Accuracy Analysis of Short-term Traffic Flow Prediction Models for Vehicular Clouds *

Fan Zhang
PARADISE Research
EECS, University of Ottawa
Ottawa, Canada
fzhan009@uottawa.ca

Robson E. De Grande
PARADISE Research
EECS, University of Ottawa
Ottawa, Canada
rdgrande@site.uottawa.ca

Azzedine Boukerche
PARADISE Research
EECS, University of Ottawa
Ottawa, Canada
boukerch@site.uottawa.ca

ABSTRACT

Vehicular Clouds introduces a new paradigm that addresses and potentially enhances underutilization of on-board computing resources through aggregation to solve several computational tasks in Intelligent Transportation System. The most challenging issue in Vehicle Cloud is the task allocation among the dynamically changing amount of available resources. For further research towards this issue, a realistic road traffic system models which could generate traffic flow with high accuracy must be designed. In this paper, we conduct a study on short-term traffic flow predictions for our envisioned road traffic prediction system. Five prediction models, including double exponential smoothing (DES), seasonal autoregressive moving average (SARIMA), K-nearest neighbor (KNN), back-propagation neural network (BP-NN) and support vector regression (SVR), are implemented. Then, three different error metrics are used to evaluate the performance of these models. Finally, the results shows that SARIMA and BP neural network are two precise and stationary prediction models and thus are the best candidates to be embedded in an road traffic load prediction system.

methodologies

CCS Concepts

•General and reference → Estimation; •Mathematics of computing → *Time series analysis;* **•Computing methodologies → Machine learning approaches;**

Keywords

Traffic Model; Vehicular Clouds; Traffic Analysis

1. INTRODUCTION

The past few decades have witnessed the great development of vehicle networks. Vehicular networks interconnect vehicles on the road by wireless communication technologies, including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Traffic safety related messages, like post crash notification, cooperative control warning, and road hazard control notification, could be delivered to drivers among vehicles [14]. By reducing traffic congestions and avoiding vehicle collisions, vehicular networks improves driving experience and safety traffic system. Thus, vehicular networks are now considered as a key component for Intelligent Transportation System (ITS). However, several services, such as the selection of alternative routes and the synchronization of traffic lights for the increasingly heavy road conditions, are considerably computationally intensive [5]. They do not require additional enabling infrastructure that expands on the current Vehicular Networks-based ITS. Hence, there is a need to expand the current ITS solutions into a more flexible paradigm, which can provide more efficient access to high computing power.

A source of such computing resources can be promptly accessible in the vehicles parked or traversing a region of the traffic network. With the advancement of automobile industry, the *smart cars* nowadays are equipped with a series of ICT devices, which includes an on-board computer, large storage devices, GPS system, and numerous sensors [9]. Therefore, smart vehicles are be referred as *computer on wheels*. A recent report shows that registered vehicles on American streets and roadways is almost 256 million [20]. However, most of these vehicles stay in parking lot, parking garage and driveway for some hours everyday. During these period, the computing and storage resources of these vehicles are idle or underutilized. As a result, if these idle resources of smart vehicles are well exploited, many services could be feasible and contribute to a more safe traffic system.

Therefore, a promising paradigm called Vehicular Cloud Computing (VCC) has been proposed by Olariu et al. [2]. VCC aims to utilize idle on board resources, like storage and computing capability to support computational services in ITS. Several typical instances of VCC could be *airport as a data center*, *parking lot data cloud*, and others, which all exploit storage resources on long-term parking vehicles [17]. In this scenario, the VCC acts like the conventional Cloud whose nodes remain stationary. This scenario is classified as the static formation of VCC. Contrarily, the dynamic formation of VCC is defined as gathering computing resources on moving vehicles on the road. The dynamic VCC still remains as a challenging task mainly because of the changing computing resources, namely the vehicles, from time to time in a given formation area [26]. Hence, allocating computational

*This work is partially supported by NSERC, the Canada Research Chair program, MRI/ORF research funds, and the Ontario Distinguished Research Award.

PE-WASUN'16, November 13-17, 2016, Malta, Malta

DOI: http://dx.doi.org/10.1145/2989293.2989297

tasks to vehicles among dynamic Vehicular Cloud is rather sophisticated. Especially in the border of formation area, the tasks on a leaving vehicle should be transferred to the inside vehicles, and similarly, the arriving vehicles should be assigned with tasks rather than keep its computing resources underutilized. Moreover, the overall goal of task allocation on dynamic VCC is to make each vehicle assigned with equal task which is similar with but more complex than load balancing in conventional cloud computing due to the highly mobility and changing amount of vehicles in dynamic VCC.

Under the context of dynamic VCC, we first expect to build a road traffic system which could be as realistic as possible. Then, task allocation strategy of dynamic VCC could be tested on this model. In the context of road traffic simulation, several urban mobility simulators as a reasonably representative testing tool, providing comprehensive vehicle mobility models and great facilities in importing Open Street Map, for instance. However, the amounts of vehicles are randomly generated by such simulators with an uniform distribution or simply through a modified distribution. It is noticeable that the current traffic simulators lack on well fitting real traffic load in our daily life. This motivated us to generate real, or realistic, traffic flows through predictions. Hence, we divide the modeling of our road traffic system into tow steps: a short-term traffic flow prediction based on historic traffic volume and a road traffic modeling by aggregating the forecasting traffic volume with mobility models.

In this paper, we focus on first providing an accurate analysis on existing available prediction tools most fit for forecasting traffic flows, namely the short-term forecasting of traffic flow based on historic data. Five most commonly used prediction models are observed, determining the model that best suits into the requirements and characteristics of our envisioned road traffic prediction system.

The remaining of this paper is organized as follows. Section 2 summarizes the common concept and recent work of vehicular cloud computing and then introduces related works for traffic flow prediction. Section 3 shows the detailed theory and key formulas of these prediction models as well as several advantages and disadvantages. Section 4 presents the prediction results and a comprehensive performance analysis through three types of error metrics. Section 5 concludes this paper and provides future work directions.

2. RELATED WORKS

Estimating the number of vehicles in a road segment enables a series the proper provisioning of services and utilization of resources. An overview on the general characteristics of the recent existing works on Vehicular Clouds contextualizes on such needs. On the other hand, several flow prediction models have already been proposed to estimate traffic load, so knowing which model suits best in context of urban scenarios empower us on to build an intricate and more realistic model that can accurately forecast traffic flow.

2.1 Vehicular Clouds

Vehicular Cloud is a newly proposed concept inspired by modern Cloud Computing technologies to exploit underutilized on board resources such storage, computing resources, network access and so on. Most recent work on vehicular cloud are on taxonomy definition level, including architecture [3] [20], service models [2] [17], privacy and security challenges [4] [22]. Mainly, the architecture and service models of vehicular cloud are summarized as follows:

- Architecture: inside vehicle layer, communication layer, and cloud layer are three main layers of Vehicular Cloud. The inside vehicle is the bottom layer and consists of various sensors like body senor, environment sensor and the vehicle's internal sensor. The main function of inside vehicle layer is to collect driver's body information like pressure and temperature as well as road condition information for applications on top layer. The middle layer is the communication layer which takes advantages of the Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication technologies in vehicular networks. Thus, this layer aids on exchanging data between vehicles through the cloud layer. The cloud layer can be classified into three sublayers: a cloud infrastructure layer containing storage unit and computation unit, a cloud platform layer and an application layer defining the real-time services and cloud primary services. This current architecture of vehicular cloud shows that the conventional cloud computing is still a key component of vehicular cloud. Since the resources we can access on vehicles are quite limited sometimes, the conventional cloud computing could provide stationary storage and computing resources to ITS.

- Service models: Four typical service models are Network as a Service (NaaS), Cooperation as a Service (COaaS), Storage as a Service (STaaS) and Computation as a Service (CaaS). The NaaS allows drivers to rent out their underutilized network access to those drivers whose cars do not present network connections. The COaaS requires all the drivers that subscribe to a particular service to publish related information to the network. For example, in a route optimization service, all the drivers need to provide their geographical positions to generate the traffic density map. Then, optimized routes are available for drivers by this service. The STaaS like *parking lot data cloud* provides idle storage capacity to users who need additional storage to run their applications. Specially, a replication-based storage should be used in order to increase the reliability and availability of stored data since some vehicles may have left when you want the access to your data. The CaaS has already been discussed above and is the main research interest in this paper.

2.2 Short-term Traffic Flow Prediction

Short-term traffic flow prediction plays a important role in Intelligent Transportation System. Several applications in ITS for example the advanced traffic management system(ATMS) and advanced traveler information systems(ATIS) requires future traffic volumes to determine control policies for future traffic management [1]. In this paper, the traffic volume is predicted specially as the input to our expected road traffic system so that the system could provide more realistic environment for the task allocation research for vehicular cloud.

Traffic flow is a type of time series with obvious seasonality pattern. Many forecasting methods for time series could be applied to traffic flow prediction. Among these methods, there are two main categories:

- Statistical Methods: these methods can be smoothing models such as Naïve [18], historical average (HA) [11], and exponential smoothing (ES) [12]. These models are simple and real-time, but their prediction errors are usually the higher than the other prediction models. Parameters in these models can be prescribed by rule of thumb. More complex statistical models are autoregressive integrated moving average (ARIMA) [6] and seasonal auotoregressive integrated moving average (SARIMA) [21]. The ARIMA and SARIMA are generalization model for most time series in real life; thus, they can produce rather precise prediction results. However, the parameter estimation for ARIMA and SARIMA would be more sophisticated and requires adequate amount of historic data.

- Machine Learning Methods: these methods are now broadly applied to forecasting because of their highly prediction accuracy and easy implement. Consequently, there is no need to estimate most parameters of these models, just let the machine learn a large number of training samples with a particular learning algorithm. As a result, the model parameters can be determined during the learning process. There are various machine learning methods like K-nearest Neighbor (KNN) [25], Artificial Neural Network (ANN) [7], Radial Basis Function network (RBF) [23], Support Vector Regression (SVR) [24], and Bayesian [16].

Among the prediction models above, two statistical models including ES and SARIMA and three machine learning methods including KNN, ANN and SVR are implemented and compared in this paper. By comprehensive performance analysis, our aim is to decide the prediction models, as well as the situations in which the models, that best fit into the conditions for our envisioned road traffic prediction system.

3. PREDICTION MODELS

Since traffic load has been a crucial aspect for the management of transportation systems, several techniques have applied to predict or estimate this type of load in order to enable or ease the management and decision-making. The models are not only restricted to regression analysis but also expanded into deep learning analysis, which greatly increases the complexity of the solution, as well as the precision of the estimate. Several methods exist and the most prominent and fit are considered in this work and described in the following subsections.

3.1 Double Exponential Smoothing

A time series usually follows certain tendency and stability, so it can be reasonably inferred from the respective previous data points. The exponential smoothing (ES) method was proposed under this assumption and is widely used in most real-time forecasting models for the convenience it brings in providing relatively precise estimates requiring low-complexity calculations. The double exponential smoothing (DES) was expanded from the simple ES by Holt in 1957 [19]. The DES contains a *level* variable S_t and a *trend* variable b_t to model data that shows a trend fluctuation. DES can be basically concluded through the Formulas 1, 2 and 3.

$$S_t = \alpha Y_t + (1-\alpha)(S_{t-1} + b_{t-1}) \qquad 0 < \alpha < 1 \qquad (1)$$

$$b_t = \beta(S_t - S_{t-1}) + (1-\beta)b_{t-1} \qquad 0 < \beta < 1 \qquad (2)$$

$$\hat{Y}_{t+m} = S_t + mb_t \qquad (3)$$

where S_t is an estimate of the current level of the series, b_t is the average growth of the series. α and β are the smoothing factors which can be set by experience or some parameter optimization algorithms. $\hat{Y}_{t+m}$ denotes the forecasting value of the series at time $t+m$. The initial estimate of level and trend can selected as described in Formulas 4 and 5.

$$S_1 = Y_1 \qquad (4)$$

$$b_1 = Y_2 - Y_1 \qquad (5)$$

From the formulas above, we can see that DES is essentially an weighted moving average of a time series. As DES smooths the series, the weights of the old data decreases exponentially while the newer data is assigned with larger weights. Hence, the predicting results greatly depend on the newer data, which means DES can be implemented with limited historic data. It is noteworthy that the prediction results of DES always is shaped showing certain delay when compared with the actual values since DES is a typical casual system. This delay increases when there are rapid, or sharp, oscillations in the series of values, resulting in a larger prediction error.

3.2 Seasonal Autoregressive Integrated Moving Average

Box and Jenkins explored the characteristics presented in the context that most of the series of values existing in economics and industrial systems exhibit a particular kind of non-stationary behavior. This behavior can be generalized as an autoregressive integrated moving average (ARIMA) model [6]. Since then, ARIMA model has been widely used in time series forecasting and exhibits higher precision than many other statistical models. For time series which present obvious seasonality, such as the traffic flows, the seasonal ARIMA (SARIMA) can be a better prediction model [21]. We use X_t to denote a non-stationary time series, then SARIMA can be expressed as described in Formulas 6, 7, 8, 9, and 10.

$$\phi(L)(1-L)^d\Phi(L)(1-L^S)^D X_t = c + \theta(L)\Theta(L)\varepsilon_t \qquad (6)$$

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \qquad (7)$$

$$\theta(L) = 1 - \theta_1 L - \theta_2 L^2 - \cdots - \theta_q L^q \qquad (8)$$

$$\Phi(L) = 1 - \Phi_1 L - \Phi_2 L^2 - \cdots - \Phi_P L^P \qquad (9)$$

$$\Theta(L) = 1 - \Theta_1 L - \Theta_2 L^2 - \cdots - \Theta_Q L^Q \qquad (10)$$

where L is the lag operator and we have $L^a X_t = X_{t-a}$. d and D represent the order of nonseasonal differencing and seasonal differencing respectively. S denotes the period. ε_t is a Gaussian white noise which has zero mean and constant variance. $\phi_i, i = 1, 2, \ldots, p$ and $\Phi_i, i = 1, 2, \ldots, P$ are the coefficients of the nonseasonal autoregressive part and the seasonal autoregressive part, respectively. $\theta_i, i = 1, 2, \ldots, q$ and

$\Theta_i, i = 1, 2, \ldots, Q$ are the coefficients of nonseasonal moving average part and seasonal moving average part, respectively. The above formulas can be concluded as SARIMA $(p, d, q)(P, D, Q)_S$.

To fit a SARIMA model with a time series, we need to take four main steps [15]:

1. *Model Identification*: Plot the time series and determine the order of nonseasonal differencing d and seasonal differencing D to stationarize the series. For the stationarized series, Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are plotted to indicate the possible polynomial order p, q, P and Q;

2. *Model Estimation*: Use the maximum likelihood method to estimate the coefficients ϕ_i, θ_i, Φ_i and Θ_i. When fitting the series well enough, the model with fewer parameters is preferred;

3. *Diagnostic Checking*: Calculate the residuals and to see if the residuals follows a Gaussian distribution with zero mean and constant variance. If not, the residuals can be applied to modify the model;

4. *Forecasting Validation*: Validate the model with actual observed data.

3.3 K-nearest Neighbor

K-nearest Neighbor (KNN) is a non-parametric prediction model based on similarity measurement between input data and historic data [25]. In the prediction of time series, we define an input data to KNN as an state vector, as described in Formula 11.

$$X_t = \{x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-q}\} \tag{11}$$

where X_t is the state vector and x_t is the instant value at time t.

Then, the similarity between historic data and input data can be measured by the Euclidean distance, represented in Formula 12.

$$d = \sqrt{\sum_{i=0}^{q} (x_{t-i} - x_{h,t-i})^2} \tag{12}$$

Every time an KNN model receives an state vector, it compares the state vector with all the historic data in the database and searches for K nearest vectors. Finally, the forecasting value $\hat{x}_{t+1}$ can be simply expressed as the average of k nearest neighbors, seen in Formula 13.

$$\hat{x}_{t+1} = \frac{1}{K} \sum_{i=1}^{K} x_{h,i,t+1} \tag{13}$$

In this paper, we adopt a more robust approach, which weights the K nearest neighbors based on their distances to the input state vector [27], as indicated in Formulas 14 and 15.

$$\hat{x}_{t+1} = \sum_{i=1}^{K} w_i x_{h,i,t+1} \tag{14}$$

$$w_i = \frac{d_i^{-1}}{\sum_{i=1}^{K} d_i^{-1}} \tag{15}$$

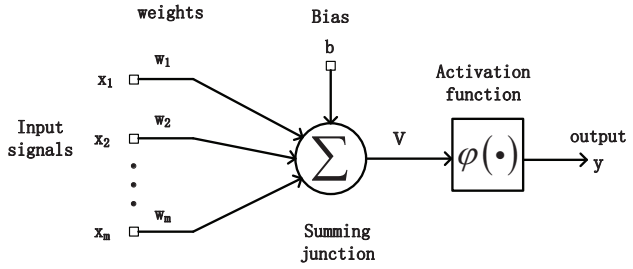

Figure 1: A model of a NN neuron.

where w_i is the weight applied to the ith neighbor, and d_i is the corresponding distance.

The KNN is also called lazy learning algorithm since it stores the training samples directly instead of compressing them into a regression model. This method is intuitive so that it is broadly applied in prediction and classification. It is worth noting that this method requires higher storage requirements than other machine learning methods, and the similarity measurement is highly computational processing demanding as the historic database expands.

3.4 Artificial Neural Network and Back Propagation Algorithm

The artificial neural network (ANN) is motivated by the way our human brain works to learn a certain behavior or conception. The ANN have the functionalities like pattern recognition and classification, so they are widely used as a basis of artificial intelligence. According to the Kolmogorov theorem, a three-layer neural network are capable of modeling arbitrary non-linear functions with high accuracy [25]. Hence, the ANN can be employed as a prediction model for time series especially with a large number of historical data.

A neuron, as depicted in Figure 1, is a basic component of a neural network [10]. The neuron contains a set of synapses, and each of which is characterized by a weight of its own. An input signal x_j at a synapse j is multiplied by its weight w_j. All the weighted input signals and a bias unit b are fed into a summing junction, producing an intermediate signal, as in Formula 16, which is called induced local field.

$$v = b + \sum_{j=1}^{m} w_j x_j \tag{16}$$

Then, a activation function, denoted by $\varphi(.)$ is applied to limit the output of neuron. The sigmoid function is the most common choice of activation function for its differentiability.

A single neuron attempts to build a linear relationship between input signals and output signals. In order to reflect a nonlinear function, at least three-layer neurons are needed to form a neural network. Figure 2 shows a typical three-layer feedforward neural network. Note that neurons in the input layer are actually simple nodes whose outputs remain the same as the inputs. The hidden layer and output layer behave as previously. In order to fit a particular nonlinear mapping, adequate numbers of training sample and an efficient learning algorithm are required to train the network.

The back propagation (BP) algorithm is the most common used learning method for neural network. BP first feedforwards the input signal among the network and produces the corresponding output signal. Comparing the real output with our desired output, an error signal is generated and back propagated from the back end to the front end of net-

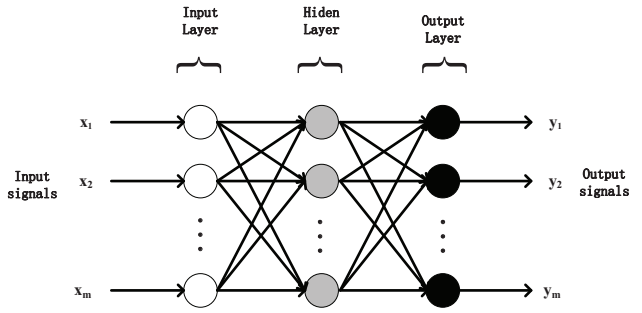

Figure 2: A three-layer feedforward neural network.

work. Based on the structure of neural network in Figure 2, we assume that a training sample consists of an input vector $\{x_1, x_2, \ldots, x_m\}$ and a desired output vector $\{y_1, y_2, \ldots, y_m\}$ while the actual output data is $\{\hat{y}_1, \hat{y}_1, \ldots, \hat{y}_m\}$. Thus, an error function can be defined as indicated in Formula 17.

$$E = \frac{1}{2}\sum_{i=1}^{m}(y_i - \hat{y}_i) \tag{17}$$

During the error back propagation process, a correction on the jth weight of a neuron k can be simply expressed as in Formula 18.

$$\Delta w_{jk} = -\eta\frac{\partial E}{\partial w_{jk}} \tag{18}$$

where $-\eta$ is the learning rate of BP algorithm. Formula 18 shows the core idea of BP algorithm, which is known as steepest gradient method. It adjusts the weights on the network, attempting to minimize the error function during each training process. Moreover, it is noticeable that the learning rate η presents an important role in approximating the minimum error function since it directly influences the correction convergence. Usually, the smaller we set the learning rate η, the smaller are the changes to the weights in the network, which guarantees a more accurate mapping. However, the smaller learning rate indeed slows down the convergence of the network. Increasing the learning rate can effectively speed up the training process while the optimum error function may not be reached, resulting in an unstable network.

3.5 Support Vector Regression

The support vector machine (SVM) is a binary classification model which builds a hyperplane as decision surface to maximize the margin of separation between two linear-separable datasets [10]. The hyperplane is actually a regression function, so SVM can also be applied to regression analysis, which is known as support vector regression (SVR) [13]. Again, we assume that a training sample as an input vector $\{x_1, x_2, \ldots, x_m\}$ and a desired output vector $\{y_1, y_2, \ldots, y_m\}$. Then, the regression function of SVR is defined as in Formula 19.

$$f(x) = \sum_{i=1}^{m} w_i\varphi(x_i) + b \tag{19}$$

where $\{w_i, w_2, \ldots, w_m\}_{i=1}^{m}$ is the weight vector, b is the basis and $\varphi()$ denotes a nonlinear function which reflects the input vector into a higher dimensional feature space. Similar to the neural network, SVR aims to get the optimal weight vector by minimizing a risk function R_{emp} as described in Formulas 20 and 21.

$$R_{emp} = \frac{1}{m}\sum_{i=1}^{m} L_\epsilon(f(x_i), y_i) \tag{20}$$

$$L_\epsilon(f(x_i), y_i) = \begin{cases} |f(x_i) - y_i| - \epsilon & |f(x_i) - y_i| > \epsilon \\ 0 & otherwise \end{cases} \tag{21}$$

where L_ϵ is called $\epsilon - intensive$ loss function. Then, Formula 19 and 20 can be transformed into a convex quadratic optimization problem expressed as presented in Formula 22.

$$\begin{aligned} \min_{w,b,\xi,\xi^*} \ & \frac{1}{2}||w^2|| + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \\ s.t. \quad & f(x_i) - y_i \le \epsilon + \xi_i, \\ & f(x_i) - y_i \ge \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \ge 0. \end{aligned} \tag{22}$$

The ξ_i^* and ξ_i^* are the slack variables representing the allowable deviation between the actual output and desired output. With the constraints above, this optimization problem can be solved by Lagrange multiplier and the final regression function is depicted as:

$$\begin{aligned} & f(x) = \sum_{i=1}^{m}(a_i^* - a_i)K(x_i, x) + b \\ & s.t. \quad 0 \le a_i^* \le C, 0 \le a_i \le C. \end{aligned} \tag{23}$$

where a_i^*,a_i are the Lagrange multipliers, and $K(x_i, x)$ is the kernel function which represents the inner product of $\varphi(x_i)$ and $\varphi(x)$. Common used kernel functions are linear kernel function and polynomial kernel function, radial basis kernel function. Usually, the radial basis kernel function is the first choice, defined as indicated in Formula 24.

$$K(x, z) = exp\left(\frac{||x - z||^2}{2\gamma^2}\right) \tag{24}$$

4. ANALYSIS AND RESULTS

In this paper, our objective is to implement an one-day traffic flow forecasting based on the five prediction models introduced in the previous section. The analysis is realized through comparing the capability of each algorithm in achieving close matching between its estimate and the actual real. This study is conducted through implementing and applying the afore-described models over an available real data sample. Then three different error metrics are used to evaluate the performance of these algorithms.

4.1 Dataset

The machine learning methods for prediction require a large volume of historic data in order to build an accurate model. In this experiment, a large and reliable traffic flow database is provided by Highway England [8]. We randomly choose the traffic volume of three highway segments that was collected on March 2015, whose link reference are AL1000, AL1020 and AL1037, respectively. Figure 3 shows one-week traffic flow of AL1000. It is worth noticing that the traffic flow follows a daily period, as depicted in Figure 3. However, the patterns are a little difference among a weekday, Saturday, and Sunday. Hence, we implemented the forecasting on these three patterns separately. Note that the historic

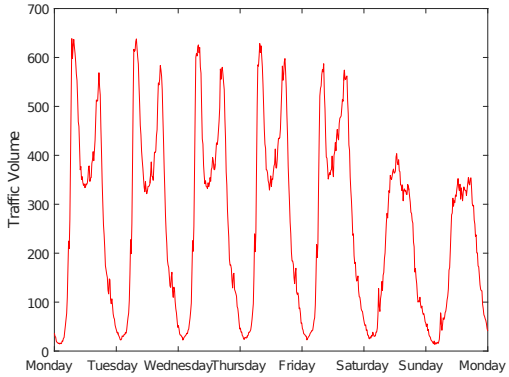

Figure 3: Traffic flow of AL1000 in one week.

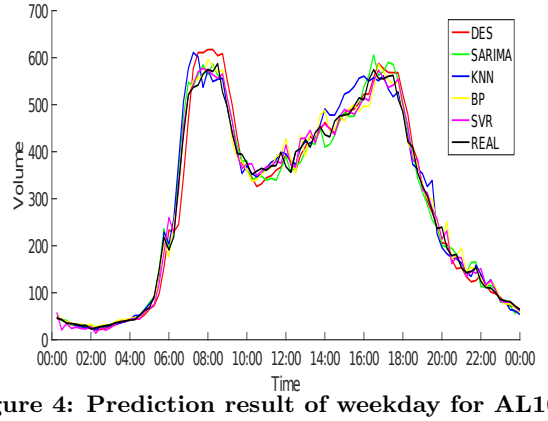

Figure 4: Prediction result of weekday for AL1000.

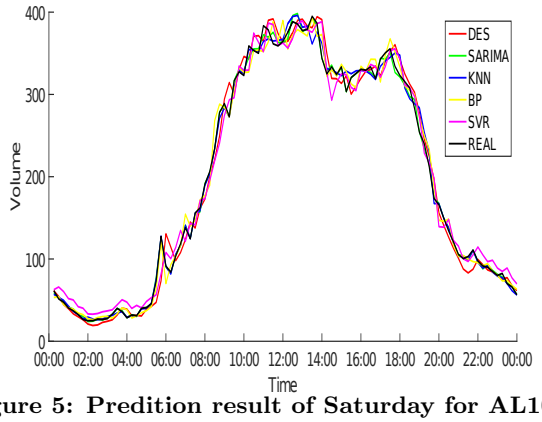

Figure 5: Predition result of Saturday for AL1000.

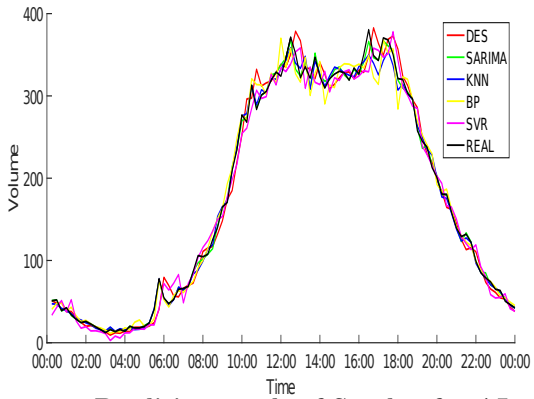

Figure 6: Predition result of Sunday for AL1000.

data for each pattern are equally extracted from their corresponding three previous days on the same day so that the comparison could make sense, regarding the traffic flow load oscillations that follow one-day-time-span commuting seasonality.

4.2 Performance Metrics

Error metrics have been used in order to identify the efficiency of the each model described in this work. These metrics provide an effective view on the accuracy of the employed methods, allowing us to identify the cases in which they are more appropriate for retrieving effective traffic load estimates. In this context, three widely used error metrics are used. They basically encompass the difference between prediction results $\hat{y}_i$ and the actual values y_i. These metrics are defined in Formulas 25, 26, and 27.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 \tag{25}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{26}$$

$$r^2 = \frac{(m\sum_{i=1}^{m}\hat{y}_i y_i - \sum_{i=1}^{m}\hat{y}_i\sum_{i=1}^{m}y_i)^2}{(m\sum_{i=1}^{m}\hat{y}_i^2 - (\sum_{i=1}^{m}\hat{y}_i)^2)(m\sum_{i=1}^{m}y_i^2 - (\sum_{i=1}^{m}y_i)^2)} \tag{27}$$

where mean-absolute-error (MSE) defined in Formula 25 and mean absolute percentage error (MAPE) described in Formula 26 are proportional to the error. The r^2 (R-square) in Formula 27 is inversely proportional to error; it is used to reflect the fitting extent between $\hat{y}_i$ and y_i, so the R-square is always less than 1.

4.3 Results

Figures 4, 5, and 6 are the prediction results of three seasonality patterns for AL1000. From these figures, we can see that almost all the results of these models fit the actual traffic flow well during the low traffic volume period which begins from 0 a.m to 4 a.m. Only the SVR model shows slightly fluctuating curves during this period. Then as the traffic volume increases gradually to peak level, the prediction results of DES and SVR start to display a latency to actual traffic volume while the KNN, BP neural network and SARIMA keep fiting the real curve well. This illustrates that DES and SVR are less insensitive to the trend of traffic volume than other three models during this period. Among the peak hours (e.g. 7 a.m-9 a.m and 14 a.m-18 a.m of a weekday) of these three patterns, all the prediction curves expect for SARIMA deviate actual curve severely and show a delay. The dramatic fluctuation of traffic volume during these periords are rather difficult to forecast becasuse of its stochasticity. Hence, the SARIMA which considers a random entry could have a relatively good predicion while the other four models will have much more prediction errors during the peak hours. Figures 4, 5, and 6 just show the fitting results intuitively and we could not distinguish the performance among these models by these figures. Hence, three metrics for different patterns are collected in Table 1.

From Table 1, the SARIMA model takes dominant position in almost every metric. There is no doubt that SARIMA is the best prediction model among these five models. The

Table 1: Performance Evaluation over Three Patterns

	Weekday			Saturday			Sunday		
Model	**MSE**	**MAPE(%)**	r^2	**MSE**	**MAPE(%)**	r^2	**MSE**	**MAPE(%)**	r^2
DES	844.8669	7.99	0.9805	259.0883	8.57	0.9862	232.5445	9.03	0.9870
SARIMA	297.9257	5.78	0.9922	17.8561	1.69	0.9990	21.7969	1.86	0.9987
KNN	616.8795	6.50	0.9857	69.1885	2.92	0.9960	56.8554	3.96	0.9968
BP-NN	383.3754	5.72	0.9896	231.2599	6.66	0.9868	208.7322	8.18	0.9876
SVR	386.5236	8.22	0.9898	258.3330	12.14	0.9858	232.3257	13.08	0.9865

Table 2: Performance Evaluation over Three highway segments

	AL1000			AL1020			AL1037		
Model	**MSE**	**MAPE(%)**	r^2	**MSE**	**MAPE(%)**	r^2	**MSE**	**MAPE(%)**	r^2
DES	844.8669	7.99	0.9805	1542.5566	8.89	0.9771	443.8291	7.45	0.9851
SARIMA	297.9257	5.78	0.9922	517.5391	6.63	0.9914	329.4267	6.75	0.9874
KNN	616.8795	6.50	0.9857	1646.4914	10.05	0.9752	572.9095	7.88	0.9828
BP-NN	383.3754	5.72	0.9896	952.9764	7.98	0.9840	645.4714	10.67	0.9748
SVR	386.5236	8.22	0.9898	567.5516	8.05	0.9898	220.0043	8.08	0.9917

DES works worse than any other models; however, DES could still be broadly used for its timely prediction and simplicity. Among the three machine learning methods, BP neural network shows a more stable performance since it fits each pattern equally well. The KNN presents the lowest errors for Saturday and Sunday traffic flow prediction, but its performance dramatically worsens for weekday pattern. This particular performance behavior may lie on the fact that the weekday present more fluctuating values than the weekend. It is worth nothing that SVR model shows relatively low MSE and good R-square while its MAPE always remains the highest among these models. Since MAPE is a percentage error, it is sensitive error at at small absolute value. As the analysis above, SVR shows an obvious fluctuation between 0 a.m and 4 a.m resulting in a high MAPE. Consequently, SVR does not forecast well during low traffic volume periods.

The Table 2 shows the performance of these algorithms on different road segments. Again, Table 2 demonstrates that SARIMA and DES are the best and worst prediction methods among these models. Moreover, the KNN shows a worse prediction results on weekdays than on weekends as discussed above. In conclusion, the SARIMA and BP neural network can be two better choice for our envisioned system because of their prediction accuracy and stability.

This paper aims to analyze the prediction accuracy of the presented models for short-term forecasting of traffic flows. However, an extended experiment for long-term prediction give us a broader understanding on short-term prediction. We select three road segments, AL1244, AL1344, and AL1800, and use the traffic flow of 200 consecutive weekdays to conduct the prediction performance analysis. The experiment is designed to fit the prediction model with the first part of total samples and then to test the prediction accuracy with the remaining part of total samples. Since the DES model is a timely forecasting method, we can exclude it in the long-term prediction scenario. Table 3 shows the performance of the other models measured by R-square in 5 different ratios of training samples over the testing samples.

In Table 3, all the models work better as we increase the portion of training samples in the total samples and decrease the prediction range. Among the four models, the BP neural network have the best prediction results while the SARIMA shows the worst results. Since the total samples cover traffic flow over 200 days, uncertain factors, such as traffic accidents, weather, and festivals, lead to many extremely different traffic flow patterns. SARIMA model is more sensitive to these irregular seasonality patterns than the other models, and it thus produces much worse performance than that in the short-term predictions. It is worth noticing that all the models show worst results with ratio 7/3 for road AL1334 since irregular patterns take a dominant part of the testing samples. In conclusion, the short-term prediction is preferred for higher prediction accuracy, and BP-NN can provide reasonable long-term predictions when extensively previously trained.

Table 3: Table 3 caption

	AL1244				
	1/9	**3/7**	**5/5**	**7/3**	**9/1**
SARIMA	0.8755	0.8751	0.8621	0.9072	0.9117
KNN	0.9087	0.9186	0.9062	0.9162	0.9307
BP-NN	0.8693	0.8988	0.9188	0.9346	0.9504
SVR	0.9119	0.9159	0.9095	0.9394	0.9291
	AL1334				
	1/9	**3/7**	**5/5**	**7/3**	**9/1**
SARIMA	0.8863	0.8781	0.8656	0.8529	0.8949
KNN	0.9172	0.9141	0.9041	0.8953	0.9315
BP-NN	0.9059	0.9119	0.9174	0.9065	0.9461
SVR	0.9251	0.9204	0.9148	0.9063	0.9342
	AL1800				
	1/9	**3/7**	**5/5**	**7/3**	**9/1**
SARIMA	0.9181	0.9201	0.9224	0.9306	0.9336
KNN	0.9392	0.9179	0.9366	0.9471	0.9517
BP-NN	0.9146	0.9409	0.9471	0.9543	0.9576
SVR	0.9215	0.9232	0.9278	0.9315	0.9249

5. CONCLUSION

In this paper, we do a beforehand traffic flow prediction for establishing a realistic road traffic system under the context of vehicular cloud. We choose five prediction models including double exponential smoothing(DES), seasonal autoregressive moving average(SARIMA), K-nearest neighbor(KNN), back-propagation neural network(BP-NN) and support vector regression(SVR) to implement. After comprehensive and reliable evaluation and comparison of their performance, SARIMA and BP neural network are selected as forecasting methods for our road traffic system.

For the future work, we will import our prediction traffic volume to SUMO to substitute the randomly generated traffic flow, targeting two main issues. Historic Data Collection:

The current historic data was collected from a single highway segment by sensors. For implementing vehicular cloud, tremendous historic traffic data on a urban area need to be collected for future work. Vehicle Mobility models: Various vehicle mobility models should be considered to model a realistic traffic system. The existing mobility models on SUMO may be not enough, hence we need to conduct a comprehensive literature review on vehicle mobility models.

6. REFERENCES

[1] J. Abdi and B. Moshiri. Application of temporal difference learning rules in short-term traffic flow prediction. *Expert Systems*, 32(1):49–64, 2015.

[2] M. Abuelela and S. Olariu. Taking vanet to the clouds. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '10, pages 6–13, New York, NY, USA, 2010. ACM.

[3] M. A. Al Mamun, K. Anam, M. F. A. Onik, and A. Esfar-E-Alam. Deployment of cloud computing into vanet to create ad hoc cloud network architecture. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pages 24–26, 2012.

[4] M. Aloqaily, B. Kantarci, and H. T. Mouftah. Vehicular clouds: State of the art, challenges and future directions. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*, pages 1–6. IEEE, 2015.

[5] S. Bitam, A. Mellouk, and S. Zeadally. Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. *IEEE Wireless Communications*, 22(1):96–102, February 2015.

[6] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley and Sons, Hoboken, New Jersey, 2016.

[7] H. Cao and F. Han. The urban arterial traffic flow forecasting based on bp neural network. In *In proceedings of the 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control*, pages 393–397, 2014.

[8] H. England. Highways agency network journey time and traffic flow data.

[9] L. Gu, D. Zeng, and S. Guo. Vehicular cloud computing: A survey. In *2013 IEEE Globecom Workshops (GC Wkshps)*, pages 403–407, Dec 2013.

[10] S. Haykin. *Neural Networks and Learning Machines*. Pearson Education, Inc., Upper Saddle River, New Jersey, 2009.

[11] A. G. Hobeika and C. K. Kim. Traffic-flow-prediction systems based on upstream traffic. In *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*, pages 345–350, Aug 1994.

[12] C. C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

[13] W. Hu, L. Yan, K. Liu, and H. Wang. A short-term traffic flow forecasting method based on the hybrid pso-svr. *Neural Processing Letters*, 43(1):155–172, 2016.

[14] M. S. Kumar, V. and N. Chand. Applications of vanets: Present and future. *Communications and Network*, 5:12–15, 2013.

[15] S. V. Kumar and L. Vanajakshi. Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, 7(3):1–9, 2015.

[16] T. Pamuła and A. Król. The traffic flow prediction using bayesian and neural networks. In *Intelligent Transportation Systems–Problems and Perspectives*, pages 105–126. Springer, 2016.

[17] M. K. Sharma and A. Kaur. A survey on vehicular cloud computing and its security. In *Next Generation Computing Technologies (NGCT), 2015 1st International Conference on*, pages 67–71. IEEE, 2015.

[18] B. L. Smith, B. M. Williams, and R. K. Oswald. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4):303 – 321, 2002.

[19] J. Tang, G. Xu, Y. Wang, H. Wang, S. Zhang, and F. Liu. Traffic flow prediction based on hybrid model using double exponential smoothing and support vector machine. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 130–135, Oct 2013.

[20] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344, 2014.

[21] B. M. Williams and L. A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

[22] G. Yan, D. Wen, S. Olariu, and M. C. Weigle. Security challenges in vehicular cloud computing. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):284–294, 2013.

[23] W. Yang, D. Yang, Y. Zhao, and J. Gong. Traffic flow prediction based on wavelet transform and radial basis function network. In *Logistics Systems and Intelligent Management, 2010 International Conference on*, volume 2, pages 969–972, Jan 2010.

[24] D. Zeng, J. Xu, J. Gu, L. Liu, and G. Xu. Short term traffic flow prediction based on online learning svr. In *Power Electronics and Intelligent Transportation System, 2008. PEITS'08. Workshop on*, pages 616–620. IEEE, 2008.

[25] L. Zhang, H. Wei, Z. Li, Y. Zhang, L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong. An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia - Social and Behavioral Sciences*, 96:653 – 662, 2013.

[26] T. Zhang, R. E. De Grande, and A. Boukerche. Vehicular cloud: Stochastic analysis of computing resources in a road segment. In *Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 9–16. ACM, 2015.

[27] Z. Zheng and D. Su. Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. *Transportation Research Part C: Emerging Technologies*, 43, Part 1:143 – 157, 2014. Special Issue on Short-term Traffic Flow Forecasting.