Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model

Se-do Oh, Young-jin Kim, and Ji-sun Hong

***Abstract*—Current urban traffic congestion costs are increasing on account of the population growth of cities and increasing numbers of vehicles. Many cities are adopting intelligent transportation systems (ITSs) to improve traffic efficiency. ITSs can be used for monitoring traffic congestion using detectors, such as calculating an estimated time of arrival or suggesting a detour route. In this paper, we propose an urban traffic flow prediction system using a multifactor pattern recognition model, which combines Gaussian mixture model clustering with an artificial neural network. This system forecasts traffic flow by combining road geographical factors and environmental factors with traffic flow properties from ITS detectors. Experimental results demonstrate that the proposed model produces more reliable predictions compared with existing methods.**

***Index Terms*—Intelligent transportation system (ITS), traffic flow prediction, pattern recognition, artificial neural network (ANN), Gaussian mixture model (GMM) clustering.**

I. Introduction

SEVERAL countries around the world implement intelligent transportation systems (ITSs) to reduce expenses from traffic congestion. In Korea, for example, Seoul and other major cities that experience heavy traffic are actively adopting ITSs, such as those for traffic information notifications, variable signal controls, and public transportation notifications. By providing traffic information, ITSs reduce the costs associated with traffic congestion. However, current services only consider data gathered from vehicle detectors. To improve the efficiency of ITSs, the precise prediction of future traffic conditions is required; moreover, this prediction must be provided to customers.

Traffic congestion is caused by complex interactions of several factors. These factors include temporal changes in traffic volume, road architecture, weather conditions, accidents, repair work, and so on. Unfortunately, information on each of these factors is gathered and managed by separate entities. Hence, researching and predicting future road conditions is difficult because disparate traffic congestion factors must be gathered into a single model. This challenge has prompted the primary use of past traffic time-series analyses in previous research; however, this information is unreliable in cases of accidents or road condition changes. In addition, the road shape of certain areas must be limited to make predictions. To overcome these limitations, we propose a system that predicts traffic conditions by using more specific cause factors (variables) that influence traffic congestion. In this paper, we present a traffic flow prediction system that accounts for both ITS measuring variables and other environmental variables. Current systems primarily estimate a single or a few road conditions based on limited road status information. Our system, on the other hand, separates road status into environmental variables of several roads (e.g., average straight line, number of crosswalks in the area, etc.); accordingly, it can combine other roads into a single prediction. Moreover, unlike existing systems, the proposed system can respond to unexpected situations, such as weather changes or lane reductions, and it thereby provides more effective forecasting. Based on experimental results, our system obtains better prediction results compared to a system that employs an existing methodology.

Manuscript received June 10, 2014; revised October 21, 2014 and February 13, 2015; accepted March 26, 2015. Date of publication May 4, 2015; date of current version September 25, 2015. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2014R1A1A2058368. The Associate Editor for this paper was F. Chu. (*Corresponding author: Young-jin Kim*).

The authors are with the Department of Industrial and Management Systems Engineering, College of Engineering, Kyung Hee University, Yongin 446-701, Korea (e-mail: sdoh@khu.ac.kr; yjkim@khu.ac.kr; jshong@khu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2015.2419614

II. Literature Review

To date, many models have been developed for traffic prediction. Existing traffic prediction studies can be classified into use of parametric methods (e.g., the time-series analysis (TSA) and support vector regression (SVR) models), non-parametric methods (e.g., the k-nearest neighbors (KNN) model), and artificial intelligence methods (e.g., the artificial neural network (ANN) model).

The most commonly applied parametric method is the TSA model. In particular, auto-regressive integrated moving average (ARIMA) TSA models [1]–[4] have been widely used for traffic prediction using observed traffic variables. Furthermore, in many existing studies, changes of the ARIMA model have been conducted. Kamarianakis and Prastacos [5], and Min and Wynter [6], for example, respectively proposed the space-time autoregressive integrated moving-average model to satisfy interrelations between links. Moreover, Stathopoulos and Karlaftis [7] developed a traffic congestion estimation model using a multivariate time-series state space model. This model employs as input variables the amount of traffic measured by loop detectors in five different points on a straight road. They proved that this method provides better results than the ARIMA method, which uses only one variable.

In addition, some researchers have compared the seasonal ARIMA (SARIMA) model, which is a TSA model, with other methods and showed its excellent prediction performance.

Smith *et al.* [8] compared SARIMA and nonparametric (data-driven regression) models. The authors in this research concluded that the SARIMA model has better performance than the nonparametric regression models. Lippi *et al.* [9] additionally compared time-series analysis methods and SVR models and concluded that SARIMA showed the best performance.

The TSA method is known to show excellent performance for predicting the traffic of fixed links. However, in cases in which varying road links are to be predicted, the parameters for each road link must be estimated. Therefore, an apparent limitation exists when it is applied to regular road networks with many missing points.

SVR is another parametric method that is popular in many prediction applications. References [10]–[12] SVR is an extension of support vector machine (SVM) for generalizing the regression problem. Unlike the TSA model, this method predicts the linear and nonlinear relationship of data through machine learning. For most SVR models, several studies have been conducted to replace the TSA method in traffic prediction. Compared to TSA, a less user-defined parameter setup is required by SVR to reflect the tendency of data through machine learning; it is the recommended model for the practical traffic prediction method. However, the results of SVR greatly depend on kernel function and parameters. Therefore, selecting the appropriate kernel function and parameters for SVR is a trial-and-error process.

Parametric methods, such as TSA models and SVR, are advantageous in that the model with the best performance can be selected by understanding the model and adjusting the parameters. However, these methods can be time-consuming and costly for selecting and setting up appropriate parameters. Therefore, these methods make it difficult to quickly manage new roads or road structure changes.

The representative non-parametric method is the KNN algorithm. Clark [13], for example, introduced the multivariate nonparametric regression traffic congestion estimation model based on KNN. This model makes predictions based on speed, occupancy, and traffic flow measurements taken by a loop detector on a per-minute basis. In addition, Xiaoyu *et al.* [14] introduced the short-term traffic flow forecasting method using two-tier KNN. The two-tier KNN algorithm improves the pattern matching section by adding the state pattern matching step. The authors showed the results of applying short-term traffic flow forecasting on a particular section of road in Beijing.

The KNN algorithm performs prediction depending on k numbers of sample data without a model. It does not require a priori knowledge. Moreover, the algorithm is simple and shows good performance; consequently, it is becoming increasingly popular. However, KNN requires long computation time if the historical data increases. Furthermore, it is sensitive to the outliers of historical data; therefore, data filtering is essential.

The most commonly used model in artificial intelligence (AI) methods is the ANN model. For ANN, nonlinear regression can be executed through machine learning; it has become popular as a traffic prediction model with its excellent prediction performance. References [15]–[19] recently, the combined models of ANN with particular preprocessing methods (fuzzy and others) have improved performances. Quek *et al.* [20], for example, used a fuzzy neural network to estimate short-term traffic flow. They developed a prediction model based on a pseudo outer-product fuzzy neural network using the truth-value-restriction method (POPFNN-TVR). This system was shown to have better functionality than traditional feed forward neural networks using back propagation learning. Further, Chan *et al.* [21] combined a fuzzy neural network and the Taguchi method to create a traffic flow prediction model. They used the Taguchi method to set a reasonable number of on-road sensors and demonstrated that the collected information was useful.

A type of black box model, ANN is limited in that it cannot be analyzed after executing learning. However, it well reflects the collected data features. Additionally, it is flexible in the application of limited information (prediction of various non-linked multiple-road information) and is popular as a good prediction method.

Most existing research applies ITS data from the past or from a nearby area; only existing ITS measurement variables (e.g., traffic volume, travel time, speed, etc.) are used as inputs to predict future road conditions. These methodologies cannot provide precise predictions if there is a change in environmental variables (e.g., if there is construction or repair work, or changes in road structure or weather conditions). Therefore, it is important to develop a prediction system that utilizes more of the variables that cause traffic congestion.

However, challenges exist in developing this more comprehensive prediction system. First, too many variables influence traffic congestion. Traffic congestion is the combined result of traffic volume, time, date, weather, the number of lanes, the structure of the road, buildings that cause traffic congestion (e.g., large department stores, markets, hospitals, etc.), and so on. Many of these variables are beyond the detection range of ITSs and belong to different agencies; thus, they are difficult to collect and merge. Second, preprocessing these variables is difficult. The ranges of each variable can be very different. For example, traffic volumes have very large values, whereas the numbers of lanes in the roads are small single-digit values. Furthermore, each variable has a different impact on traffic congestion; it is therefore difficult to determine how much weight to assign to each variable.

To overcome these problems, we propose an urban traffic flow prediction system to predict short-term traffic speeds based on a multifactor pattern recognition model (MPRM). This model normalizes variables that have different ranges and automatically calculates weight values based on learning. Accordingly, this model predicts short-term traffic speeds.

III. Theoretical Background

A. Short-Term Traffic Flow Prediction

Traffic flow prediction, which is designed to predict future traffic flow of a certain road segment, is an important subject in ITS research. Short-term traffic flow prediction predicts the traffic volume, speed, and density of the next time interval and is an important task among traffic prediction processes. Short-term traffic flow prediction typically operates for the 5 to

30 minutes of the next time interval. In this study, we used 15-minute interval ITS data. Our system predicts the passing speed of the next time interval of 15 or 30 minutes.

B. Traffic Flow Properties

For most ITS traffic flow monitoring systems, the most important properties are speed (S), traffic volume (V), and density (D). Suwon's ITS authority (http://its.suwon.go.kr) manipulates data collected from VDS on the road to produce these three properties and store them in a database for ITS services.

The speed at a given time $S(t_i)$, is defined as the mean speed of vehicles at t_i. This is measured by taking a reference area on the roadway over a fixed period of time. In practice, it is measured by the use of loop detectors:

$$S(t_i) = \left(\frac{1}{m}\right) \sum_{j=1}^{m} s_j \tag{1}$$

where m is the number of vehicles passing the fixed point at a given time t_i and s_j is the speed of each vehicle. S is the variable that shows the state of traffic congestion in a very straightforward way. Therefore, our prediction system aims to calculate S in the near-future time interval. Consequently, S in the recent past is an important input variable to predict S in the near future.

Traffic volume, V, is the number of vehicles passing a reference point per unit of time. V is the variable strongly related to traffic congestion. If traffic volume increases, traffic congestion of the same road under the same condition will typically also increase. Hence, as input variables, we use V of roads that are linked to the road whose traffic we intend to predict. V is the variable that shows how much pressure will be exerted on the target road; it is an important input variable in our prediction system.

We use the additional property of density (D) in our prediction process. V has differing weight values depending on the structure of the road (number of lanes and crossings, etc.) to be predicted. Thus, it is difficult to predict if we will only use our target road's V. Therefore, we import D to solve above problem. D is defined as the number of vehicles per unit area of the roadway. D at a given time t_i is equal to the inverse of the average spacing at a given time t_i

$$D(t_i) = \frac{V(t_i)}{nL} \tag{2}$$

where $V(t_i)$ is the value of V measured on t_i time interval for the road that has n number of lanes, and L is the length of the road. D is the variable that can identify the number of cars on the road based on the length of the road and number of lanes. This information will revise the prediction that is based on only V.

In the present research, we used the three measuring properties (S, V, D) as important input variables. Moreover, we selected environmental variables as input variables to predict speeds in the future.

C. Artificial Neural Network Fitting Using Back-Propagation Algorithm

ANN is a mathematical simulation model that simulates the structure and learning principles of the human brain. ANN has a shape of layers—at least more than two—composed of multiple nodes. In these nodes, patterned knowledge is saved as the weighted connection strength of the synapse. We can apply a multilayer neural network (NN) that uses the function of the back-propagation learning algorithm to fit an unknown non-linear function from the data. In this method, when an input pattern is inserted into the nodes of the input level, this signal is transformed on the weight of each unit and sent to a hidden layer. It will then emit a signal in the output level at the end. From that point, the system adjusts the connection strength by back-propagation from the previous layer to reduce the difference between the output value and target value. Finally, based on the given data pattern, a non-linear function that minimizes the total error is created [26], [27].

We used ANN to discover functional relations and patterns between factors that influence traffic congestion and average traffic speed in the next time interval. In practice, it is almost impossible to guess the exact mathematical function between these causal factors and the average traffic speeds in the next time interval. We hypothesized that it is a complex non-linear function. Thus, we decided that the ANN nonlinear regression was the most suitable method to identify a non-linear function based on the data. In this study, we used the Levenberg-Marquardt (LM) algorithm applied to the back-propagation NN to fit the function. The LM algorithm we used provides generally faster convergence and better estimation results than the other training algorithms [23].

D. Gaussian Mixture Model Clustering

To classify a pattern, analyzing the distribution of data is very important. In a database in which many different types of data exist, most data is multimodally distributed. The method that assumes this multimodal distribution as a mixture of several Gaussian distributions and that estimates the distribution is called Gaussian mixture model clustering (GMM). The total probability density function $(p(x|\theta))$ is defined as a linear combination of N simple probability density functions:

$$p(O|\theta) = \sum_{j=1}^{N} p(O|w_j, \theta_j)\alpha_j \tag{3}$$

where θ is a parameter of the total probability density function, and O is the observed value. θ_j is the parameter of the jth component of the Gaussian probability density function, which includes the mean (μ_j) and covariance matrix (Σ_j).w_j is the probability variable that is the component of the jth distribution, and a_j is the weight value of the jth component's relative importance in the total mixed probability density function. To estimate all the parameters showing the distribution of data as θ by using this mixed probability model, the parameters of GMM that have N components can be expressed as follows:

$$\theta = (\mu_1, \mu_2, \ldots, \mu_N, \Sigma_1, \Sigma_2, \ldots, \Sigma_N, \alpha_1, \alpha_2, \ldots \alpha_N). \tag{4}$$

If we know N distributions, we can apply a clustering method by using each distribution's Gaussian probability density function. The method calculates each Gaussian distribution probability for a specific observed value and selects a distribution cluster with the highest probability value. This method divides data into N clusters based on data density. This method is called the GMM clustering method.

We can typically estimate that this data is produced by N Gaussian distributions by using given information or graphical analysis. However, we usually do not know from which distribution this observed value is produced. In this case, we can solve the problem with the GMM estimation method with the EM algorithm. The objective of the EM algorithm is to estimate an optimum parameter for the observed value that does not know which Gaussian distribution is used. The EM algorithm is a repetitive optimizing algorithm consisting of two steps: expectation (E step) and maximization (M step).

1. E step: Used to calculate an expectation of observed data and the hidden probability variable.
2. M step: Used to find the optimized parameter value from the expectation.

By repeating the E and M steps, we can obtain a more accurate parameter. If this repetition reaches a terminating condition, it will deduct the estimated Gaussian parameters [24], [25].

In this study, we applied GMM clustering with the EM algorithm to reasonably divide data from the reference database. We then applied ANN nonlinear regression for each cluster and used the local ANN for prediction.

IV. Outline of Developed System

The proposed system employs both observed variables from ITSs and other environmental variables for prediction. In the case of public information, such as weather conditions, we collected data from online information systems; in the case of restricted information, we obtained data from ITSs and a geographic information system (GIS) authority (the Suwon Road Traffic Information Center). We collected as many variables as possible relating to traffic congestion and constructed a reference database for the research. We extracted road structure variables from the GIS and weather information of the prediction time interval from a meteorological information system (MIS). All extracted data was preprocessed and combined with ITS measuring data of relevant roads to produce input pattern arrays. Each input pattern array was a record comprising the reference database.

Using the reference database, we performed ANN learning to identify functional relations between $X(t_i)$ and $S(t_{i+1})$of the roads.($X(t_i)$ is a input pattern array of t_i time interval; see Table I) To improve the prediction accuracy, we divided data clusters that have similar patterns to produce a lattice structure cluster. For each divided cluster, ANN nonlinear regression was applied. Then, in the prediction phase, the system performed the same preprocessing for the new data and predicted $S(t_{i+1})$ on the road with the ANN of the cluster.

TABLE I
DESCRIPTION OF INPUT PATTERN ARRAY OF t_i; $X(t_i) = [x_i, x_2, \ldots, x_{15}]$

	Elements of $X(t_i)$
Road for prediction	Measured S from previous time interval t_{i-1} (x_1)
	D from previous time interval (x_2)
	Number of small intersections (x_3)
	Number of bus stops (x_4)
	Number of crosswalks(x_5)
	Average number of straight lanes (x_6)
	Current time interval number t_i (x_7)
Left linked road	V to the predicted road's direction from previous time interval (x_8)
	Number of left turn lanes (x_9)
Right linked road	V to the predicted road's direction from previous time interval (x_{10})
	Number of right turn lanes (x_{11})
Previously linked road	V to the predicted road's direction from previous time interval (x_{12})
	Number of straight lanes (x_{13})
Weather information (unit: mm/15min)	Total amount of rainfall in a previous time interval t_{i-1} (x_{14})
	Total amount of snowfall in a previous time interval t_{i-1} (x_{15})

A flowchart of this system is shown in Fig. 1 the MPRM procedures are outlined below.

<Information extracting>

1. From ITS, GIS, and MIS databases, extract historical record information that can comprise the input pattern array.

<Initial setting>

1. Compose the input pattern array by standardizing historical data values.
2. Save the input pattern array in the reference database.
3. For a reference database dataset, GMM clustering is applied and the cluster of the lattice structure is estimated. Save the estimated scope of the cluster information in the model structure database.
4. Input pattern arrays in each cluster individually perform ANN learning (input value: input pattern array, target value: average speed of the next time interval).
5. Save each ANN learning in the model structure database.

<Prediction>

1. Apply the same method as the initial setting; compose the input pattern array for prediction.
2. Discriminate clusters of input pattern arrays for prediction.
3. Predict speed using an ANN that belongs to the clusters.

V. Research

A. Explanation of Area and Sample ITS Data

In this study, we used road information and observed variables from Suwon city in South Korea. Suwon is the seventh

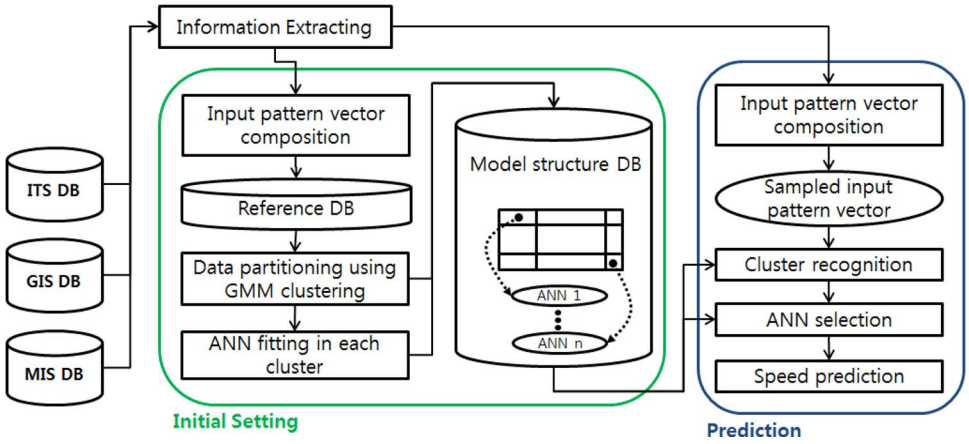

Fig. 1. Flowchart of the prediction system of this research.

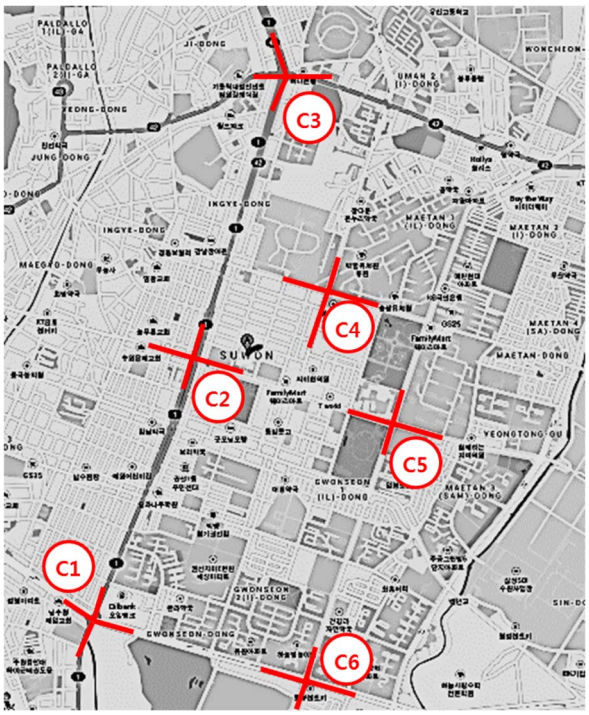

Fig. 2. Map of area used for this study (Suwon city).

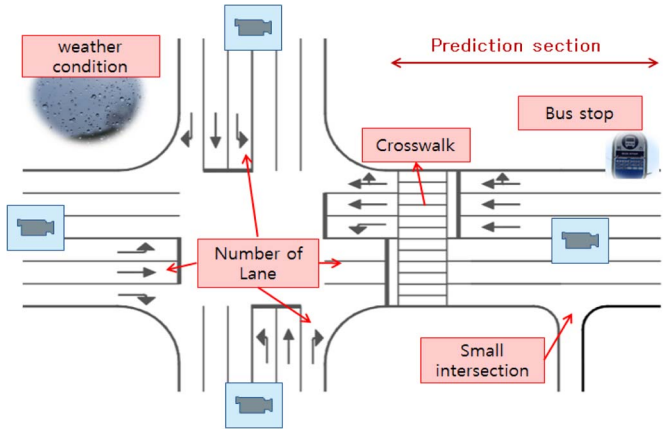

Fig. 3 Environmental variables that influence traffic congestion.

most populous city in Korea, and the fourth in population density. Therefore, it has many areas that experience traffic congestion. In this research, we used monitoring information from Dongseo-ro and Ingye-ro, which are located in the center of Suwon. Fig. 2 shows a map of the testing area of this research. We marked on the map the crossroads we tested for this study.

As shown in Fig. 2, the major roads of Suwon usually intersect perpendicularly and thus create a crossroad shape. Suwon's road management department installed VDS on each block of the crossroads. These monitor S, V and D via CCTV, laser detectors, or loop detectors. For monitoring, statistics on a 15-minute basis are used. In this research, we used ITS data from August 2012 through August 2013. This data was separated by 7:3 ratio random sampling into 82,398 samples for learning and 35,172 samples for testing. Data cleaning involved the removal of data with any missing elements. Korea has invested a great deal in the ITS system and supplies a live reporting service with the acquired measured data. To offer this live reporting service, institutions of Korea keep concerning about precise data monitoring. Therefore, there are many filtered and accumulated data available at most institutions of Korea. Of course, there were periods when the VDS was not operating due to major construction or sensor failure, leading to missing data, but this was minimal.

B. Environmental Properties of the Road

Among several types of road, we normalized the crossroads and selected the environmental elements involved in traffic congestion. ITS measuring variables were used as the important factors to predict congestion. In addition, the prediction model included environmental variables that influence the degree of congestion, including the number of lanes that connect the target road, bus stops, crosswalks, small intersections, and weather conditions on the appropriate time interval.

The general shape of the road and the environmental variables that create traffic congestion are indicated in Fig. 3.

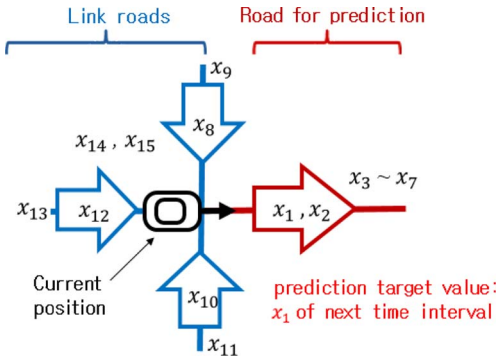

Fig. 4. Schematic diagram of input pattern arrays for the target road.

The environmental data was used for some input values of the MPRM. There are several advantages to using this data. First, MPRM enables simultaneous traffic prediction of multiple roads (road networks). This is because crossroad structural feature data (lanes of roads, lanes of linked roads, the number of bus stops, crosswalks, and small intersections) that impacts congestion was used as the input. In the MPRM model, this input data was variable when calculating the congestion level caused by the crossroad structure. Moreover, it was the variable that distinguished roads with other structures.

Second, prediction could be executed while responding to environmental changes of roads. Rainfall and snowfall were the main weather phenomena impacting congestion. Especially in Korea, summer is a rainy season and often experiences heavy rain; in winter, it often snows. During these times, vehicles tend to be driven slowly so that the traffic pattern is quite different from usual case. Therefore, if the amount of rainfall and snowfall are used as input data, better prediction could be executed in these situations. Moreover, if the roads are closed because of short-term construction, information on road lanes and lanes of linked roads could be adjusted for executing predictions. Furthermore, it could be used as a simulator predicting traffic level before construction.

Therefore, to develop a prediction system with the above advantages, we used the road's environmental variables in the prediction. The impact on congestion of these environmental variables was obviously in a nonlinear relationship. In addition, it was difficult to guess the impact of each environmental variable. Therefore, we applied an ANN nonlinear regression model that could execute nonlinear regression onto MPRM.

C. Input Pattern Array Creation

Our system predicts the degree of traffic congestion using S. Therefore, we only selected the congestion variables that influence the speed of the predicted target road and created input vectors for our prediction model.

The description of defined pattern array of t_i time interval is shown in Table I; the schematic diagram of input pattern arrays for the target road is displayed in Fig. 4

We created input pattern arrays from 15 properties that were extracted from ITS, GIS and MIS databases. The input pattern array additionally included the data of roads linked to the target road. Volumes (Vs) of the linked roads were important input values. In this study, we used the Vs of roads that were directly linked to the starting point of the target road. Realistically, the factor that influences S of the current section is the total traffic from all routes that can reach this section within 15 minutes. However, it is impossible to apply all the factors on account of missing data, the curse of dimensionality, and the problem of different weights. For practical application, we selected a reasonable ellipsis. We applied Vs from linked roads immediately adjacent to and connected with the target road. We limited the crossroads for prediction to one block units only; therefore there were no other bypasses at the starting point. Accordingly, all Vs were directly transferred to the target road via the linked road. Finally, the Vs of linked roads could be found from the total sum of the Vs that could be calculated by the traffic volume from all the routes arriving to the connection area within the time interval. Hence, we composed a prediction system that assumes that the Vs transfer from the nearby area to the target road was measured by the Vs from three linked roads. Additionally, because the influence of V depended on the number of relevant lanes, the prediction was revised by the number of lanes of the linked roads related to the given road (x_9, x_{11}, x_{13}).

Each property of the input pattern array had a different range; we therefore applied min-max normalizing to those values (0 to 1). The ANN learned the influence of each property and automatically calculated the weights. Therefore, we proceeded to the individual normalization for each single property.

D. Data Partitioning Based on GMM clustering

The input pattern arrays in the reference database included information on several crossroads. Because the reference database had many records, it was to be properly divided. The reference database had several different similar groups based on time zone and similar road properties. In this case, prediction results were not sufficient by training a global ANN function to all data in the reference database. This was because each cluster's different properties were regarded as minor errors and were normalized. Of course, we could have precisely adjusted the parameters to reflect the distribution of nonlinear data patterns, but this may have caused overfitting.

To solve this problem, we divided the data clusters by the EM algorithm and adopted GMM clustering. We divided each cluster into several selected continuous elements with one-dimensional GMM clustering; i.e., a data partition method using a hyper-cuboid lattice structure on the pattern array vector space. The input pattern array for this research had more than three dimensions; therefore, it was impossible to exactly visualize all clusters. Consequently, we provided an example that was limited to two-dimensional data.

Fig. 5 is a histogram of x_2 and x_{12}; Fig. 6 shows the estimated probability density function of these variables from applying the one-dimensional GMM model with the EM algorithm. We could estimate that x_2 was created from three Gaussian probability distributions, and x_{12} was created from four Gaussian distributions.

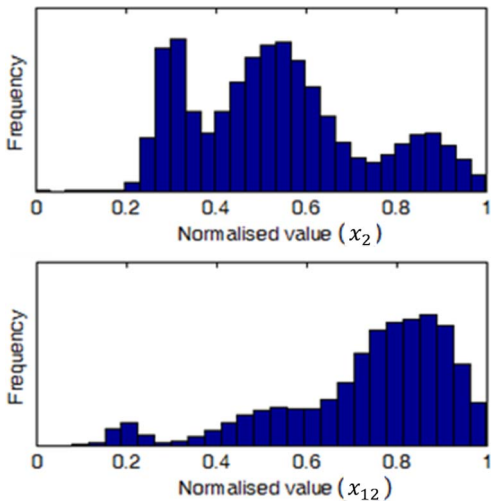

Fig. 5. Histogram of x_2, x_{12} from the reference database.

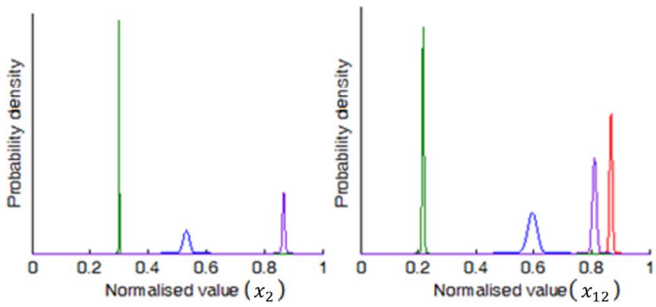

Fig. 6. Result of GMM model application on x_2, x_{12}.

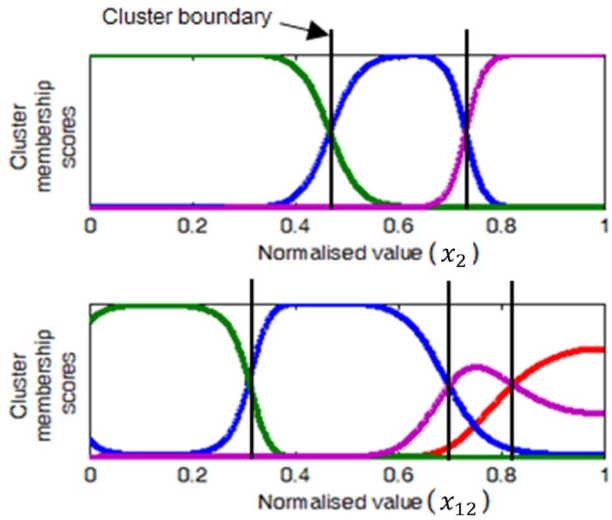

Fig. 7. Cluster membership scores and cluster boundary for the range of x_2, x_{12}.

Gaussian distributions estimated by the GMM model were used to represent each cluster. We calculated the posterior probability on each distribution for the sampled data and assigned it to the cluster with the highest posterior probability. Fig. 7 shows calculated cluster membership scores and cluster boundaries from the related variables.

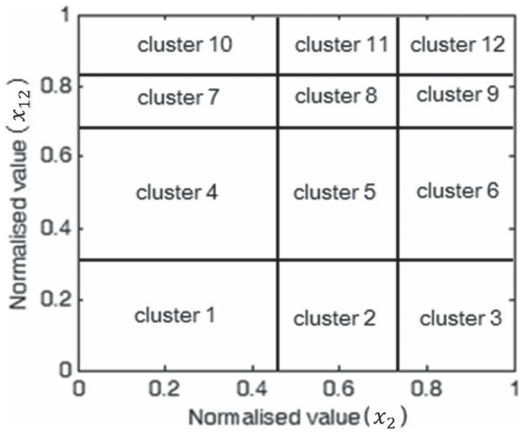

Fig. 8. Estimation of each cluster.

TABLE II
RESULT OF SAMPLE DATA CLUSTERING

Measurement range	Related variables	Number of cluster range
Road for prediction	x_1	4
	x_2	3
Left linked road	x_8	6
Right linked road	x_{10}	6
Previously linked road	x_{12}	4

We could estimate the lattice structure cluster on the plane coordination that has the range of two variables by using calculated individual cluster boundaries. Fig. 8 shows an estimate for each cluster.

In this way, we applied one-dimensional GMM clustering to the continuous measuring variables, x_1, x_2, x_8, x_{10} and x_{12}, to divide the vector space into grid sections. We could have also used GMM clustering for data partitioning of multi-dimensional Gaussian distributions. However, that method causes time delays due to cluster membership score calculations. In the case of multi-dimensional GMM clustering, the boundary of each cluster was a hyper-curved surface and we could not pre-calculate the range of the data. Therefore, the system had to calculate all the probabilities that a certain cluster contained sample data, which caused time delay. Hence, we recommend using lattice structure clustering with one-dimensional GMM on several elements. This method divides clusters into hyper-cuboid lattices; it is therefore possible to divide the range of data in advance. Using this procedure, we did not need to calculate the probability of new observed values. Table II is the result of clustering the sample data, which was divided into 1,728 clusters.

E. Local ANN Nonlinear Regression

To predict future values of S, it was important to identify the functional relation between the input pattern arrays of the current time and S of the next time interval. We used ANN nonlinear regression by assigning an input pattern array as input and S of the next time interval as the target to show the relation between these two values with a nonlinear function.

In the previous stage, we divided data with GMM clustering. Therefore, as a result of ANN learning, local ANNs with a number identical to the number of clusters were created. (For fitting, we used LM back-propagation NN, which are both accurate and fast at learning.) In this research, the local ANN network was composed of 15 input nodes, 40 hidden nodes, and 1 output node. Each local ANN was identical. The number of hidden nodes has been considered one of the most important parameters. In ANN, the unnecessary hidden nodes could reduce learning convergence speed. On the other hand, ANN with enough hidden nodes can have a more accurate multidimensional nonlinear function relationship compared to ANN with fewer numbers of hidden nodes. In the past, obtaining a reasonable hidden node for a computer calculation time was a critical issue. However, for the Levenberg-Marquardt back-propagation NN case, it guaranteed a fast learning speed compared to other ANN types; moreover, remarkable development of current computer hardware performance sufficiently supports acceleration of ANN learning. Therefore, the difference of learning speeds due to differing numbers of hidden nodes is not the critical problem that was in the past. In this study, we selected 40 hidden nodes to produce sufficiently good results for the comparative method, which was NR-ANN's learning result. As a comparison under the same condition, we applied ANN with the same number of hidden nodes for MPRM. We could have strived to seek the optimal value by applying fewer hidden nodes on the clustered data. In this study, however, we aimed to focus on system development using MPRM; therefore, we applied the fixed 40 hidden nodes.

The termination condition of ANN was set as less than a very small number (0.000001) on the change of total mean squared error calculated during each epoch. Additionally, it was set to terminate when it reached 1,000 epochs. For MPRM, the learning was with 1728 local ANNs and an average terminating epoch was 181.32 epoch. ANN should thereby address the overfitting problem (i.e., a function passing through all the coordinate points of the training set if there is no stop condition). To prevent overfitting, a sampling of the validation set that was not engaged with learning was performed when ANN learning began. Once the learning of each epoch was completed, the error of validation set was calculated. This calculation was repeated for all epochs and learning was not stopped until the value did not change. As in our study, if the size of the learning set is sufficiently large, the convergence evaluation employing the validation set with no engagement of learning can be a method for preventing the overfitting problem.

In Table III, the ANN nonlinear regression model of MPRM is outlined. Fig. 9 shows the structure of ANNs used in this research.

TABLE III
ANN NONLINEAR REGRESSION MODEL OF PRESENT STUDY

Input layer	15 nodes [15 input pattern array]
Hidden layer	1 layer, 40 nodes
Output layer	1 nodes [time mean speed of next time interval]
ANN type	Levenberg-Marquardt back-propagation NN
Normalization range	[0,1]
Number of local ANNs	1,728
Learning set	82,398 samples
Test set	35,172 samples

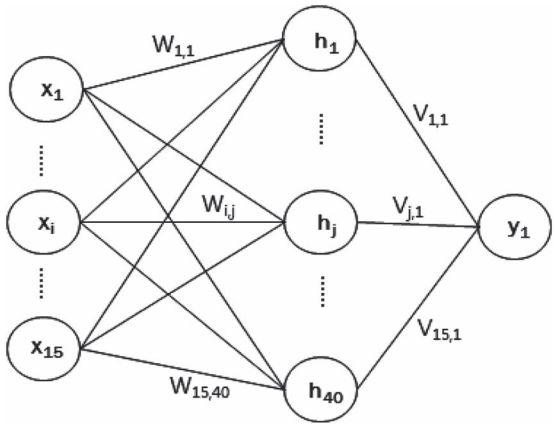

Fig. 9. Structure of local ANN in the present study.

VI. Comparison of Other Models

Next, we compare our methodology with others to show that our system enhances performance. Methodologies from previous research have their own assumptions and different types of data structures; it is therefore impossible to directly compare their results with ours. Therefore, we applied our sample data to existing methodologies and compared them.

A. Current Speed Prediction Method

The current speed prediction (CSP) method uses S of the previous time interval as the prediction value for the next time interval. This method is mainly used in current traffic congestion notification services. If road C_i predicts S as $\hat{S}(C_i, t_{i+1})$ for t_{i+1} time interval, and measured S of the road C_r is $S(C_r, t_i)$ for t_i time interval, then the CSP can be summarized by the following equation:

$$\hat{S}(C_i, t_{i+1}) = S(C_i, t_i). \tag{5}$$

B. Historical Mean Prediction Method

The historical mean prediction (HMP) method predicts the result with average S on each road for each time interval. If there are n past data points for time interval, jth historical S of the road C_i is $S_j(C_i, t_{i+1})$ for t_{i+1} time interval, then the HMP can be expressed as the following equation:

$$\hat{S}(C_i, t_{i+1}) = \frac{1}{n} \sum_{j=1}^{n} S_j(C_i, t_{i+1}). \tag{6}$$

We set the n as the total number of S of time interval t_{i+1} from the historical data.

C. Multiple Linear Regression Method

The multiple linear regression (MLR) method can be used to identify dependent variables and independent variables with linear functional relations if there are more than two independent variables. This method uses least squares point estimates to estimate a linear function that can represent the data. References [22], [28] If the selected input pattern array X's estimation speed is $\widehat{S}(X, t_{i+1})$ for time t_{i+1} interval, we can predict the value by using the linear function estimated by MLR.

$$\hat{S}(X, t_{i+1}) = \beta_0 + \sum_{j=1}^{15} \beta_j x_j \tag{7}$$

where $[x_1, x_2, \ldots, x_{15}]$ is X and $\beta_0, \beta_1, \ldots, \beta_{15}$ are regression coefficients.

D. Multivariate Nonparametric Regression Method Using k-Nearest Neighbor

The multivariate nonparametric regression method using k-nearest neighbor (MNR-KNN) extracts k neighbor (short distance) input pattern arrays for the observed value's input pattern array from historical data (of the reference database). From that point, it uses the future value of k neighbor input pattern arrays to estimate the result. Reference [13] we applied the average number of k neighbor inputs to the future value of the pattern arrays. If there are k neighbor input pattern arrays, $X_1, X_2, \ldots, X_k$, and the historical S of these arrays are $S(X_j, t_{i+1})$ for t_{i+1} time interval, it can be expressed as follows:

$$\hat{S}(X, t_{i+1}) = \frac{1}{k} \sum_{j=1}^{k} S(X_j, t_{i+1}). \tag{8}$$

In this study, we set the parameter k as 15.

E. Nonlinear Regression Method Using ANN Fitting

The nonlinear regression method using ANN fitting (NR-ANN) provides an estimate based on the ANN global function by learning all data in the reference database. This method is identical to our proposed model except that it does not have GMM clustering. The network structure of NR-ANN is composed of 15 input nodes, 40 hidden nodes, and 1 output node, which is identical to our model. The learning method is cross validation, and the learning algorithm is Levenberg-Marquardt back-propagation, which are the same as in our proposed model.

F. Nonlinear Regression Method Using Support Vector Regression

SVR is a generalized method that is adjusted for the regression problem of the SVM used for classification. Once the appropriate kernel function is applied to SVR, the nonlinear regression can be executed. Reference [11] if the selected input pattern array X's estimation speed is $\widehat{S}(X, t_{i+1})$ for the t_{i+1} time interval, the ultimate regression equation estimated based on the machine learning of SVR is expressed as:

$$\hat{S}(X, t_{i+1}) = \sum_{j=1}^{m} (\alpha_j^* - \alpha_j) \boldsymbol{K}(X, X_j^{sv}) + b. \tag{9}$$

Here, α_j^*, α_j are parameters decided during SVR learning, b is a constant term of regression equation, X_j^{sv} is a support vector of the learning set, and $\boldsymbol{K}(X, X_j^{sv})$ is a kernel function. To apply SVR, the user-defined parameters, such as error penalty parameter C, the insensitive loss function tube size ε, and kernel function $\boldsymbol{K}$, should be set. This will impact the result of the SVR machine learning.

The kernel function used in this research is the radial basis function (RBF), the most commonly used kernel function in nonlinear regression.

$$K(X, X_j^{sv}) = \exp\left(-\frac{\left\|X - X_j^{sv}\right\|^2}{2\sigma^2}\right) \tag{10}$$

σ is a parameter of RBF, and the user-defined parameters C, ε, σ significantly affect SVR model estimation. Therefore, SVR requires considerable research time at the initial setup stage for establishing appropriate parameters. To set appropriate parameters, we conducted an experiment using a model with 1,000 different parameters. The setup range of each parameter was C $= [0.1 \sim 1]$, $\varepsilon = [0.01 \sim 0.1]$, and $\sigma = [0.1 \sim 1]$. We applied changes of ten stages each with uniform differences. Among them, the result of the model with the least number of prediction errors was compared with MPRM (C $= 0.9$, $\varepsilon = 0.01$, and $\sigma = 0.2$).

VII. Test

A. Performance Test Procedure

The performance test experiment was executed based on the procedure shown in Fig. 1; it followed the procedures outlined below.

<Preprocessing>

1. Extract information from ITS, MIS, and GIS databases that compose the input pattern array.
2. Compose the input pattern array.
3. Distribute the training and test sets (7:3).

<Training>

1. Apply GMM clustering on the training set. Save the lattice structure cluster scope information.
2. Input pattern arrays in each cluster individually perform ANN learning (input value: input pattern array, target value: average speed of next time interval).
3. Save each ANN learning.
4. Calculate the training time.

<Performance test>

1. Discriminate clusters of input pattern arrays in the test set.

2. Predict speed using the local ANN that belongs to the clusters.
3. Calculate the mean absolute error(MAE) of the next time interval between the predicted and actual values.
4. Calculate the prediction time.

We applied 7:3 random sampling in this experiment because we believe it is important to reflect the impact of rainfall and snowfall in the model learning. We aimed to design a prediction model with the data considering the impact of environmental changes. Unfortunately, our data set included data of only one year. If we used the initial 70% and later 30%, only fall, winter, and spring data would have been included in the training set; only summer data would have been included in the test set. As mentioned earlier, summer in Korea is a heavy rain season. Subsequently, the difference in rainfall amounts is significant compared to other seasons; therefore, the road speed patterns are irregular. For MPRM, which we suggested, it is not required to set the input to match the time flow as with time series analysis. If it is assumed that a vehicle passes a particular road, MPRM will use for prediction using the functional relationship of the input data that was observed from the time interval one step before on the given road and the average speed of the current time interval through which the vehicle passes. This method does not require an extended linkage with data of past time intervals. Therefore, even if the performance is evaluated using random sampling, the result of the performance evaluation will not be notably different. Therefore, the 7:3 random sampling was executed.

B. Test Results

We used MAE to measure the prediction performance. MAE measures how close forecasts or predictions are to the eventual outcomes. MAE is given by

$$\text{MAE} = \frac{1}{N}\sum_{j=1}^{N}|S_j - \hat{S}_j| \tag{11}$$

where $\hat{S}_j$ is the prediction S of j sample, S_j is the actual speed of j sample, and N is the sample number. MAE was used to evaluate prediction accuracy because it is an evaluation standard with no unit change. For example, if MAE has a value of 2, the prediction result has an average ± 2 km/h error level.

In this study, we calculated the total MAE and rush hour MAE of each model. Rush hour was defined as time intervals from 7:00 to 10:00 (GMT +9:00) and 17:00 to 20:00. During this time, the variation of traffic volume expanded; the estimation error likely increased. Estimation during rush hour is more important than other times because there is more congestion. Therefore, we used the rush hour MAE as a major evaluation factor for the test. Results of the MPRM showed better estimation performance over other methodologies. These test results are outlined in Table IV; Fig. 10 displays the MAE graph of each model per interval.

We evaluated prediction performance by extracting the data set only when there was rain or snow in the test set. The performance evaluation results are shown in Table IV. As expected, the multivariable methods using environmental information showed better prediction results on changing weather situations compared to the multivariable methods, which did not use environmental information. Among these, the prediction performance of MPRM was the best.

Compared to the other methods, MPRM showed good prediction performance. It outperformed the methods, especially CSP and HMP, which did not use environmental variables. Furthermore, it showed good results when our data set was applied to the other methods. We believe it was the features of MPRM that reasonably partitioned data with similar features through GMM clustering and applied ANN on each cluster to execute regional prediction.

In addition, we compared the computation times of the proposed MPRM and the other methods; the results are shown in Table IV. The computation time of each model could be divided into initial setting time (parameter optimization time and training time) and prediction time. The parameter optimization time was the time parameter for appropriate estimation. It should be noted that it was difficult to accurately measure the parameter optimization time; therefore, it was recorded with relative state. The training time was the time required to estimate each model using the learning set. The prediction time was the time required to calculate the prediction value of all test sets. The same computer was used in the performance comparison.(CPU: Intel I7; RAM: 64 GB; programming language: MATLAB R2013a)

According to the results, MPRM showed the second longest training time. This is because it executed learning in ANN to estimate the nonlinear functional relationship of the input array and target value. However, learning was executed by using only the learning data that belonged to each cluster through GMM clustering; therefore, the time for each ANN to collect learning was reduced. Consequently, the initial setting time was reduced compared to the NR-ANN method. For MPRM, ANN learning was another technique that could be used to reduce the initial setting time using parallel computing. Unlike the other methods, MPRM enabled the partitioning of the learning sets of each cluster into different computers; accordingly, simultaneous learning using different computers was possible. In fact, for the prediction model case, the initial setting time could be accomplished during unused system time; therefore, the importance of setting time is lower compared to the prediction time.

For SVM, its prediction accuracy followed MPRM and it showed the second longest prediction time. Moreover, it had a long user-defined parameter optimization time compared to other methods. MNR-KNN, which showed a similar performance as SVM, required the finding of k numbers of neighboring points in the database at the prediction stage. This delayed prediction time. Further, if there were many requests in real-time prediction, the service time would have increased.

Although MPRM showed the third longest training time, its prediction time was 8.47 second; therefore, its performance was satisfactory. At the prediction step, if prediction was executed with an ANN saved in a cluster unit in each computer with parallel computing, the prediction time would have been reduced. If there are significant prediction needs in the future,

TABLE IV
TEST RESULTS (TOTAL MAE, MAE OF RUSH HOURS, MAE OF RAINFALL OR SNOWFALL CONDITIONS AND COMPUTATIONAL TIMES)

Prediction model	Total MAE	MAE of rush hours	MAE of rainfall or snowfall conditions	Computational time (s)		
				Initial setting time		Prediction time
				Parameter optimization time	Training time	
CSP	3.0642	3.5634	4.5727	None	-	-
HMP	3.1645	3.7970	5.2475	None	13.08	-
MLR	2.9921	3.5007	4.2648	Short	401.65	4.04
MNR-KNN	2.5852	2.9847	3.3541	Short	-	75435.33
NR-ANN	2.6050	3.1697	3.5542	Short	2457.21	7.66
NR-SVM	2.4485	2.9797	3.3827	Long	1600.78	187.45
MPRM	2.2819	2.5885	3.1881	Short	1777.01	8.47

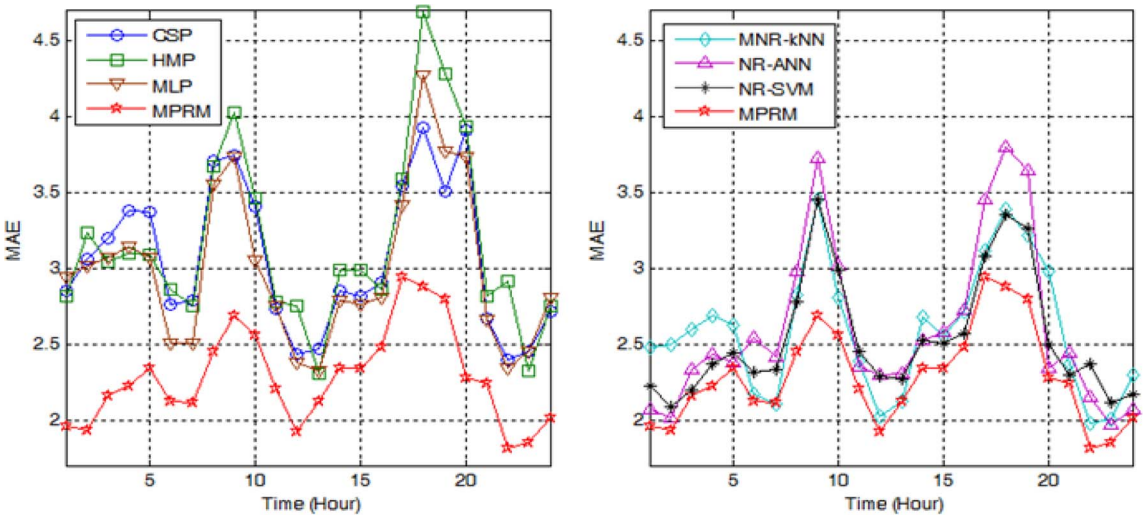

Fig. 10. MAE graph of each model per time interval.

MPRM will have the advantage of increasing service bandwidth through parallel computing.

VIII. SUMMARY

For efficient ITS service, a system that precisely predicts a city's road conditions is critical. In this paper, we proposed a new prediction system that uses more than current measurements. This new system utilizes traffic flow properties that are detected from ITS detectors, geographical information, and environmental information including weather conditions of the road. To achieve this goal, we extracted the required information from GIS and MIS databases and merged it with proper preprocessing. To perform prediction with data from combined reference databases, we proposed an MPRM that integrates a GMM clustering method and an ANN local function fitting method. Our proposed methodology shows better performance than the existing methodologies mentioned in this paper. This system can provide an accurate prediction on future road conditions. In addition, it can perform adaptive forecasting to predict traffic congestion caused by lane reductions for construction or incidents due to rain and snow. In the future, we will develop an improved prediction model that will additionally account for accidents and buildings that trigger heavy traffic congestion.

REFERENCES

[1] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.
[2] B. M. Williams, M. Asce, L. A. Hoel, and F. Asce, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.
[3] D. Billings and J. S. Yang, "Application of the ARIMA models to urban roadway travel time," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2006, pp. 2529–2534.
[4] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with ARIMA-GARCH model," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 607–612.
[5] Y. Kamarianakis and P. Prastakos, "Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches," in *Proc. 82nd Annu. Meet. TRB*, Washington, DC, USA, 2003, pp. 1– 25.
[6] W. Min and L. Wynter, "Real-time road traffic prediction with spatiotemporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, Aug. 2011.
[7] A. Stathopoulos and M. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transp. Res. C, Emerg. Technol.*, vol.11, no. 2, pp. 121–135, Apr. 2003.
[8] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.
[9] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[10] J. T. Ren, X. L. Ou, Y. Zhang, and D. C. Hu, "Research on network level traffic pattern recognition," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, 2002, pp. 500–504
[11] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.
[12] F. Wang, G. Z. Tan, C. Deng, and Z. Tian, "Real-time traffic flow forecasting model and parameter selection based on ε-SVR," in *Proc. 7th WCICA*, 2008, pp. 2870–2875.
[13] S. Clark, "Traffic prediction using multivariate nonparametric regression," *J. Transp. Eng.*, vol. 129, no. 2, pp. 161–168, Mar. 2003.
[14] H. Xiaoyu, W. Yisheng, and H. Siyu, "Short-term traffic flow forecasting based on Two-tier K-nearest neighbor algorithm," in *Proc. Intell. Integr. Sustain. Multimodal Transp. Syst.*, 2013, pp. 2529–2536.
[15] C. Ledoux, "An urban traffic flow model integrating neural network," *Transp. Res. C, Emerg. Technol.*, vol. 5, no. 5, pp. 287– 300, Oct. 1997.
[16] R. Yasdi, "Prediction of road traffic using a neural network approach," *Neur. Comput. Appl.*, vol. 8, no. 2, pp. 135–142, May 1999.
[17] S. Ishak, P. Kotha, and C. Alecsandru, "Optimization of dynamic neural network performance for short-term traffic prediction," *Initiatives Inf. Technol. Geospatial Sci. Transp.*, no. 1836, pp. 45–56, 2003.
[18] E. Vlahogianni and M. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 3, pp. 211–234, Jun. 2005.
[19] H. Yin, S. C. Wong, J. Xu, and C. K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 2, pp. 85–98, Apr. 2002
[20] C. Quek, M. Pasquier, and B. B. S. Lim, "POP-TRAFFIC: A novel fuzzy neural approach to road traffic analysis and prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 133–146, Jun. 2006.
[21] K. Y. Chan and T. S. Dillon, "On-road sensor configuration design for traffic flow prediction using fuzzy neural networks and Taguchi method," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 1, pp. 50–59, Jan. 2013.
[22] J. Rice and E. V. Zwet, "A simple and effective method for predicting travel times on freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 3, pp. 200–207, Sep. 2004.
[23] M. S. Bascil and F. Temurtas, "A study on hepatitis disease diagnosis using multilayer neural network with Levenberg Marquardt Training Algorithm," *J. Med. Syst.*, vol. 35, no. 3, pp. 433–436, Oct. 2011.
[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000, pp. 95–157.
[25] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Oxford, U.K.: Elsevier, 2009, pp.703–709.
[26] S. L. Fausett, *Fundamentals of Neural Networks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993, pp. 289–333
[27] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000, pp. 323–373.
[28] B. L. Bowerman, R. O'Connell, and A. Koehler, *Forecasting, Time Series, and Regression*. Stamford, CT, USA: Thomson Brooks/Cole, 2004, pp. 136–197.

Se-do Oh is currently working toward the Ph.D. degree with the Department of Industrial and Management Systems Engineering, College of Engineering, Kyung Hee University, Yongin, Korea. His research interests include the areas of artificial intelligence, knowledge discovery in databases, machine diagnostics, and pattern recognition.

Young-jin Kim received the Ph.D. degree from the University of California Berkeley, Berkeley, CA, USA, in 1991. Since 1994, he has been a Professor with the Department of Industrial and Management Systems Engineering, College of Engineering, Kyung Hee University, Yongin, Korea. His research interests include the area of artificial intelligence, particularly in diagnostics and design.

Ji-sun Hong received the master's degree Kyung Hee University, Yongin, Korea, in 2014. She is currently with the Department of Industrial and Management Systems Engineering, College of Engineering, Kyung Hee University. Her research interests include big-data analysis, design, and inspection systems.