fractal dimension of any attractor that results when the data are plotted in a reconstructed phase space with some high embedding dimension. Normally, the capacity dimension for a chaotic time series is less than 5. A capacity dimension greater than 5 implies essentially random time series data. For more details of the related capacity dimension test, see Mandelbrot (2000).

(3) Correlation dimension

Correlation dimension, applied to characterize chaotic attractors, is widely used by the physicists to test for chaos in time series data (Hilborn, 1994). Compared with other measures such as capacity dimension, it has a computational advantage because it uses the trajectory points directly and does not require a separate partitioning of the state space. This method is based on the correlation integral $C(\varepsilon)$. Grassberger and Procaccia (1983) define the correlation dimension of a time series as

$$D^N = \lim_{\varepsilon \to 0}[\log C(\varepsilon) / \log \varepsilon] \tag{3-19}$$

where N is the embedding dimension. A correlation dimension greater than about 5 implies essentially random data. For more details of the related Grassberger-Procaccia correlation dimension test, see Kantz and Schreiber (1997).

(4) Relative Lz complexity

It is a measure of the algorithmic complexity of the time series. Maximal complexity (randomness) has a value of 0. In the calculation, each data point is converted into a single binary digit depending on whether its value is greater than or less than the median value. For more details of the related Relative Lz complexity test, see Kantz and Schreiber (1997).

(5) Kolmogorov entropy

Similar to Lyapunov exponent, Kolmogorov entropy (or Kolmogorov-Sinai invariant) also focuses on the concept of SDIC for chaos. Consider two trajectories representing time paths that are extremely close so as to be indistinguishable. However, as time passes, these two trajectories diverge so that they become distinguishable. The Kolmogorov entropy (*K*) measures the speed with which such a divergence takes place and is given by (Hilborn, 1994)

$$K = \lim_{\varepsilon \to \infty} \lim_{m \to \infty} \lim_{N \to \infty} \ln\left(\frac{C^M(\varepsilon)}{C^{M+1}(\varepsilon)}\right) \tag{3-20}$$

where

$C^M(\varepsilon)$ = correlation integral with M historical data.

m = number of historical time series data

N = number of dimension reconstructing phase space

If a time series is non-complex and completely predictable, K will approach to zero. If a time series is completely random, the value will approach to very large. That is, the lower the value of K, the more predictable the system is. For chaotic systems, one would expect small K values.

(6) Delay time

Delay time, or correlation time, is a measure of the time over which the data correlate with preceding data. It is the value where the correlation function first falls to 1/e of its fully correlated value (Kantz and Schreiber, 1997).

(7) Hurst exponent

Hurst exponent is a measure of the extent to which the data can be represented by a random walk which is called Brownian motion. In such a case that a time series trajectory x_t , on average, moves away from its initial position by an amount proportional to the square root of time, we say that its Hurst exponent is 0.5 (Kantz and Schreiber, 1997). Thus, we can judge whether the time series is random or not by this test. It is determined from the square root relation between increments and time intervals as follows:

$$\left(\Delta x^2\right) \propto \Delta t^{2H} \qquad (3\text{-}21)$$

where H is the Hurst exponent. For a time series data, H close to 0.5 implies the data is random and uncorrelated. H greater than 0.5 indicates the time series data are positively correlated (or persistence). H less than 0.5 indicates the time series data are anti-correlated (or anti-persistence).

3.4 Definitions of AID Performance

The detection performance is evaluated by three criteria: detection rate (DR), false alarm rate (FAR), and time to detection (TTD), which are defined as follows.

- DR is defined as the ratio of number of detected incidents to the actual number of actual incidents. The DR is given as a percentage.

- FAR is defined as the fraction of incorrect detections to the total number of incident-free. The FAR is typically expressed as a percentage.

- TTD is the difference between the time the incident is detected by the algorithm and the actual time the incident occurs.