A Prediction Model Based on Time Series Data in Intelligent Transportation System

Jun Wu[1,2], Luo Zhong[2], Lingli Li[2], and Aiyan Lu[2]

[1] School of Computer Science, Hubei University of Technology,
1 Lijiadun Road, Wuhan, Hubei, P.R. China
[2] School of Computer Science and Technology, Wuhan University of Technology,
122 Luoshi Road, Wuhan, Hubei, P.R. China

Abstract. Intelligent Transportation System has a new kind of complicated time series data which would be the traffic flow, average speed or some other traffic condition information at the same time period. All above data is useful and important for our traffic system which includes the traffic flow prediction, tendency analysis or cluster. With the development in time series analysis model and their applications, it is important to focus on how to find the useful and real-time traffic information from the Intelligent Transportation System. Using this method of building models for the Intelligent Transportation System is the way to solve the traffic prediction problem and make control of the massive traffic network.

Keywords: Time Series Data, Prediction Model, Data Mining, Intelligent Transportation System.

1 Introduction

Intelligent Transportation System (ITS) [1] is becoming a very significant enabling technology in many sectors. Recent advances in hardware development have enabled the creation of widespread traffic networks. Currently, there are many ongoing projects that use Intelligent Transportation System (ITS) for traffic environmental monitoring and data acquisition applications, such as wildlife tracking, habitat monitoring, and building monitoring. Intelligent Transportation System establishes a large-scale, in real-time, accurate and efficient transport management system to integrate people, vehicles and traffic road unified closely. But there are still some difficult problems during the development, such as limited storage, low network bandwidth, poor inter-node communication, limited computational ability, and low power capacity still persisting. Several techniques [2-5] have been proposed to alleviate the problem of limited power at the network level and at the data management level. Another methods [6,7,8] at the data management level is in-network query processing or aggregation.

With in-network aggregation[9,10], a part of the computational work of the aggregation is performed within the sensor node before it sends the results out to the network. It can be easily illustrated by the following simple example of a sensor network used to monitor the average or the maximum temperature in a building [11,12]. The default way to implement this is to have each sensor send its temperature

Y. Yang, M. Ma, and B. Liu (Eds.): ICICA 2013, Part II, CCIS 392, pp. 420–429, 2013.
© Springer-Verlag Berlin Heidelberg 2013

reading up the network to the base station, with intermediate nodes responsible for just routing packets. In network aggregation, communication among sensor nodes is structured as a (routing) tree with the base station as its root. In this scheme, each node would incorporate its own reading together with the average computed so far by its children. As such, only one packet needs to be sent per node and each intermediate node computes the new average temperature before sending information further up the network. As a result, being able to transmit less data (because of aggregation instead of having to forward all the packets) will reduce energy consumption at the sensor nodes. However the urban traffic develop rapidly, we could not only rely on the old Intelligent Transportation System, but also need to build some useful models to adapt to our urban traffic development to improve the efficient use of traffic data.

This paper would focus on the topic of the prediction of daily traffic information which has characteristics of time series. The majority of this paper will be concerned with discussions of this point which focus on the time series analysis in prediction model. Firstly we discuss about the time series character and the data in Intelligent Transportation System (ITS), and it is easy to find their similarity. Furthermore it is necessary to find the corresponding model for our Intelligent Transportation System (ITS) which could adapt to the traffic development and especially to solve how to use this time series traffic data to make the prediction with our complicated traffic data. In this paper, we have tried two models, one is the Autoregressive integrated moving average (ARIMA) model, and the other is Generalized Regression Neural Network (GRNN) model. At the end, from comparison the prediction result between these two models we found the better model for our traffic time series data.

2 Time Series Data

Business, economic, engineering and environmental data are often collected in roughly equally spaced time intervals, for example, hour, week, month, or quarter. In many problems, such time series data may be available on several related variables of interest.

A time series system is a collection of a space of input series, a space of output series, and an operation carrying an input series into an output series.

Suppose $X(t) \qquad (t=0,\pm1,...)$ denotes an input series and $Y(t) \qquad (t=0,\pm1,...)$ the corresponding output series.

Then a common time series system has the form as following:

$$Y(t)=\mu+\sum_{u=-\infty}^{\infty} a(t-u)X(u)+\varepsilon(t) \qquad (t=0,\pm1,...)$$

For some sequence of filter coefficients $a(u) \qquad (u=0,\pm1,...)$, for some constant, and for some zero mean noise series $\varepsilon(t) \qquad (t=0,\pm1,...)$.

This problem of time series system identification is that of determining characteristics of the system from corresponding stretches of input and output series, as the following definition:

$$\{X(t),Y(t)\} \qquad (t=0,...,T-1)$$

Firstly suppose that the series $X(.)$ and $\varepsilon(.)$ are stationary and independent.

Secondly Let $c_{XY}(u) = \operatorname{cov}\{X(t+u), Y(t)\}$

Denote the cross covariance function of the two series and let the auto covariance functions $c_{XX}(u)$ and $c_{YY}(u)$ be defined similarly.

Thirdly the time series system leads to the relationship $c_{XY}(u) = \sum_v a(v) c_{XX}(v+u)$

For suitable $a(.)$

Let $f_{XY}(\lambda) = (2\pi)^{-1} \sum_u c_{XY}(u) e^{-iu\lambda}$

Denote the cross-spectrum of the series $X(.)$ with the series $Y(.)$ and make corresponding definitions of the power spectra $f_{XX}(\lambda)$, $f_{YY}(\lambda)$

Let $A(\lambda) = \sum_u a(u) e^{-iu\lambda}$, denote the transfer function of the filter.

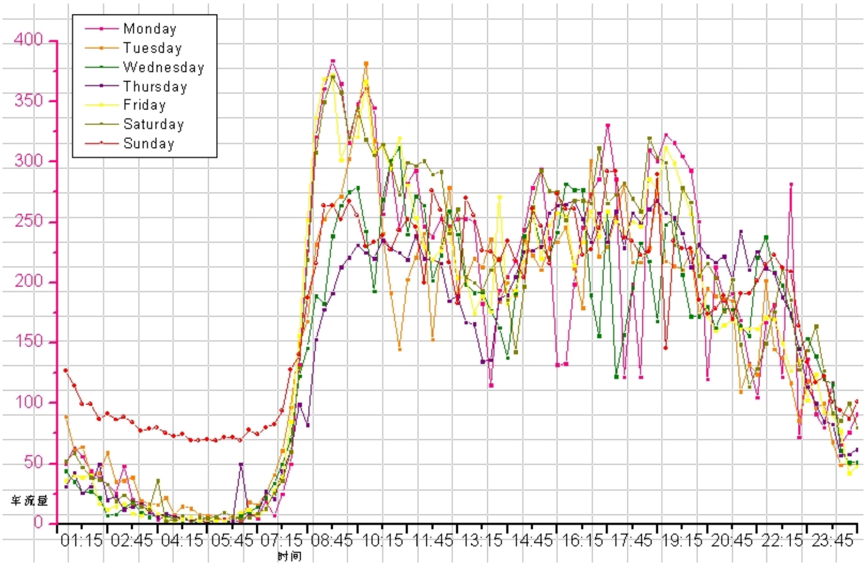

Fig. 1. Daily traffic time series sample from Monday to Sunday

Then the relationship leads to $f_{XY}(\lambda) = A(-\lambda) f_{XX}(\lambda)$ or, if $f_{XX}(\lambda) \neq 0$, to $A(\lambda) = f_{YX}(\lambda)\{f_{XX}(\lambda)\}^{-1}$.

The parameter $f_{YX}(\lambda)\{f_{XX}(\lambda)\}^{-1}$ is called the complex regression coefficient of the series $Y(.)$ on the series $X(.)$ at frequency λ. It provides the transfer function of the best linear filter for predicting the series $Y(.)$ from the series $X(.)$ [13].

The time series [14] is a stochastic process that varies over time, usually observed at fixed intervals. There are a lot of information which belong to time series, for example daily temperature, rainfall, monthly unemployment levels, the annual income and the traffic data. All above are the typical examples of the time series. Let’s have a look of our traffic time series sample which be shown in Fig 1, and Fig 2. This is the weekly example which includes 288*7 points for every 5 min to get.

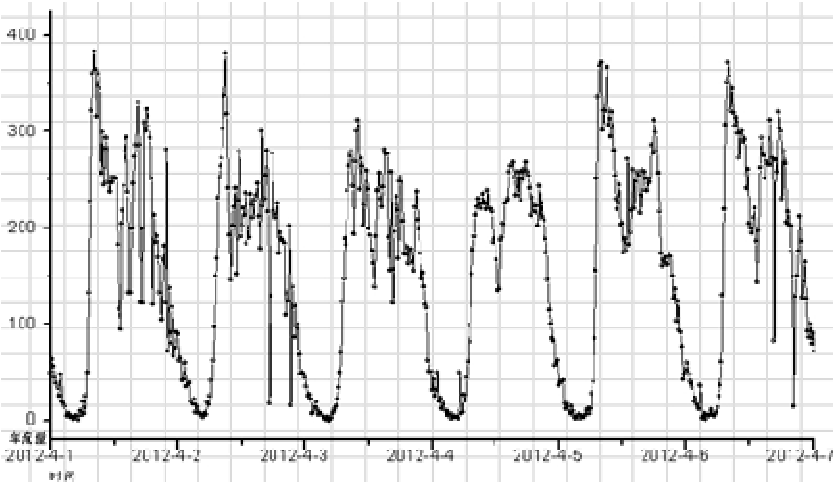

Fig. 2. Weekly traffic time series sample from the Intelligent Traffic System

As far as time series data are concerned, distinctions can be made as to whether the data are discrete-valued or real-valued, uniformly or non-uniformly sampled, uni-variate or multivariate, and whether data series are of equal of unequal length. Non-uniformly sampled data must be converted into uniformed data before clustering operations can be performed. This can be achieved by a wide range of methods, from simple down sampling based on the toughest sampling interval to a sophisticated model and estimation approach.

From the view of data mining, it is possible to get these assortments which include Cluster, Classification and Prediction with the time series data. Clustering is necessary when no labeled data are available regardless of whether the data are binary, categorical, numerical, interval, ordinal, relational, textual, spatial, temporal, image, multimedia, or mixtures of the above data types. The goal of clustering is to identify structure in an unlabeled data set by objectively organizing data into homogeneous groups where the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized. A hierarchical clustering method works by grouping data objects into a tree of clusters. There are generally two types of hierarchical clustering methods: agglomerative and divisive. Agglomerative methods start by placing each object in its own cluster and then merge clusters into larger and larger clusters, until all objects are in a single cluster or until certain termination conditions such as the desired number of clusters are satisfied. Divisive methods do just the opposite. A pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed.

Model-based methods assume a model for each of the clusters and attempt to best fit the data to the assumed model. There are two major approaches of model-based methods: statistical approach and neural network approach. An example of statistical approach is Auto Class, which uses Bayesian statistical analysis to estimate the number of clusters. Two prominent methods of the neural network approach to clustering are competitive learning, including ART and self-organizing feature maps. Unlike static data, the time series of a feature comprise values changed with time. Time series data are of interest because of its pervasiveness in various areas ranging from science, engineering, business, finance, economic, health care, to government. Given a set of unlabeled time series, it is often desirable to determine groups of similar time series. These unlabeled time series could be monitoring data collected during different periods from a particular process or from more than one process. The process could be natural, biological, business, or engineered. Works devoting to the cluster analysis of time series are relatively scant compared with those focusing on static data. However, there seems to be a trend of increased activity. From all above analysis, it is easy to find two of the reasons for analyzing and modeling such special series jointly are:

(1) To understand the dynamic relationships among them. They may be contemporaneously related, one series may lead the others or there may be feedback relationships.

(2) To improve accuracy of predictor. When there is information on one series contained in the historical data of another, better predictor can result when the series are modeled jointly.

3 Prediction Model

3.1 ARIMA Model

The Autoregressive integrated moving average (ARIMA) [15] model is a precise forecasting model for short time periods. However, in our society today, due to factors of uncertainty from the integral environment and rapid development of new technology, we usually have to forecast future situations using little data in a short span of time.

Step1. The ARIMA (p,d,q) model is described by parameters in Eq. (1) to (4):

$$\tilde{\Phi}_{\mathrm{p}}(B)W_t = \tilde{\theta}_q(B)a_t \qquad (1);$$

$$W_t = (1-B)^d(Z_t - \mu) \qquad (2);$$

$$\tilde{W}_t = \tilde{\varphi}_1 W_{t-1} + \tilde{\varphi}_2 W_{t-2} + \ldots + \tilde{\varphi}_p W_{t-p} + a_t - \tilde{\theta}_1 a_{t-1} - \tilde{\theta}_2 a_{t-2} - \ldots - \tilde{\theta}_1 a_{t-q} \qquad (3);$$

Where Z_t are observations, $\tilde{\varphi}_1,\ldots,\tilde{\varphi}_p$ and $\tilde{\theta}_1,\ldots,\tilde{\theta}_q$ are fuzzy numbers in the model, so that we could modify this Equation:

$$\tilde{W}_t = \tilde{\beta}_1 W_{t-1} + \tilde{\beta}_2 W_{t-2} + \ldots + \tilde{\beta}_p W_{t-p} + a_t - \tilde{\beta}_{p+1} a_{t-1} - \tilde{\beta}_{p+2} a_{t-2} - \ldots - \tilde{\beta}_{p+\mathrm{q}} a_{t-q} \quad (4);$$

Step2. A general ARIMA formulation model is selected to the traffic data in Intelligent Transportation System (ITS). This selection is of the main characteristics of the 5min traffic time series.

Step3. A model is identified for the chosen data

Step4. The parameters of this model are estimated. The optimum solution of the parameter $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_{p+q}^*)$ and the residual s_{a_t}.

Then it is easy to get the fuzzy ARIMA model and its new parameters which shown as Eq.(5):

$$\tilde{W}_t = \langle \alpha_1, c_1 \rangle W_{t-1} + ... + \langle \alpha_p, c_p \rangle W_{t-p} + a_t - \langle \alpha_{p+1}, c_{p+1} \rangle a_{t-1} - ... - \langle \alpha_{p+q}, c_{p+q} \rangle a_{t-q} \quad (5);$$

Step5. If the hypotheses of the model are validated, go to the next step, otherwise go to Step3 to refine the model.

Step6. The model would be ready for prediction.

The Autoregressive integrated moving average (ARIMA) has been widely and successfully applied to various systems such as social, economic, financial, scientific and technological, agricultural, industrial, transportation, mechanical, meteorological, ecological, hydrological, geological, medical, military, etc., systems.

The historical data must be less than what the ARIMA model requires which limits its application. The fuzzy regression model [16-18] is an interval forecasting model suitable for the condition of little attainable historical data. In order to make the model include all possible conditions, the spread is wide when data includes a significant difference or bias. The purpose of this paper is to combine the advantages of the fuzzy regression and ARIMA models to formulate the fuzzy ARIMA model and to full the limitations of fuzzy regression and the ARIMA model.

3.2 GRNN Model

The ANN modeling consists of two steps: the first step is to train the network; the second step is to test the network with data, which were not used for training. The processing of adaptation of the weights is called learning. During the training stage the network uses the inductive-learning principle to learn from a set of examples called the training set [19]. Learning methods can be classified as supervised and unsupervised learning. In supervised learning, for each input neuron there is always an output neuron.

As the request of intelligence traffic system is on the way, and as the training speed and predict precision is being higher and higher, the application of Neural Network in traffic field is an unshakable wise choice. Next the algorithm based on the ANN modeling [20] will be given.

In the previous traffic volume forecast by neural network, the sigmoid function usually acts as the approximate base function of forward propagation neural network. However for the lack of suitable theory frame, the topological definition of forward propagation neural network already revealed the flaw in many places. From research works and applications of Artificial Neural Network, it is easy to find that Neural Network has special advantage in traffic field, in which the most important is that it can deal with the non-linear problem perfectly in this field. The Generalized Regression Neural Network is special good at solving the traffic flow forecasting. The architecture for the Generalized Regression Neural Network (GRNN) is shown in Fig 3.

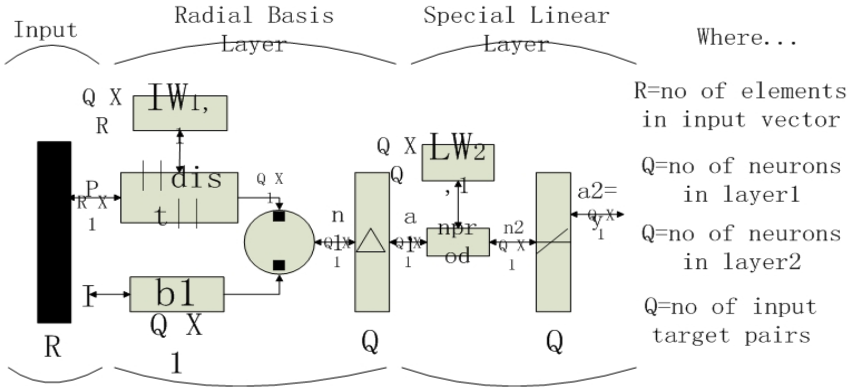

Fig. 3. The architecture of Generalized Regression Neural Network (GRNN)

The first layer is just like that for input of the neural networks. It has as many neurons as there are input/ target vectors. Again, the first layer operates just like the new be radial basis layer described previously. Each neuron's weighted input is the distance between the input vector and its weight vector, calculated with dist. Each neuron's net input is the product of its weighted input with its bias, calculated with net prod. Each neuron's output is its net input passed through radial bas. If a neuron's weight vector is equal to the input vector (transposed), its weighted input will be 0, its net input will be 0, and its output will be 1. If a neuron's weight vector is a distance of spread from the input vector, its weighted input will be spread. The second layer also has as many neurons as input/target vectors, but here LW{2,1} is set to T.

A larger spread leads to a large area around the input vector where layer 1 neurons will respond with significant outputs. As spread becomes larger the radial basis function's slope becomes smoother and several neurons can respond to an input vector. The network then acts as if it is taking a weighted average between target vectors whose design input vectors are closest to the new input vector. As spread becomes larger more and more neurons contribute to the average, with the result that the network function becomes smoother.

```
Generalized Regression Neural Network Prediction
Input: O;//Original Data Set $\left\{Data_{Traffic\quad Flow}\right\}$
Output: PVF;//Prediction Vehicle Flow $\{y_{Vel}\}$
Begin
Train = O.getData(0.7, randomSeed);
Vali = O.getData(0.15, randomSeed);
Test = O.getData(0.15, randomSeed);
GRNN = createModel();
GRNN1 = GRNN.train( Train );
$b_{\rho 0}$ = getWeightVector( $\rho_0$ );
```

```
Layer0(x) = generateFunction( $b_{\rho 0}$ );
$b_{\rho 1}$ = getWeightVector( $\rho_1$ );
Layer1(x) = generateFunction( $b_{\rho 1}$ );
i = 1;
tempValue = 0;
while(MSE $_{Layer(i-1)(x)}$ < MSE $_{Layer(i)(x)}$)
i++;
$b_{\rho i}$ = getWeightVector( $\rho_i$ );
Layeri(x) = generateFunction( $b_{\rho i}$ , tempValue);
$b_{\rho i0}$ = getWeightVector( $\rho_{i0}$ );
Neural00(x) = generateFunction( $b_{\rho i0}$ );
$b_{\rho i1}$ = getWeightVector( $\rho_{i1}$ );
Neural01(x) = generateFunction( $b_{\rho i1}$ );
j = 1;
while ( MSE$_{Neural(i)(j-1)(x)}$ < MSE $_{Neural(i)(j)(x)}$ )
j++;
$b_{\rho ij}$ = getWeightVector( $\rho_{ij}$ );
Neuralij(x) = generateFunction( $b_{\rho ij}$ , tempValue);
tempValue = $[y_{Vel}]_{\rho i-1}$;
GRNN2 = GRNN1.validate( Vali );
End
```

4 Conclusion

Here we have taken the real-time time series data which belong to the urban traffic tunnel in Wuhan. The training sample is the sample for these two model and next 288 data points as test samples to show the efficiency of the prediction method. The test-data results of our prediction would be compared with the actual data which are shown in the Fig. 4.

In this paper we reviewed several measures of Intelligent Transportation System models which are in order to solve the complicated data aggregation and its prediction work. Our main goal was to show that the better time series prediction models for the Intelligent Transportation System which can perform efficiently in this task. At end, from the test sample we could find that the Autoregressive integrated moving average (ARIMA) model works better than Generalized Regression Neural Network (GRNN) at the prediction model for short time periods.

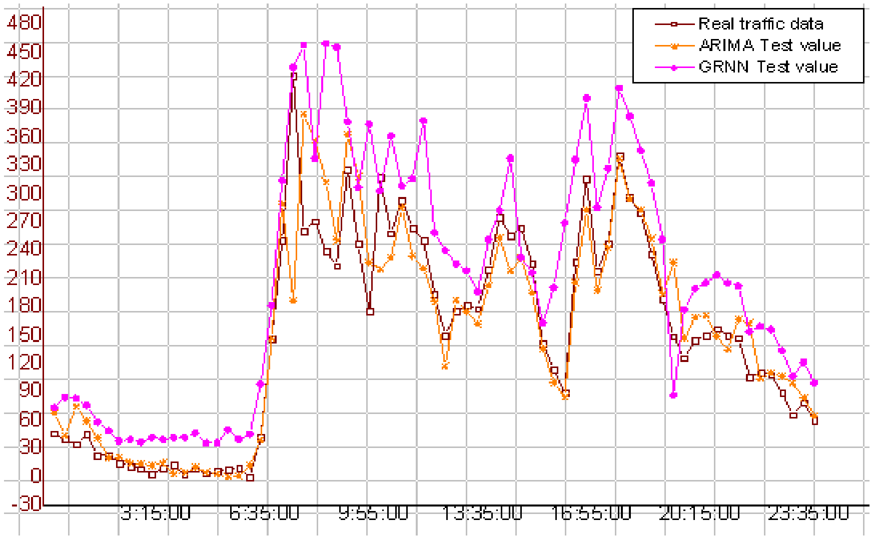

Fig. 4. Result

References

1. Beresford, A.R., Bacon, J.: Intelligent Transportation System. IEEE Pervasive Computing 5(4), 63–67 (2006)
2. Bunn, D.W., Karakatsani, N.: Forecasting electricity prices, London Business School Working Paper (2003)
3. Chen, Y., Dong, G., Han, J., Wah, B.W., Wang, J.: Multi-dimensional regression analysis of time-series data streams. In: VLDB 2002 Proceedings of the 28th International Conference on Very Large Data Bases, pp. 323–334 (2002)
4. Zadeh, L.A.: The roles of soft computing and fuzzy logic in the conception, design and deployment of intelligent systems. In: Proceedings of the 6th IEEE International Conference on Fuzzy Systems, Barcelona, Spain (1997)
5. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining Data Streams. A Review.ACM SIGMOD Record Homepage Archive 34(2), 18–26 (2005)
6. Nicolaisen, J.D., Richter Jr., C.W., Sheblé, G.B.: Price signal analysis for competitive electric generation companies. In: Proceedings of the International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, London, UK, pp. 66–71 (2000)
7. Kohzadi, N., Boyd, M.S., Kermanshahi, B.: A comparison of Artificial neural Network and Time Series Models for forecasting commodity prices. Neural Computing 10(2), 169–181
8. Zhang, G.P.: Time series forecasting using a hybrid ARIMA and NN model. Neural Computing, 159–175 (2003)
9. Saini, L.M., Soni, M.K.: Artificial neuralnetwork based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods. IEEE Proc. -Gener. Transm. 149(5), 578–584 (2002)

10. Hagan, M.T., Menhaj, M.B.: Training feedforward networks with the Marquardt algorithm. IEEE Trans. Neural Network. 5(6), 989–993 (1994)
11. Fahrmair, M., Spanfelner, B.: Security and privacy rights management for mobile and ubiquitous computing. In: Workshop on UbiComp Privacy, Tokyo, Japan, September 11 (2005)
12. Huang, E., Antoniou, C., Wen, Y., Ben-Akiva, M.: Real-Time Multi-Sensor Multi-Source Network Data Fusion Using Dynamic Traffic Assignment Models. In: Intelligent Transportation Systems, pp. 1–6. ITSC (2009)
13. Kang, Y., Lee, H., Chun, K., Song, J.: Classification of Privacy Enhancing Technologies on Life-cycle of Information. In: Proceeding of The International Conference on Emerging Security Information, Systems, and Technologies, pp. 66–70 (2007)
14. Breeden, J.L.: Modeling data with multiple time dimensions. Computational Statistics and Data Analysis 51(9), 4761–4785 (2007)
15. Tseng, F.-M.: Fuzzy seasonal ARIMA model for forecasting. Fuzzy Sets Systems 126, 367–376 (2002)
16. Jang, J.S.: Predicting chaotic time series with fuzzy if-then rules. In: IEEE International Conference on Fuzzy Systems, San Francisco, USA, pp. 1079–1084 (1993)
17. Liao, Kao, W.-H., Fan, Y., Ming, C.: Data aggregation in wireless sensor networks using ant colony algorithm. Journal of Network and Computer Applications 31(4), 387–401 (2008)
18. Cooper, J., James, A.: Challenges for database management in the internet of things. Source. IETE Technical Review 26(5), 320–329 (2009)
19. Ritchie, K.M.S.: Comparison of Traditional and Neural Classifiers for Pavement Crack Detection. ASCE 120(4), 552–569 (1994)
20. Breeden, J.L.: Modeling data with multiple time dimensions. Computational Statistics and Data Analysis 51(9), 4761–4785