On-road sensor configuration design for traffic flow prediction using fuzzy neural networks and Taguchi method

Kit Yan Chan, *Member, IEEE*, Tharam S. Dillon, *Fellow, IEEE*

***Abstract* — On-road sensors provide proactive traffic control centers with current traffic flow conditions in order to forecast the future conditions. However, the number of on-road sensors is usually huge, and not all traffic flow conditions captured by these sensors are useful for predicting future traffic flow conditions. The inclusion of all captured traffic flow conditions is an ineffective means of predicting future traffic flow. Therefore, the selection of appropriate on-road sensors, which are significantly correlated to future traffic flow, is essential, although the trial and error method is generally used for the selection. In this paper, the Taguchi method, which is a robust and systematic optimization approach for designing reliable and high-quality models, is proposed for determinations of appropriate on-road sensors, in order to capture useful traffic flow conditions for forecasting. The effectiveness of the Taguchi method is demonstrated by developing a traffic flow predictor based on the architecture of fuzzy neural networks which can perform well on traffic flow forecasting. The case study was conducted based on traffic flow data captured by on-road sensors located on a Western Australia freeway. The advantages of using the Taguchi method can be indicated: (a) traffic flow predictors with high accuracy can be designed; and (b) development time of traffic flow predictors is reasonable.**

***Index Terms*—sensor configuration, traffic flow prediction, fuzzy neural networks, Taguchi method, traffic flow control, orthogonal array, fuzzy systems, on-road sensor**

I. Introduction

Traffic flow predictors are generally used to provide future traffic flow conditions up to ten minutes ahead to proactive traffic control centers [22], in order to generate remedial actions to improve the mobility of transportation [25]. Prior to developing those predictors, an appropriate on-road sensor configuration, which illustrates the on-road sensor locations and the number of on-road sensors, has to be determined. Even though reasonable results can be obtained using statistical methods [3, 19, 20, 26], and more convincing results can be obtained using the universal estimator [18, 24], namely fuzzy neural networks (FNNs) [4, 12, 27], the determination of appropriate on-road sensor configuration has yet to be resolved.

Zhang et al. [29] and Lachtermacher and Fuller [15] also mentioned that determination of an appropriate input configuration for artificial network design is significant for time-series forecasting. It is not effective to use all traffic flow patterns captured by all the on-road sensors, as some patterns are useless for forecasting purposes. Ignoring useful patterns or including too many useless patterns, which are captured by the on-road sensors, may significantly affect the forecasting accuracy. Also, the optimal configuration of the FNN is closely correlated with the configuration of the on-road sensors, where the traffic flow patterns captured by the on-road sensors are fed into the FNN. When there are *n* on-road sensors installed along the freeway, people need to test 2^n configurations in order to find the appropriate one. Hence, it is impossible to test all on-road sensor configurations, when the number of on-road sensors is large. However, so far no research studies have specified appropriate on-road sensor configurations for traffic flow predictors. The on-road sensor configuration is usually determined by trial and error.

Therefore, it is desirable to develop a systematic and effective methodical approach for determining the appropriate on-road sensor configuration for FNN, in order to obtain more accurate forecasting. In quality control, the Taguchi method has been successfully used to design reliable and high-quality products at low cost for various products such as automobiles and consumer electronics [5, 14], where the number of design factors of products is huge, and finding appropriate design configurations is impossible by conducting experiments for all design configurations. The Taguchi method intends to identify the appropriate design configuration in order to optimize the defined performance characteristic and reduce the sensitivity of engineering designs to the sources of variation.

Similarly, determination of appropriate on-road sensor configurations can be considered as designs of high quality products [10], where both designs are intended to seek the appropriate configuration involving a large number of design factors. In this paper, we propose the Taguchi method as a means of developing optimal on-road sensor configurations for FNN for traffic flow forecasting. In accordance with the Taguchi method, the on-road sensor configurations are arranged in an inner orthogonal array. The Taguchi method conducts systematic trials based on orthogonal arrays to study the on-road sensor configurations using a small number of trials, and then it estimates the significant on-road sensors which are critical for the development of traffic flow predictors.

Kit Yan Chan are with the Department of Electrical and Computer Engineering, Curtin University, Perth, Western Australia, Australia (corresponding author to provide phone: 61-8-9266 9269; fax: 61-8-9266 7548; e-mail: kit.chan@curtin.edu.au). Tharam S. Dillon is with Curtin University, Perth, Western Australia, Australia.

A case study is conducted using traffic flow data captured by on-road sensors installed on a Western Australia freeway. Results obtained by the proposed approach are compared with those obtained by the stochastic algorithms namely genetic algorithms (GA) [9] and particle swarm optimization (PSO) [7] for optimizing input node configurations of artificial networks. The effectiveness of both algorithms in solving some difficult optimization problems such as noise controller design, [2, 21], estimation of sensor node locations [17], image retrieval [16], etc. has been demonstrated. Also, both GA [8] and PSO [28] have been applied for optimizing input node configurations of artificial networks. Based on the comparisons, two advantages of using the Taguchi method merge: (a) high accuracy for traffic flow forecasting; and (b) short development time for the FNN.

The rest of the paper is organized as follows. Section 2 defines and describes the on-road sensor configuration of the FNN for traffic flow forecasting. In Section 3, the main operations of using the Taguchi method for determining appropriate on-road sensor configuration are discussed. Finally, the discussion of the results and several conclusions regarding on-road sensor configuration design using the Taguchi method, are given in Section 4.

II. Traffic flow prediction using on-road sensors

Figure 1 shows the commonly used on-road sensor configuration of a section of a freeway, which consists of *n* on-road sensors (S_1, S_2, …and S_n) between the starting point A and the end point B. When people are driving at location A and are intending to go to destination C, future traffic flow condition at B is essential in order for the traffic control center to indicate the best route for them. If the traffic flow condition at location B is predicted to be smooth in the near future, the traffic control center will advise the drivers to go directly via the freeway to B, leave the freeway by the off-ramp at B, and then reach C. Alternatively, if the traffic flow conditions are predicted to be congested at B, the traffic control center will advise them to leave the freeway by the off-ramp in A, and then use the minor road to go to C.

To forecast future traffic flow conditions at location B illustrated in Figure 1, a traffic flow predictor was developed based on the traffic flow conditions collected by the on-road sensors, where S_i is the *i*-th on-road sensor with *i*=1,2…,*n*. S_i is used to capture the traffic flow condition, $y_i(t)$, at time *t* with a sampling time of T_s. This traffic flow condition is usually indicated by the average speed of cars. If this average speed is close to the speed limit of the freeway, the traffic flow on the freeway is smooth. If the average speed of the cars is far below the speed limit, traffic congestion has occurred.

The car speed is captured by the on-road sensor which consists of two inductive loop detectors namely (ILDs) separated by a small distance. Each ILD consists of a big loop of metallic coil which is buried beneath a lane of the freeway. These two ILDs are connected to a roadside station, which provides power to the loops and processes the information obtained from the ILDs to determine if a car is passing over. To determine the car speed, the time takes for the car to travel between the two ILDs is first captured. The speed of the car is calculated based on the time difference between the two captures and the distance between the two ILDs. Then, the car speeds are transferred to the proactive traffic control center. Traffic flow conditions are indicated by these car speeds. Hence, the proactive traffic control center can use those traffic flow conditions to control ramp meters, to identify congestion points, and to detect traffic incidents etc.

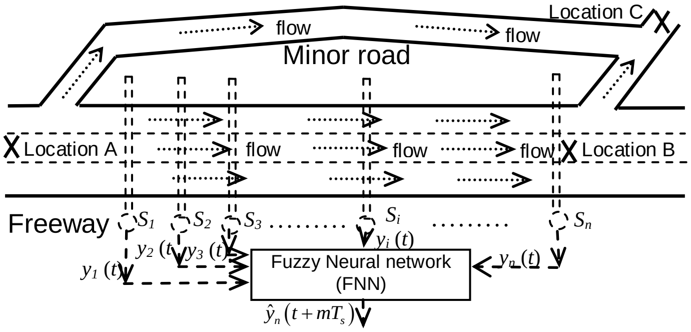

Fig. 1 Traffic flow forecasting using on-road sensors on a freeway

Then, the future traffic flow condition at location B, $y_n(t+m)$, which is *m* sample time ahead, can be predicted by the fuzzy neural network (FNN) with a window size 1, when the current traffic flow conditions captured by the *n* on-road sensors, $y_1(t)$, $y_2(t)$, …., and $y_n(t)$, are available. The FNN consists of n_{rule} fuzzy rules of which the *g*-th fuzzy rule is given by:

$$R_g:\ \text{IF}\ y_{p(1)}(t)\ \text{is}\ \text{A}_{1,g}\left(y_{p(1)}(t)\right)\ \text{AND}\ y_{p(2)}(t)\ \text{is}\ \text{A}_{2,g}\left(y_{p(2)}(t)\right)$$

$$\text{AND ..... AND}\ y_{p(n_S)}(t)\ \text{is}\ \text{A}_{n_S,g}\left(y_{p(n_S)}(t)\right)$$

$$\text{THEN}\ z_g(t)=w_{0,g}+\sum_{i=1}^{n_S}y_{p(i)}(t)\cdot w_{i,g} \qquad (1)$$

where $g=1,2,....,n_{rule}$; the $p(1)^{th}$, $p(2)^{th}$,..., and $p(n_s)^{th}$ on-road sensors are used by the FNN for forecasting the future traffic flow conditions; n_s is the total number of on-road sensors used by the FNN; and $y_{p(i)}(t)$, with $i=1,2,...,n_s$, is the traffic flow condition captured by the $p(i)^{th}$ on-road sensor; $p(i)\in[1,2,...,n]$ with all $p(i)\neq p(j)$, $i,j=1,2,...,n_s$, but $n_s\leq n$.

$w_{i,g}$ is the *i*-th consequent parameter with respect to the polynomial of the *g*-th rule; and $A_{i,g}\left(y_{p(i)}(t)\right)$, with $i=1,2,...,n_s$, is the *i*-th membership function with respect to the *g*-th fuzzy rule, which is given by:

$$A_{i,g}\left(y_{p(i)}(t)\right)=e^{-\left(y_{p(i)}(t)-\bar{y}_{p(i),g}\right)^2/2\sigma^2_{p(i),g}}. \qquad (2)$$

$\bar{y}_{p(i),g}$ and $\sigma_{p(i),g}$ are the mean value and the standard deviation of the membership function respectively. The grade of membership of each rule is defined as:

$$\mu_g(t)=A_{1,g}\left(y_{p(1)}(t)\right)\times A_{2,g}\left(y_{p(2)}(t)\right)\times...\times A_{n_S,g}\left(y_{p(n_S)}(t)\right)$$

or

$$\mu_g\left(t\right) = \prod_{i=1}^{n_s} A_{i,g}\left(y_{p(i)}\left(t\right)\right),$$

where $g = 1, 2, ..., n_{rule}$. The predicted future traffic flow condition at the location of the *n*-th on-road sensor, $\hat{y}_n\left(t + T_s\right)$, is given by:

$$\hat{y}_n\left(t + m \cdot T_s\right) = \frac{\sum_{g=1}^{n_{rule}} \mu_g\left(t\right) z_g\left(t\right)}{\sum_{g=1}^{n_{rule}} \mu_g\left(t\right)}, \tag{3}$$

or

$$\hat{y}_n\left(t + m \cdot T_s\right) = \frac{\sum_{g=1}^{n_{rule}}\left(\prod_{i=1}^{n_s} A_{i,g}\left(y_{p(i)}\left(t\right)\right) \cdot \left(w_{0,g} + \sum_{i=1}^{n_s} y_{p(i)}\left(t\right) \cdot w_{i,g}\right)\right)}{\sum_{g=1}^{n_{rule}}\left(\prod_{i=1}^{n_s} A_{i,g}\left(y_{p(i)}\left(t\right)\right)\right)}. \tag{4}$$

Both equation (3) and (4) can be denoted by the following functional relationship:

$$\hat{y}_n\left(t + m \cdot T_s\right) = f_{FNN}\left(y_{p(1)}\left(t\right), y_{p(2)}\left(t\right), ..., y_{p(n_s)}\left(t\right)\right) \tag{5}$$

where f_{FNN} represents the function of the FNN, which consists of the FNN parameters, $w_{0,g}$, $w_{i,g}$, $\bar{y}_{p(i),g}$ and $\sigma_{p(i),g}$, with $i = 1, 2, ..., n_s$ and $g = 1, 2, ..., n_{rule}$.

The generalization capability of the FNN is evaluated by the mean absolute relative error (e_{MARE}), which indicates the differences between the real collected traffic flow data and the forecasting obtained by the FNN. Based on the collected traffic flow data, e_{MARE} can be found by the following formulation:

$$e_{MARE} = \frac{1}{N_{Data}} \sum_{k=1}^{N_{Data}} \frac{\left| y'_n\left(t\left(k\right) + m \cdot T_s\right) - \hat{y}'_n\left(t\left(k\right) + m \cdot T_s\right)\right|}{y'_n\left(t\left(k\right) + m \cdot T_s\right)}, \tag{6}$$

where N_{Data} is the number of pieces of traffic flow data; and $\hat{y}'_n\left(t\left(k\right) + m \cdot T_s\right)$ is an estimate of the traffic flow condition with *m* sampling time ahead, which is given by:

$$\hat{y}'_n\left(t\left(k\right) + m \cdot T_s\right) = f_{FNN}\left(y'_{p(1)}\left(t\left(k\right)\right), y'_{p(2)}\left(t\left(k\right)\right), ..., y'_{p(n_s)}\left(t\left(k\right)\right)\right);$$

The *k*-th piece of traffic flow data can be written as:

$$\left[y'_n\left(t\left(k\right) + m \cdot T_s\right), y'_{p(1)}\left(t\left(k\right)\right), y'_{p(2)}\left(t\left(k\right)\right), ..., y'_{p(n_s)}\left(t\left(k\right)\right)\right],$$

where $y'_{p(i)}\left(t\left(k\right)\right)$ is the average car speed collected from the $p\left(i\right)^{\text{th}}$ on-road sensor at the time, $t\left(k\right)$, and $y'_n\left(t\left(k\right) + m \cdot T_s\right)$ is the average car speed collected from the n^{th} on-road sensor at the time, $\left(t\left(k\right) + m \cdot T_s\right)$.

A FNN, can be obtained by performing two main task, in order to obtain accurate forecasting:

(i) Determination of the appropriate on-road sensor configuration for the FNN: It can be performed by determining the appropriate on-road sensors which are used for capturing traffic flow conditions to the FNN. Hence, the values of $p\left(1\right)$, $p\left(2\right)$, ...and $p\left(n_s\right)$, and the value of n_s need to be determined.

(ii) Determination of the optimal FNN parameters: After the appropriate on-road sensor configuration is determined, the optimal FNN parameters can be ascertained. This is performed by determining the optimal FNN parameters, $w_{0,g}$, $w_{i,g}$, $\bar{y}_{p(i),g}$ and $\sigma_{p(i),g}$, with $i = 1, 2, ..., n_s$ and $g = 1, 2, ..., n_{rule}$.

The literature indicates that well-established algorithms have been developed for determining the optimal FNN parameters, when a set of historical traffic flow data is available. In this research, Jang's algorithm [6] is used to determine the FNN parameters with respect to the pre-determined FNN input configuration, because of its fast convergence.

Determination of the on-road sensor configuration is important, as it may significantly affect the FNN in forecasting future traffic flow. When patterns of significant on-road sensors are not included, the trained FNN may result in under-learning. The FNN cannot anticipate important behaviors of traffic flow. Hence, the FNN cannot fully learn the traffic flow behaviors, or alternatively, the FNN can only partially learn some behaviors of the traffic flow. When too many patterns of insignificant on-road sensors are included, unnecessary effort is required to train the FNN. As unnecessary patterns are fed into the FNN, effective learning behaviors are no longer applied in the FNN. Alternatively, the learning of spurious behaviors occurs in the FNN.

However, for pre-defining the on-road sensor configuration, the trial and error method is generally still used. Also, it is impractical to test all on-road sensor configurations, as it may involve a large amount of testing time. For example, when there are only 20 sensors on the freeway, 1048575 (=2^{20}-1) on-road sensor configurations are required to be tested. To pre-determine the on-road sensor configuration, a systematical and effective method, namely the Taguchi method [14, 15], is proposed. It has been widely used to reduce variation in the quality characteristics of products and improve manufacturing robustness. The operations of the Taguchi method for determining the on-road sensor configuration of FNN are detailed in the following section.

III. DETERMINATION OF ON-ROAD SENSOR CONFIGURATION USING TAGUCHI METHOD

A case study was conducted based on a real configuration of on-road sensors installed along the Mitchell Freeway, Western Australia, in order to illustrate the use of the Taguchi method for determining appropriate on-road sensor configuration of FNN.

A. Overview of the on-road sensor configuration

Traffic flow conditions were collected by 14 on-road sensors (D_1 to D_{14}) installed along the Mitchell Freeway, Western Australia, as illustrated in Figure 2. Three on-road sensors, namely D_1 to D_3, were installed at the off-ramp, at the on-ramp, and between the off-ramp and on-ramp, for the Reid Highway. For the other sections of the freeway, the on-road sensors were also installed at the off-ramp, at the on-ramp, and between the off-ramp and on-ramp of Karrinyup Road (D_6 to D_8), Cedric

Street (D_9 to D_{11}) and Hutton Street (D_{12} to D_{14}), respectively. For Erindale Road, two on-road sensors, namely D_4 and D_5, were installed at the off-ramp and near the on-ramp respectively. The sampling time used for capturing the traffic flow conditions for all on-road sensors was half minute (or 30 seconds). The beginning section, Reid Highway, and the end section, Hutton Street, are 7 kilometers apart, where the speed limit within all the sections along the freeway is 100 km/hour. Based on the captured past and current traffic flow conditions, the FNN was developed to forecast future traffic flow conditions with five sampling times ahead (or two and half minutes ahead).

The traffic flow data used for developing the FNN was collected in the 12-th week of 2009. This data was collected during the peak morning traffic period (7.30-9.30 am) on the five business days, Monday, Tuesday, Wednesday, Thursday and Friday. This data was divided into two data sets, training data and test data. The training data was the data collected from Monday to Thursday. It was used to train the FNN for traffic flow forecasting. The test set was the data collected for Friday. It was used to evaluate the generalization capability of the FNN in forecasting future traffic flow conditions.

As the sampling time of 30 seconds was used by the on-road sensors to capture traffic flow conditions, the total time used for capturing traffic flow conditions from Monday to Thursday is 480 minutes (i.e. 4 days x 2 hours). Hence, 960 pieces of traffic flow data were used as the training data. The total time used for capturing traffic flow conditions from Friday is 120 minutes (i.e. 1 days x 2 hours). Hence, 240 pieces of traffic flow data were used as the test data.

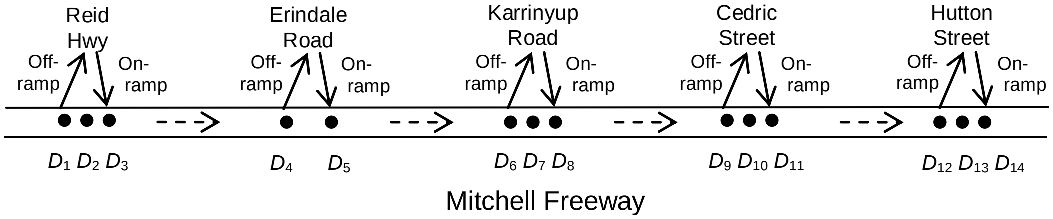

Fig. 2 On-road sensor configuration on the Mitchell Freeway

B. Trail design using orthogonal array

The first step in using the Taguchi method is to determine the design parameters which need to be studied. Then, an appropriate orthogonal array can be determined with respect to the design parameters.

Here, we intend to determine the appropriate on-road sensor configuration for the FNN. It needs to be determined which on-road sensors are required to connect to the FNN, and which on-road sensors are not required to connect to the FNN. As there are 14 on-road sensors installed along the freeway, 14 design parameters need to be studied. These design parameters are depicted in binary representation with either '0' or '1', where they determine the connection states of the on-road sensors. When the design parameter is '1', the corresponding on-road sensor is connected to the FNN and the traffic flow conditions captured by this on-road senor are fed into the FNN. When the design parameter is '0', the corresponding on-road sensor is disconnected from the FNN, and no traffic flow condition captured by this on-road sensor is fed into the FNN.

As this design problem is concerned with 14 design parameters and with 2 states (either ‘1’ or ‘0’), the orthogonal array $L_{20}\left(2^{14}\right)$ shown in Table 1, which has the same design platform, is used. Based on $L_{20}\left(2^{14}\right)$, only 20 main trials are required to be studied in $L_{20}\left(2^{14}\right)$, in order to determine the appropriate input configuration of the FNN which is engaged with 14 on-road sensors. Pairwise balancing property exists in the combinations of $L_{20}\left(2^{14}\right)$, whereby every state of a design parameter occurs the same number of times for all trials. It minimizes the number of required trails when the pairwise balancing property is retained.

For the first main trial, all on-road sensors are connected with the FNN. For the second main trial, the on-road sensors, D_2, D_5, D_6, D_7, D_8, D_{10} and D_{12} are connected with the FNN and the other on-road sensors are all disconnected.

If the full factorial design is used to study this design problem, which consists of 14 design parameters with each design parameters having 2 states, 16384 ($=2^{14}$) main trials are required to be conducted in order to determine the most appropriate input configuration for the FNN. When the orthogonal array, $L_{20}\left(2^{14}\right)$, is used, only 20 main trials are required to be conducted. Therefore, 16364 trials (= 16384 trials - 20 trials) can be saved by using the orthogonal array. Much less efforts are required by the using the orthogonal array, $L_{20}\left(2^{14}\right)$, than those required by the full factorial design.

C. Performance evaluation of each trial

The performance for the input configuration of the FNN needs to be evaluated by: (a) the accuracy of traffic flow forecasting, and (b) the robustness of traffic flow forecasting. To address these two objectives, equation (7) which evaluates the signal to noise ratio (SNR) is used:

$$\eta=-10\times\log_{10}\frac{1}{n-1}\sum_{i=1}^{n}\left(e_{MARE}^{i}-\mu\right)^{2} \tag{7}$$

where η is the SNR of the FNN for traffic flow forecasting; n is the number of trials with different initial values of FNN parameters; μ is the mean value of e_{MARE}^{i}; and e_{MARE}^{i} is defined by equation (6) which indicates the mean absolute error between the actual traffic flow conditions and the forecasts obtained by the FNN. All the algorithms and computation involved in this study were implemented using Matlab 7.7 in a PC which has a CPU of Intel(R) Core(TM)2 Duo 2.66GHz and a memory of 7.99GB.

Based on equation (7), two aspects by defining a type of signal-target problem [11] can be addressed. The robustness of the FNN, which is affected by the varieties of different initial values of FNN parameters, can be addressed as external to traffic flow forecasting. It evaluates the accuracy of the traffic flow forecasting by comparing the forecasts of the FNN and the actual traffic flow conditions. It also evaluates the level of robustness against the noise factors. If the SNR is larger, then the error between the actual traffic flow conditions and the forecasts is smaller, and the robustness of accuracy is larger. The trial with highest SNR demonstrates that the design parameters can represent the highest accuracy and the highest robustness.

D. Analytical results

Based on the orthogonal array, $L_{20}\left(2^{14}\right)$, 20 main trials representing by the 20 combinations of the on/off states of the 14 on-road sensors are conducted. Five results with respect to each main trial were collected as illustrated by e^{i}_{MARE}, with *i*=1, 2, ..., 5, in Table 2. These results were established by the initial FNN parameters, prior to each learning session. They indicate the generalization capabilities of the trained FNN as defined by equation (6). Then, the SNR were computed by using the equation (7), for each main trial of the orthogonal array $L_{20}\left(2^{14}\right)$. The compiled results for all trials are shown in Table 1.

Trials	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	**Number of connected on-road sensors**	e^{1}_{MARE}	e^{2}_{MARE}	e^{3}_{MARE}	e^{4}_{MARE}	e^{5}_{MARE}	**S/N**	**Rank of SNR**
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	14	9.03	9.11	8.89	8.99	8.56	46.90	12
2	0	1	0	0	1	1	1	1	0	1	0	1	0	0	7	9.22	9.23	9.18	8.61	8.97	42.79	16
3	0	0	1	0	0	1	1	1	1	0	1	0	1	0	7	8.87	9.16	9.18	9.20	8.89	52.09	5
4	1	0	0	1	0	0	1	1	1	1	0	1	0	1	8	9.23	8.95	9.02	8.87	9.29	50.28	9
5	1	1	0	0	1	0	0	1	1	1	1	0	1	0	8	9.02	8.91	8.90	8.76	8.85	62.92	1
6	0	1	1	0	0	1	0	0	1	1	1	1	0	1	8	9.11	9.07	9.08	8.88	8.80	55.29	3
7	0	0	1	1	0	0	1	0	0	1	1	1	1	0	7	8.54	8.95	9.05	8.88	9.25	42.90	15
8	0	0	0	1	1	0	0	1	0	0	1	1	1	1	7	9.13	9.30	8.99	9.32	9.20	56.23	2
9	0	0	0	0	1	1	0	0	1	0	0	1	1	1	6	8.89	9.05	9.10	8.93	8.66	51.25	7
10	1	0	0	0	0	1	1	0	0	1	0	0	1	1	6	8.56	8.76	8.85	8.86	9.18	46.00	14
11	0	1	0	0	0	0	1	1	0	0	1	0	0	1	5	9.31	8.86	9.39	8.60	9.42	36.11	19
12	1	0	1	0	0	0	0	1	1	0	0	1	0	0	5	8.96	9.22	8.74	9.04	9.16	49.27	10
13	0	1	0	1	0	0	0	0	1	1	0	0	1	0	5	8.57	8.88	8.89	8.83	9.01	52.53	4
14	1	0	1	0	1	0	0	0	0	1	1	0	0	1	6	9.49	9.33	8.84	8.47	9.10	34.18	20
15	1	1	0	1	0	1	0	0	0	0	1	1	0	0	6	8.71	9.07	9.14	8.80	9.50	39.48	17
16	1	1	1	0	1	0	1	0	0	0	0	1	1	0	7	9.12	8.99	8.76	9.11	8.65	47.11	11
17	1	1	1	1	0	1	0	1	0	0	0	0	1	1	8	8.60	8.97	9.02	8.72	8.79	50.91	8
18	0	1	1	1	1	0	1	0	1	0	0	0	0	1	7	8.86	9.25	8.87	8.84	8.93	51.63	6
19	0	0	1	1	1	1	0	1	0	1	0	0	0	0	6	8.88	8.83	9.12	9.51	8.55	36.46	18
20	1	0	0	1	1	1	1	0	1	0	1	0	0	0	7	9.12	9.41	9.21	8.85	8.96	46.59	13

Table 1 Orthogonal array, $L_{20}(2^{14})$, and trial results

By using the orthogonal array $L_{20}(2^{14})$, 20 main trials, with each main trial consisting of 5 trials, were conducted. Thus, a total of 100 trials needed to be conducted. If the full factorial design is used, a total of 81920 trials (i.e. 16384 ($=2^{14}$) main trials consisting of 5 trials) need to be conducted. When 30 seconds are required for each trial, 2457600 seconds (i.e. 40960 minutes, 682.7 hours or 28.4 days) are required for the full factorial design. Requiring 28.4 days to design a predictor for traffic flow forecasting is not practical. When the Taguchi method is used, only 3000 seconds (i.e. 100 trials, 50 minutes or 0.83 hour) are required. Hence, 681.9 hours (i.e. 28.4 days) can be saved by using the Taguchi method. Therefore, a significant amount of computational effort and time can be saved. This demonstrates the effectiveness of using the Taguchi method in determining appropriate on-road sensor configurations for traffic flow forecasting.

Also, Table 1 shows the SNR of the 5 trials with respect to the 20 main trials, and the number of on-road sensors connected to the FNN. It shows that the 5th trial which is engaged with the eight on-road sensors can achieve the smallest SNR. It is smaller than that achieved by the 1st trial, involving all the 14 sensors. These results show clearly that we should not simply use all the sensors to develop the FNN for traffic flow forecasting, as this might not achieve the best forecasting result. Also, someone may assume that it is necessary to include the traffic flow conditions captured by the on-road sensors that are located at or near the forecasting point. However, these results show that this is not the case. The SNR obtained by the 5th trial are smaller than those achieved by the 8th and the 9th trials, where the on-road sensors located at the forecasting points are connected in the 8th and the 9th trials, but those in the 5th trial are not connected. Therefore, it is necessary to use appropriate on-road sensor configuration to design the FNN, in order to obtain satisfactory traffic flow forecasting.

Since the combinations of the design parameters of each trial are orthogonal, it can separate out the main effect of each design parameter [1, 11]. The main effects of each design factor at each of the two levels are calculated and shown in Table 2. They are calculated by taking the average from Table 1 for a design parameter at a given level. As an example, the on-road sensor D_3 is connected with the FNN for trials 1, 3, 6, 7, 12, 14, 16, 17, 18 and 19. The average of the corresponding traffic flow condition is 46.67, which is shown in the response table that the on-road sensor D_3 is connected.

The sensitivity of each design factor is computed by taking the difference between the largest and smallest main effect for a given design factor. It reveals that the on-road sensor, D_4, shows the greatest sensitivity. This means that the one which has the largest effect on the FNN is realized by varying the on-road sensor, D_4. Similarly, the on-road sensors, D_1, D_2, D_3 and D_7, show the least sensitivity to the FNN. Hence, their effect on the FNN is little whenever they are connected with or disconnected from the FNN. The main effects of all on-road sensors are also shown graphically in Figure 3. Graphing the main effects of all on-road sensors can provide more insight at a glance, and it clearly shows that the on-road sensor, D_4, has much greater sensitivity than do all of the other on-road sensors.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}
Level 1 (connected)	47.36	48.56	46.67	47.39	47.61	46.77	46.24	48.39	51.87	47.02	47.27	48.15	50.88	47.88
Level 0 (disconnected)	46.35	53.22	51.33	33.54	39.23	43.63	45.23	53.05	50.86	51.68	46.26	52.81	55.54	52.54
Sensitivity	1.01	4.66	4.66	13.85	8.38	3.14	1.01	4.66	1.01	4.66	1.01	4.66	4.66	4.66

Table 2 Main effects of SNR of each on-road sensor

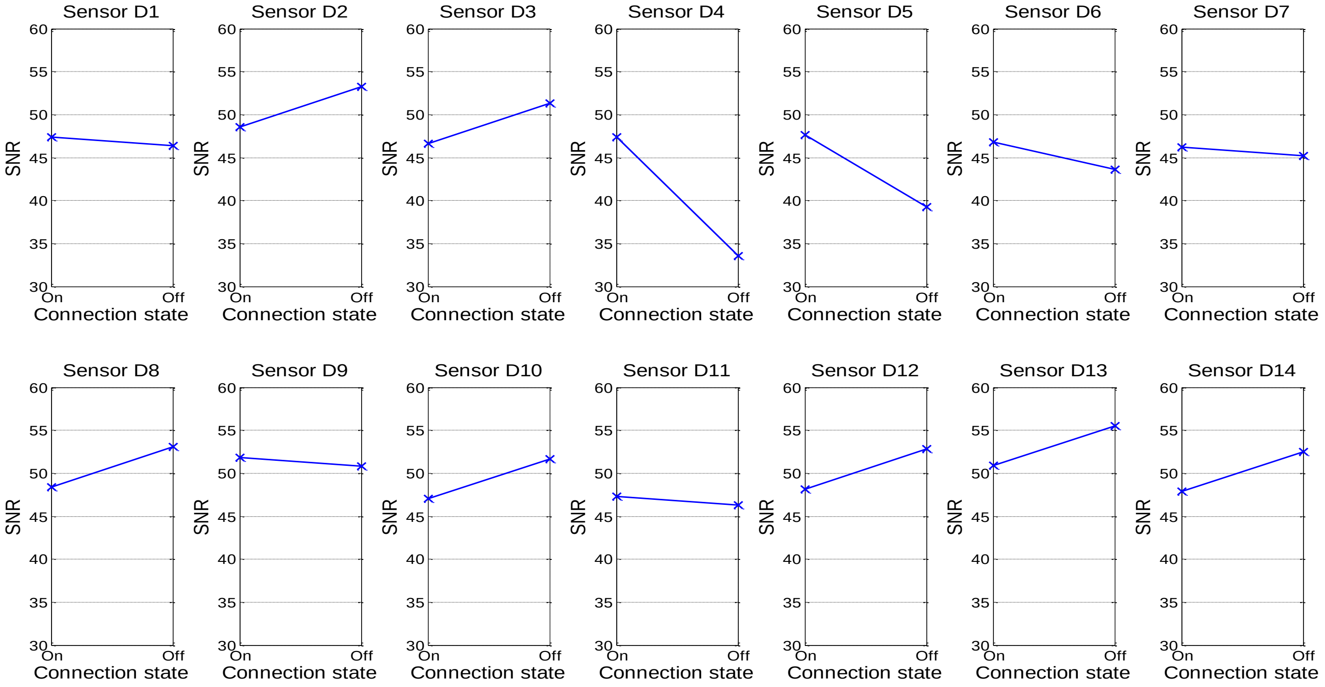

Figure 3 Main effects of SNR of each on-road sensor

The larger main effects of each on-road sensor are underlined in Table 3, i.e. D_1 is connected; D_2 is disconnected; D_3 is disconnected; D_4 is connected; D_5 is connected; D_6 is connected; D_7 is connected; D_8 is disconnected; D_9 is connected; D_{10} is disconnected; D_{11} is connected; D_{12} is disconnected; D_{13} is disconnected; and D_{14} is disconnected. Based on the connection states of the on-road sensors with larger SNR, the FNN with window size 1 formulated in equation (5) can be developed to represent the traffic flow predictor, where the FNN parameters were determined based on the training data collected from Monday to Thursday, week 12, 2009. Figure 4 shows the simulation result obtained by the FNN for forecasting traffic flow conditions based on the test data collected on Friday, week 12, 2009. It can be seen that the forecasting result is close to the actual traffic flow data. The accuracy in terms of traffic flow forecasting is 94.70%, which is considered to be satisfactory.

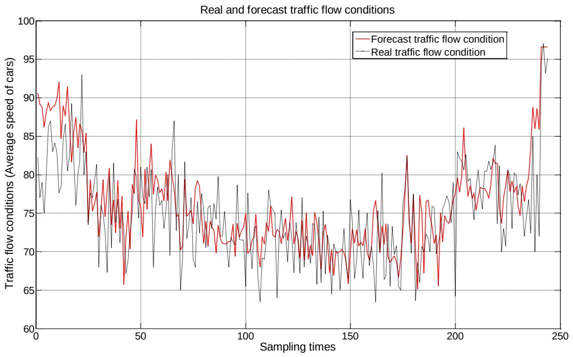

Figure 4 Forecasting of traffic flow condition

In this case study, only 7 on-road sensors are required to be connected to the FNN in order to produce satisfactory traffic flow forecasting results. Hence, the installation cost for 7 on-road sensors is less than that of 14 on-road sensors. When two ILDs are used on each on-road sensor, and each ILD costs around USD 800, USD 11,200 (= 14 x USD 800) are saved on installation of on-road sensors in this section of road [23]. When more accuracy detectors such as acoustic array detectors and video image processors are used as the on-road sensors, there is further cost-saving, as accuracy detectors cost more. Also, the chance of producing faulty forecasting is reduced, as the chance of damaging any one of the 7 on-road sensors is less than the chance of damaging any one of the 14 on-road sensors. It can also save the effort and cost of maintaining the on-road sensor networks, as only 7 on-road sensors, not all 14 of them, need to be maintained.

E. Results using different training sets

Section III.D shows the results of the FNN which is developed based on the training data captured from Monday to Thursday (four days), and the test data captured on Friday. Hence, four days of data were used for training, and one day of data was used for testing. In this section, we evaluated the forecasting accuracies when a smaller amount of training data is used for developing the FNN. We used the data captured from Tuesday to Thursday (three days), from Wednesday to Thursday (two days) and from Thursday (one day) as three sets of training data. We repeated the procedures of using the Taguchi method to generate FNN based on these three sets of training data. Figure 5 illustrates that using the training data captured from the four days can develop better FNN for forecasting than that developed based on the data of three days, which can only obtain 92.13% accuracy. Those developed based on the training data captured on two days or one day are 88.56% and 80.09% respectively. The results illustrate that using more training data can generate a more accurate FNN for traffic flow forecasting in this case study.

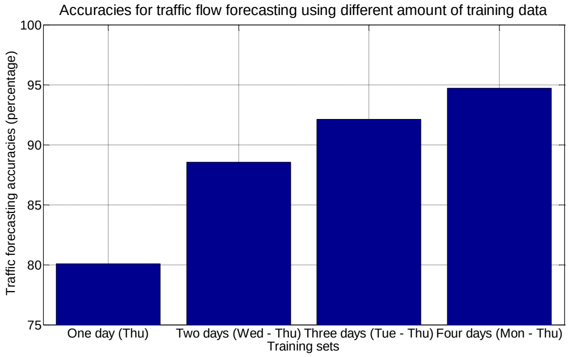

Fig. 5 Accuracies for traffic flow forecasting using different training sets

F. Comparisons with stochastic algorithms

The two approaches, namely genetic algorithms (GA) [8] and particle swarm optimization (PSO) [28], which have been used for determining appropriate FNNs, were employed as a comparison.

In the GA [8]**,** a population of chromosomes in binary representation is first generated randomly as illustrated in Figure 6. Then, a FNN is developed with respect to the input configuration which is represented by the chromosome, where the FNN is generated using Jang's algorithm [6]. The chromosomes of the GA are evaluated based on a fitness function which is identified to equation (6) intending to measure the differences between the forecasts of the FNN and the actual traffic flow conditions. Hence, the fitness of the chromosome indicates the generalization capability of the FNN represented by the chromosome. When the chromosome has better fitness, the generalization capability of the FNN is better and more accurate forecasting of traffic flow conditions can be produced by the FNN.

After evaluating all chromosomes based on the fitness function, the evolutionary operations including selection, crossover and mutation proposed in [8], are performed in order to generate a new population of chromosomes, where the new chromosomes intend to create a better input configuration which can produce a better FNN than those created by the old chromosomes. The evolutionary operations of the GA are repeated until the pre-defined number of generations is reached. The detailed operations of the GA can be referenced in [8].

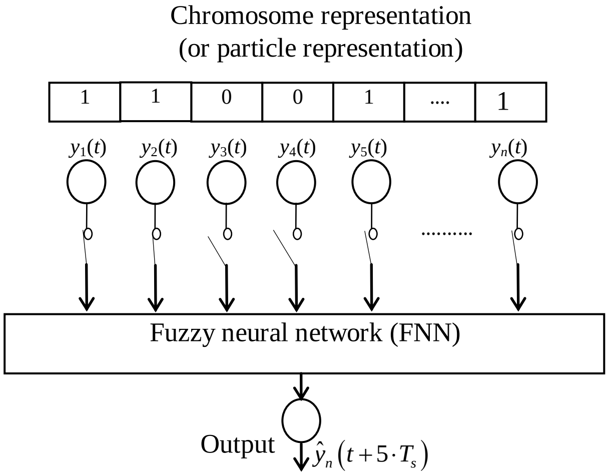
Chromosome representation
(or particle representation)
1
1
0
0
1
....
1
$y_1(t)$
$y_2(t)$
$y_3(t)$
$y_4(t)$
$y_5(t)$
$y_n(t)$
....
Fuzzy neural network (FNN)
Output
$\hat{y}_n(t+5\cdot T_s)$

Fig. 6 On-road sensor configuration of the FNN model represented in the GA and the PSO

To perform the test, the two GAs, namely GA-5-20 and GA-5-200, were used. The following parameters were used in both GA-5-20 and GA-5-200: crossover rate = 0.8; mutation rate = 0.1; and population size = 5. In GA-5-20, the pre-defined number of generations used was 20. Hence, the total computational evaluations used were 100, which were those used in the Taguchi method. This setting was established to evaluate the performance difference between the genetic algorithm and the Taguchi method, when the same computational effort was used. In GA-5-200, the pre-defined number of generations was set to 200. Hence, there were 1000 computational evaluations used in the GA-5-200, where the number of computational evaluations used by the GA-5-200 was 10 times more than those used by the Taguchi method. This setting of genetic algorithm intends to evaluate whether the GA-5-200 can obtain significantly better performance than the Taguchi method, when much more computational efforts are involved.

In the PSO [28] approach, a swarm of particles is generated in a binary representation, which can be illustrated in Figure 6, where the parameters of the FNN models are determined by using the Jang's algorithm [6]. Similar to the chromosome evaluations of the GA, each particle is evaluated based on equation (6) with respect to the input configuration specified by the particle. Hence, a better particle can produce a FNN which can more accurately forecast traffic flow conditions. After evaluating the particles, the positions and velocities of the particles in the swarm are updated based on the position of the particle and the global best position of the swarm found so far. The movement of the swarm continues, until the pre-defined number of generations has been reached. The detailed operations of the PSO can be referenced in [28].

The two PSO, namely PSO-5-20 and PSO-5-200, were used. The following PSO parameters were used in the two PSO: the maximum and minimum inertia weights were set to 0.9 and 0.2, respectively; the swarm sizes for both PSO were 5; and the initial acceleration coefficients were set to 2. In PSO-5-20, 100 computational evaluations were used in each run, which was the same as those used in the Taguchi method. By doing this, we can evaluate the performance of the PSO and the Taguchi method when the same amount of computational effort is used. In PSO-5-200, the pre-defined number of generations was set to 200. Hence, 1000 computational evaluations were used for each run, which were ten times more than those used in the Taguchi method. This allows us to determine whether the PSO can outperform the Taguchi method, when more computational efforts are used.

As both the GA and the PSO are stochastic algorithms, different results are found with different runs. Therefore, all the algorithms, GA-5-20, GA-5-200, PSO-5-20 and PSO-5-200, were run 30 times, and the results of the 30 runs were recorded. Results in terms of traffic forecasting accuracies obtained by all methods and the computational times used for all methods are shown in Figure 7 and Figure 8, respectively. Figure 6 shows that the results obtained by both GA-5-20 and PSO-5-20 are poorer than those obtained by the Taguchi method, where the computational efforts used in the three methods, GA-5-200 and PSO-5-200 and the Taguchi method, were identical, as shown in Figure 8. Figure 7 shows that the results obtained by the Taguchi method are slightly poorer than the results obtained by the GA-5-200 and PSO-5-200, while the computational efforts used by both PSO-5-200 and GA-5-200 were significantly larger than those used by the Taguchi method.

These results demonstrate that both the GA and the PSO can achieve slightly better results than the Taguchi method in terms of traffic flow forecasting accuracies, when a significantly larger amount of extra computational effort was used in both the GA and the PSO. However, when the same computational efforts were used in the three methods, the Taguchi method can obtain better results in term of traffic flow forecasting than those obtained by both the GA and the PSO. Therefore, the Taguchi method is more effective than the two methods in searching for the appropriate input node configurations of the FNN models for traffic flow forecasting.

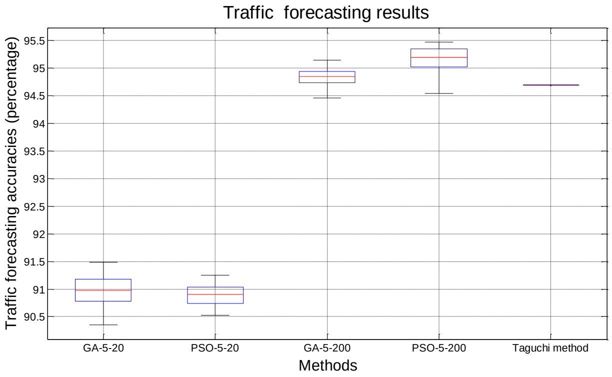

Fig. 7 Traffic forecasting accuracies obtained by the methods

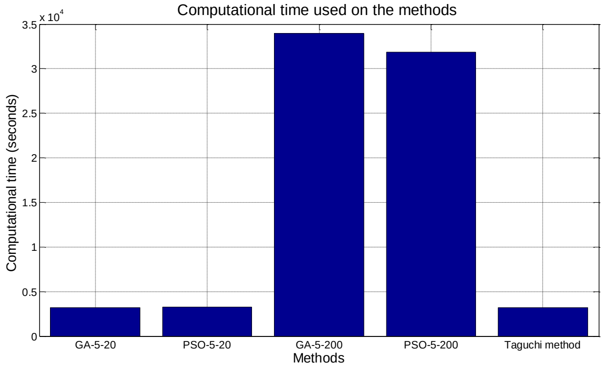

Fig. 8 Computational times used by the methods

As GA-5-20, GA-5-200, PSO-5-20 and PSO-5-200 are all stochastic algorithms, different on-road sensor configurations can be generated with different runs. Figure 9 shows the number of on-road sensors involved in each on-road sensor configuration generated by the four stochastic algorithms. For GA-5-20 and PSO-5-20, the numbers of on-road sensors involved in those configurations are mostly between eight to eleven, and between eight to ten, respectively. For GA-5-200 and PSO-5-200, the numbers of on-road sensors are mostly between seven to eight, and between six to eight, respectively, which are close to the one generated by the Taguchi method. Hence, this further demonstrates that it is not necessary to use all the on-road sensors for traffic flow forecasting in this section of freeway. As the Taguchi method is a deterministic method, the same on-road sensor configuration can be generated with different runs where the number of on-road sensors is seven. Hence, the same number of on-road sensors can be generated by the Taguchi method in different runs, while different numbers of on-road sensors are generated by the stochastic methods in different runs. The difference between the Taguchi method and the stochastic methods can be further demonstrated.

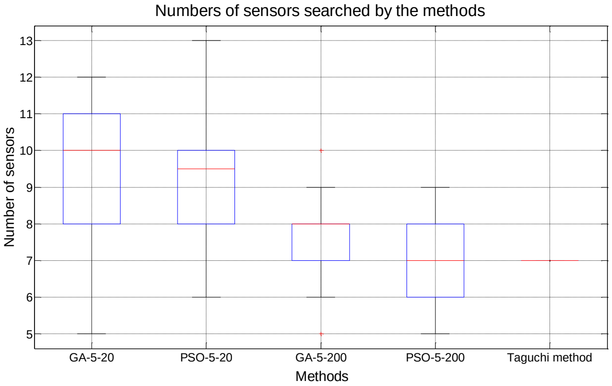

Figure 9 Number of on-road sensors searched by the methods

IV. Discussion and conclusion

In this paper, the Taguchi method, was proposed to determine appropriate on-road sensor configuration for fuzzy neural networks in order to forecast traffic flow. As the number of on-road sensors installed on the freeway is huge, it is impossible to test all individual configurations in order to determine the optimal one. The effectiveness of the Taguchi method was demonstrated by a case study intended to perform traffic flow forecasting based on fuzzy neural networks, where the traffic flow data was captured by fourteen on-road sensors installed on a section of freeway in Western Australia. Two advantages of using the Taguchi method in terms of forecasting accuracies and development time were demonstrated. Results show that the Taguchi method can provide a systematic and efficient methodology to determine the appropriate on-road sensor configurations with far less development time than that required for full factorial design. It can generate fuzzy neural networks with better traffic flow forecasting than the other existing methods including the evolutionary algorithm and the particle swarm optimization, when the same development time was used.

The Taguchi method can be further extended by the following research, which is related to the on-road sensors and the other applications:

Moving window of traffic flow predictor: we can further refine the prediction accuracy by moving windows of the time sequences, which are used for traffic flow forecasting. The exclusion of too many time sequences might exclude important traffic flow patterns, and inclusion of too many time sequences might elicit less influential patterns. Hence, finding the proper window size is often a critical balancing act. We can vary the window sizes and measure the forecasting errors based on trial and error, until the window size with the best forecasting performance is selected.

The Taguchi method, which is a more systematic approach, can be applied to determine window size in order to further refine the prediction performance. The dimension of the orthogonal array can be selected based on the number of significant on-road sensors, and the number of levels of the orthogonal array can be defined as the window sizes. The procedures discussed in this paper can be reused in order to determine the appropriate window size. Hence, the prediction accuracy can be further refined.

Determination of appropriate sensor configuration for other applications: This paper presents a mechanism that uses the Taguchi method to design on-road sensor configurations for traffic flow forecasting. Apart from the design of on-road sensor configurations, the Taguchi method can also be applied on other configuration designs which involve many sensors. For example, the Google map is produced by a set of location sensors, where determination of an appropriate configuration of location sensors is essential in order to produce a more informative map. The Taguchi method can be applied to determine appropriate configurations of the location sensors in order to create a better appreciation of the Google map.

Acknowledgement

The authors wish to express their sincere thanks to Elizabeth Chang and Jaipal Singh for many useful discussions and valuable suggestions. They would also like to acknowledge their very useful comments for this research.

References

1. D.M. Byrne and S. Taguchi, “The Taguchi approach to parameter design”, ASQC Quality Congress, Anaheim, CA, p. 168-177, 1986.
2. C.Y. Chang and D.R. Chen, “Active noise cancellation without secondary path identification by using an adaptive genetic algorithm”, *IEEE Transactions on Instrumentation and measurement*, vol. 59, no. 9, pp. 2315-2327, 2010.
3. G.A. Davis, and N.L. Nihan, “Nonparametric regression and short-term freeway traffic forecasting”, *Journal of Transportation Engineering*, vol. 177, no. 2, pp. 178–188, 1991.
4. Y. Gao and M.J. Er, “NARMAX time series model prediction: feedforward and recurrent fuzzy neural network approaches”, *Fuzzy Sets and Systems*, vol. 150, pp. 331-350, 2005.
5. B. Gunter, “A perspective on the Taguchi methods”, *Quality Progress*, pp. 44-52, 1987.
6. J.S.R. Jang, “ANFIS: adaptive-network-based fuzzy inference system”, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, 665-685, 1993.
7. K.E. Konstantinos and M.N. Vrahatis, “On the computation of all global minimizers through particle swarm optimization”, *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211-224, 2004.
8. K.F. Leung, F.H.F. Leung, H.K. Lam and S.H. Ling, “On interpretation of graffiti digits and characters for eBooks: neural fuzzy network and genetic algorithm approach”, *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 464-471, 2004.
9. K.F. Man, K.S. Tang and S. Kwong, “Genetic algorithms: concepts and applications”, *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 519-534, 1996.
10. R.J. Mayer and P.C. Benjamin, “Using the Taguchi paradigm for manufacturing system design using simulation experiments”, *Industrial Engineering*, vol. 22, no. 2 (1992) 195-209.
11. M.S. Phadke, Quality Engineering Using Robust Design, (Prentice Hall, 1989).
12. C. Quek, M. Pasquier and B.B.S. Lim, “POP-TRAFFIC: A Novel Fuzzy Neural Approach to Road Traffic Analysis and Prediction”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 133-146, 2006.
13. L.P. Sullivan, “The power of Taguchi methods”, *Quality Progress*, pp. 76-79, 1987.
14. G. Taguchi (1989) Introduction to quality engineering. Asian Productivity Organization, Tokyo.
15. G. Lachtermacher and J.D. Fuller, “Backpropagation in time-series forecasting”, *Journal of Forecasting*, vol. 14, pp. 381-93, 1995.
16. C.C. Lai and Y.C. Chen, “A user-oriented image retrieval system based on interactive genetic algorithm”, *IEEE Transactions on Instrumentation and measurement*, vol. 60, no. 10, pp. 3318-3325, 2011.
17. H.A. Nguyen, H. Guo and K.S. Low, “Real-time estimation of sensor node’s position using particle swarm optimization with log-barrier

constraint", *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3619-2628, 2011.

18. S.K. Oh, W. Pedrycz and H.S. Park, "Fuzzy relation based neural networks and their hybrid identification", *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 6, pp. 2522-2537, 2007.
19. I. Okutani and Y.J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory", *Transportation Research, Part B: Methodology*, vol. 18, no. 1, pp. 1–11, 1984.
20. P. Ross, "Exponential filtering of traffic data", Transportation Research Record, vol. 869, *Transportation Research Board*, Washington, D.C., pp. 43–49, 1982.
21. N.K. Rout, D.P. Das and G. Panda, "Particle swarm optimization based active noise control algorithm without secondary path identification", *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, pp. 554-562, 2012.
22. B.L. Smith, and M.J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches", *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 261–266, 1997.
23. U.S. Department of Transportation, Chapter1, Traffic Detector Handbook: Third Edition – Volume 1, U.S. Department of Transportation, Federal Highway Administration, May 2006.
24. A.R. Varkonyi-Koczy and B. Tusor, "Human-computer interaction for smart environment applications using fuzzy hand posture and gesture models", *IEEE Transactions on Instrumentation and measurement*, vol. 60, no. 5, pp. 1505-1514, 2011.
25. Y. Wang, M. Papageorgiou and A. Messmer, "A real-time freeway network traffic surveillance tool", *IEEE Transactions on Control Systems Technology*, vol. 14, no. 1, 18-32, 2006.
26. B.M. Williams, P.K. Durvasula, D.E. Brown, "Urban Freeway Traffic Flow Prediction: Application of Seasonal Autoregressive Integrated Moving Average and Exponential Smoothing Models", *Transportation Research Record*, no. 1644, pp. 132-141, 1998.
27. H. Yin, S.C. Wong, J. Xu and C.K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach", *Transportation Research Part C*, vol. 10, pp. 85-98, 2002.
28. J. Yu, S. Wang and L. Xi, "Evolving artificial neural networks using an improved PSO and DPSO", *Neurocomputing*, vol. 71, pp. 1054-1060, 2008.
29. G.P. Zhang, B.E. Patuwo and H.Y. Michael, "A simulation study of artificial neural networks for nonlinear time-series forecasting", *Computers and Operations Research*, vol. 28, pp. 381-396, 2001.

Kit Yan Chan received his MPhil degree in Electronic Engineering from City University of Hong Kong, Hong Kong in 2003 and his PhD degree in Computing from London South Bank University, United Kingdom in 2006. After his PhD study, he worked as a Postdoctoral Research Fellow in the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, until 2009. Currently, he is a Senior Research Fellow in the Department of Electrical and Computer Engineering, Curtin University, Australia. He has published 39 scientific papers in international journals and 2 research monographs. His research interests include computational intelligence and its applications to new product design, manufacturing process design, speech recognition and traffic flow forecasting.

Tharam Dillon is a Fellow of the IEEE, ACS and IE(Aust). He is head of the IFIP International Task Force WG2.12/24 on Semantic Web and Web Semantics, the IEEE/IES Technical Committee on Industrial Informatics, and the IFIP Technical Committee 12 on Artificial Intelligence. He has published more than 750 papers published in international conferences and journals and is the author of 6 books and has another 5 edited books. His current research interests include Web semantics, ontologies, Internet computing, e-commerce, hybrid neurosymbolic systems, neural nets, software engineering, database systems, and data mining. He is the Editor-in-Chief of the International Journal of Computer Systems Science and Engineering as well as the Engineering Intelligent Systems. He is the Co-Editor of the Journal of Electrical Power and Energy Systems.