Forecasting of Shared-Use Vehicle Trips using Neural Networks and Support Vector Machines

Ruey Long Cheu*
Department of Civil Engineering, National University of Singapore &
Institute of Transportation Studies, University of California at Davis
1 Shields Avenue, Davis, CA 95616
Telephone:(530)752-3810; Email: rlcheu@ucdavis.edu

Jianxin Xu
Department of Electrical & Computer Engineering, National University of Singapore
4 Engineering Drive 3, E4 #05-34, Singapore 117576
Telephone:+65-6874-2556; Email:elexujx@nus.edu.sg

Alvina G. H. Kek
Department of Civil Engineering, National University of Singapore
1 Engineering Drive 2, E1 #08-20, Singapore 117576
Telephone:+65-6875-5035; Email:alvina@nus.edu.sg

Wei Ping Lim
Department of Electrical & Computer Engineering, National University of Singapore
4 Engineering Drive 3, E4 #05-34, Singapore 117576

Way Ling Chen
Department of Civil Engineering, National University of Singapore
1 Engineering Drive 2, E1 #08-20, Singapore 117576

Paper 06-1458

Accepted for publication in the Transportation Research Record

March 2006

* Corresponding author
Abstract: This paper compares two trip forecasting approaches, namely neural networks and support vector machines, for a multiple-station shared-use vehicle system. The neural networks trained to perform trip forecasting belong to the multilayer perceptron model. Comparative evaluation was made with the least square support vector machines with the radial basis kernel function. The forecasting models were trained or developed with six months, and validated with one month of actual trip data from the Honda Intelligent Community Vehicle System currently in commercial operations in Singapore. Each model was designed to forecast the net flow of vehicles in a three-hourly period in any day within a month, at a particular shared-use vehicle port. The models were developed for and applied to forecast net flow at each port for the entire months of January to June 2004. Results have shown that the multilayer perceptron model has a slightly better forecast accuracy in terms of monthly average absolute error and monthly maximum absolute error.
INTRODUCTION

Multiple-Station Shared-Use Vehicle Systems

Shared-use vehicle systems are slowly emerging as an alternative mode of public transportation. In cities with high population densities, shared-use vehicle systems exhibit great promise in improving mobility, lowering emissions and congestion problems (1). Further details on the history and benefits of such systems may be found in (2) and (3).

With the advent of using information technology to assist in customer use and fleet management, the idea of shared-use vehicle systems started becoming popular among the researchers, operators and vehicle manufacturers in the U.S. during the mid- to late-1990s (4). Several pilot projects were then conducted to gain a better understanding of how to implement and operate shared-use vehicle systems. The IntelliShare (5,6) was one such project that was carried out at the University of California at Riverside. Others such as ZEV.NET, Carlink I and II (7,8) were conducted at the University of California at Irvine and the Bay Area Rapid Transit station in Dublin/Pleasanton, respectively. In Japan, among the recent pilot projects are the Motegi in Tochigi and Crayon in Toyota city (9,10). There are other on-going or completed projects in Europe (11-13). These projects provided insights on users’ response to carsharing and evaluated the potential for it to be operated as a business. They also enhanced the understanding of user behavior and their response to intelligent transportation systems technology.

Intelligent Community and Vehicle System

Honda Motor Co. named its multiple-station shared-use vehicle system as Intelligent Community Vehicle System (ICVS). Honda’s ICVS started its commercial operation in Singapore in March 2002, and markets its system to consumers under the brand name of Honda Dirac (14). ICVS currently has a fleet of 50 environment-friendly Honda Civic Hybrid vehicles operating from 12 ports. The system has 1400 members generating an average of 45 trips/day. Similar to the IntelliShare in California (5,6) and Praxitele in France (13), ICVS in Singapore provides users with the freedom of picking-up and returning vehicles at any station (known as “port” in ICVS) of their choice. A user must specify his or her return port before the commencement of the trip, but may change the destination port at any time during the trip. The user is also not required to state his/her return time. These features provide convenience and flexibility to the users, but burden the operator with the added cost of relocating vehicles to prevent under or over supply of vehicle inventory at any port in the system. The IntelliShare team previously proposed two user-based relocation methods, namely trip splitting and trip joining which successfully reduced the number of relocations required (1). However, these techniques have not been employed by ICVS in Singapore in view of their key selling points to commuters of privacy, simplicity and convenience of the system. Currently, ICVS uses its own staff to relocate the vehicles between ports if necessary. The lack of an expected return time and changeable return port is a further encumbrance to the problem. Vehicles are relocated based on checks on “real-time” vehicle inventory at the ports (checked at regular intervals), without any anticipatory information on how many vehicles will be returned to and rented out at a port. As a result, redundant relocations may be carried out. Forecasting of trip generation and attraction data is thus proposed to preempt vehicle flows in and out of the ports. Accurate trip forecasting can potentially minimize the
number of relocations in ICVS, thus reducing the operating cost. It also enables the operator to more efficiently deploy vehicles at locations that better serve customer demands (trips) and increase user satisfaction.

**Objective of Research and Paper**

The overall objective of this research is to develop trip forecasting models to aid in the estimation of the number of vehicles available in a port at a particular time of a day. An earlier paper (15) has reported the development and comparison of multilayer perceptron (MLP), regression, successive moving average and Holt’s model in forecasting the ICVS trip generation (demand for vehicles at a port), trip attraction (supply of vehicles to a port) and net flow (supply minus demand) at twenty-four rolling and staggered three-hourly time periods (e.g., 0700–1000 hrs, 0800–1100 hrs, 0900–1200 hrs, …, 0600–0900 hrs) at 5 ports for the month of March 2004. The study made use of historical data collected at 5 of the ports from April 2003 to February 2004 to develop models that were subsequently evaluated with March 2004 data. It was found that, among the four models, the MLP delivered the most accurate net flow forecast results. The average error from neural networks ranges between 0.44-0.66 veh/3-hrs while its maximum error ranges between 1.63–2.50 veh/3-hrs. A limitation of the neural network is the requirement for a large set of historical data for training. Although the other three models have lower forecast accuracy, they also have a lower data requirement.

This paper reports the development of support vector machine (SVM), a relatively new forecasting technique, and comparison with MLP in the forecasting of net flow in 24 three-hourly rolling horizon periods. Models were developed for 5 ports (ports #2 to #6 in Honda ICVS’s convention). As a departure from the past research which evaluated only the forecasting results for the month of March 2004 (15), this study performed a more comprehensive evaluation based on six months of forecasting results, from January to June 2004. A MLP and a SVM were trained to forecast net flow for a particular port, at a particular three-hourly period of a day, and for a particular month. For example, a MLP and a SVM were developed to specifically forecast net flow at port 2 from 0700-1000 hrs in any day in January 2004. To develop or train a model to forecast net flow in the present month, usable data collected in the past month was used for model validation while the usable data in the previous six months was used in the training. The accuracy of the net flow forecast was measured in terms of average absolute error, computed from all the three-hourly periods, and maximum absolute error, from all the three-hourly periods.

**MODELS DESCRIPTION**

This section provides a brief introduction to neural networks and support vector machines.

**Neural Networks**

Inspired by the biological neurons within the human brain, neural networks consist of interconnected simple processing elements organized into layers, for the input to the output layers (16). They are trained to learn the relationships between input and output pairs. This makes it highly applicable to this research as the factors identified will act as inputs to forecast future net flow. The key benefit of neural network lies in its ability to capture both the linear and non-linear relationships between the input and output data. Its only drawback is its data intensive nature. That is, the data used to train a neural network needs to cover the all the
possible patterns in the input-output space. There are four basic steps in a neural network’s model development and application:

- Assemble the training data
- Define the network structure
- Train the network
- Simulate the network response to new inputs

A major challenge in trip forecasting for ICVS is that the data is sparse and noisy (i.e., trip frequency is rather limited, and the usage pattern varied). A suitable network identified in a past research (15) is the MLP. A schematic of the MLP is shown in Figure 1. It consists of a single input layer (with \( n \) neurons) where the influencing input variables are entered, and an output later with number of output neurons designed to represent the output of the problem. Several different numbers of hidden layers (each with a different number of hidden neurons) may be used. Only the neurons at the adjacent layers are connected. Figure 1 shows a MLP with a single hidden layer with a fixed number of hidden neurons (say, \( l \) hidden neurons). The input values (denoted by \( X = \{x_1, \ldots, x_n\} \), \( n \) is the number of input neurons) are carried forward to the hidden neurons by the connections, but modified by the connection weights and the so called transfer functions. Mathematically, the output at hidden neuron \( i \) is

\[
\text{out}_i = f\left( \sum_{j=1}^{n} w_{ji} x_j + \theta_i \right) \quad \forall i = 1, \ldots, l
\]  

(1)

where \( w_{ji} \) is the value of the connection weight between input neuron \( j \) and hidden neuron \( i \), \( \theta_i \) is the constant bias value at hidden neuron \( i \), and \( f(\ ) \) is the transfer function. The same process is repeated from the hidden layer to the output layer to obtain the output vector. Neurons in the different layers may have different types of transfer function. In this research, the hyperbolic-tangent, or “tanh” transfer function was used for the \( l \) neurons in the hidden layer. Linear transfer function was used for the neurons in the input and output layers. There was only one neuron in the output layer. Hence, the output value may be denoted by \( y(X) \). By writing \( W_i = \{w_{i1}, \ldots, w_{in}\} \) we may write

\[
y(X) = \sum_{i=1}^{l} \alpha_i f(W_i \cdot X + \theta_i) + b = \sum_{i=1}^{l} \alpha_i \tanh(W_i \cdot X + \theta_i) + b
\]  

(2)

where \( \alpha_i \) denotes the value of the connection weight linking the \( i \)th hidden neuron to the only output neuron, and \( b \) denotes the bias value of the output neuron. For the rest of this paper, the term MLP refers specifically to the multilayer perceptron that has one hidden layer with the “tanh” transfer function and one neuron in the output layer with the linear transfer function.

The term “training” of a MLP refers to setting the weights and bias values to optimally perform the task (in this research, forecasting). In general, the training follows a gradient descent algorithm (16), in which the weights and biases are moved along the negative of the gradient of a performance function towards a near optimal solution.
Support Vector Machines

Support vector machine (SVM) is a recently developed computational technique introduced by Vapnik (17). The SVM was originally developed as a pattern classifier, with the class boundary defined by a hyperplane which is defined based on the concept of structural risk minimization (of classification error) using the statistical learning theory. The SVM learning algorithm directly seeks a separating hyperplane that is optimal by being a maximal margin classifier with respect to the training data. For non-linearly separable data, the SVM uses the kernel method to transform the original input space, where the data is non-linearly separable, into a higher dimensional feature space where an optimal linear separating hyperplane is constructed. Details on the fundamentals of SVM can be found in (17). An application of SVM in incident detection is reported in (18).

Consider a set of training data from two linearly separable classes. Each data point is denoted by $(X_i, y_i)$, where $i = 1, ..., l$, $X_i = \{x_1, ..., x_n\}$ and $y_i \in \{+1, -1\}$. The hyperplane separating the classes is defined by:

$$W \cdot X + b = 0 \quad (3)$$

where $W$ is the $n$-component vector perpendicular to the hyperplane and $b$ is a constant. The SVM defines the hyperplane as the solution of the following optimization problem:

$$\text{Min } L = \frac{1}{2} \|W\|^2 - \sum_{i=1}^{l} \alpha_i [y_i (W \cdot X_i + b) - 1] \quad (4)$$

where $\alpha_i$ is the Lagrange multiplier ($\alpha_i \geq 0$, $i = 1, ..., l$). For non-linear classification, the SVM maps $X \in R^n$ into a higher dimensional space $H$ through a nonlinear function $\Phi: R^n \rightarrow H$. The, $X$ in (3) is replaced by $\Phi(X)$ and $X_i$ in (4) is replaced by $\Phi(X_i)$ respectively. By carefully selecting the $\Phi: R^n \rightarrow H$ function, it may be possible to linearly separate the two classes of data points in the $H$ space by a hyperplane defined by $W \cdot \Phi(X_i) + b$.

Similar to neural networks, the SVM can be used as a pattern classifier as well as an estimator. A variant of SVM called Least Squares SVM (LSSVM) has been developed to find the optimally fitted non-linear regression function $y(X) = W \cdot \Phi(X) + b$ to a set of data $(X_i, y_i)$, $i = 1, ..., l$ (19). The LSSVM approach defines a loss function $J$:

$$J = \frac{1}{2} \|W\|^2 + \frac{1}{2} \gamma \sum_{i=1}^{l} e_i^2 \quad (5)$$

where $\gamma$ is the regularization constant (related to the unit cost of errors) and $e_i$ is the error between $y_i$ and the estimated value from the fitted regression function $W \cdot \Phi(X_i) + b$. The optimization problem is thus transform into

$$\text{Min } L = J - \sum_{i=1}^{l} \alpha_i [W \cdot \Phi(X_i) + b + e_i - y_i] \quad (6)$$
The solutions are represented by the set of linear equations

\[
\begin{bmatrix}
0 & 1^T \\
1 & \Psi
\end{bmatrix}
\begin{bmatrix}
b \\
A
\end{bmatrix}
= \begin{bmatrix}
0 \\
Y
\end{bmatrix}
\]

(7)

where \( Y = [y_1 \cdots y_l]^T \), \( A = [\alpha_1 \cdots \alpha_l]^T \), \( 1 = [1 \cdots 1]^T \), each of them is a \( l \times 1 \) matrix, and \( \Psi \) is a \( l \times l \) matrix with the \((i,j)\) element defined by

\[
\psi_{ij} = [\phi(X_i) \cdot \phi(X_j)] + \gamma^{-1}
\]

(8)

To avoid computational complexity, the term \( \phi(X_i) \cdot \phi(X_j) \) in (8) is replaced by a so called kernel function \( K(X_i, X_j) \). Commonly used kernel functions are polynomial, radial basis and sigmoid functions. In this paper, the radial basis kernel function has been used. The radial basis function is

\[
K(X_i, X_j) = \exp\left(-\frac{||X_i - X_j||^2}{2\sigma^2}\right)
\]

(9)

By selecting suitable values of \( \gamma \) and \( \sigma^2 \), substituting \( \phi(X_i) \cdot \phi(X_j) \) in (8) with (9) and solving (7) for \( \alpha_i \) \((i=1, \ldots, l)\), \( W \) and \( b \), the non-linear regression function may be reduced to

\[
y(X) = \sum_{i=1}^{l} \alpha_i K(X_i, X) + b = \sum_{i=1}^{l} \alpha_i e^{-\frac{||x-x_i||^2}{2\sigma^2}} + b
\]

(10)

Comparing (2) with (10), one would notice some similarities between the MLP and LSSVM. The two equations will be the same if one set all \( \theta_i = 0 \) in (2), and select the same form of transfer and kernel functions. In fact, the SVM with radial basis kernel function is similar to the radial function neural network (17-20). However, the radial function neural network has not been considered in this paper because the outcome of a part research (15) had found that the MLP was the best neural network tested to date, in terms of forecasting accuracy. On the other hand, the “tanh” kernel function may be used to make the LSSVM similar to the MLP with the “tanh” transfer function in the hidden layer’s neurons. However, there are additional conditions concerning the functional form and parameter values, and the solution is not easily found (20). Therefore, the research team had decided that using LSSVM with the “tanh” kernel function was not practical at that point in time. Table 1 compares some of the similarities and differences between MLP and LSSVM.
DATA SET
As mentioned above, to train a MLP or SVM to perform forecast for a particular month, usable data in the past seven months were required, with the earliest six months of data as the training set and the most recent month of data as the validation set. Therefore, to develop forecasting models for six different months from January to June 2004, data dated back from May 2003 were used. During these months, the average number of trips in the Honda ICVS was 23 trips/day. It should be noted that in assembling usable training and validation data, the data collected from the month of December 2003 and January 2004 were excluded, as the trip patterns were influenced by the seasonal Christmas and Lunar New Year holidays.

In a related research using data from April 2003 to February 2004 ([15]), it was found that the net flow at the particular port at a particular three-hourly period is closely correlated with
• the net flow at the same port at the three-hourly period that has just ended in the past one hour
• the net flow at the same port at the same time period in the previous day
• the net flow at the same port at the same time period in the same day of the previous week
Therefore, this research relied on this finding, and used the above three inputs for MLP and SVM.

MODEL DEVELOPMENT
Neural Networks
Each MLP used in this research consists of three neurons in the input layer, one for each input as mentioned above. There is only one neuron in the output layer with a linear transfer function to forecast the net flow in a three-hourly period. There is only one hidden layer. The number of neurons in the hidden layer varied in the different MLPs, and the optimal size was determined during the model development process. That is, MLPs with different number of neurons in the hidden layer were trained with six months (months $t-7$ to $t-2$) of data, and the one that produces the most accurate net flow forecast for the following month (month $t-1$) of validation data was selected to forecast for the subsequent month (month $t$) for evaluation purpose. Given that the evaluation was performed for six months of ICVS operations, the number of MLPs eventually selected to perform forecast at each port were 6 months x 24 three-hourly periods/day=144 per port. In most cases, each training data set consists of 6 months x 30 days/months=180 training data points. The entire MLP training process was implemented in Matlab with the Neural Network Toolbox ([21]). For each MLP, the number of neuron in the hidden layer was varied from 1 to 30. The input and output data was scaled to the range [-1,1]. The scaled conjugate gradient training method with the Levenberg-Marquardt algorithm was chosen for its fast convergence without over-training. The batch training method, in which weights and biases were only updated after all of the inputs and targets are presented once, were used. The concept of early stopping technique was incorporated into the network training process to prevent overfitting of the training data. Post regression analysis was applied after each update of the weights and biases on the forecasted and actual net flow values in the validation data set to obtain the best trained network. The one giving the lowest mean-square error between the network output and target output in the validation data were selected to perform forecast.
Least Square Support Vector Machines

Similar to the MLPs, the number of LSSVMs trained was 180. In LSSVM training, parameter values for $\gamma$ and $\sigma^2$ need to be first specified. A genetic algorithm (GA) was used to tune the values of $\gamma$ and $\sigma^2$. For each possible combination of $(\gamma, \sigma^2)$ the LSSVM was then found by solving (7). The solution process made use of data in the training set (data from months $t-7$ to $t-2$). The LSSVM was then applied to the validation data set (data in the month $t-1$) to evaluate the fitness of the genetic algorithm so as to evaluate the choice of $(\gamma, \sigma^2)$ values.

In the GA, the chromosome was coded as a 26 bit binary string, with 13 bits used to represent each of the parameters in $(\gamma, \sigma^2)$. A population size of 30 was used. The ranges of search was set to $[1,1000]$ at increment of 0.1 for $\gamma$ and $[0.1,100]$ at increment of 0.1 for $\sigma^2$. Roulette wheel selection method was used to pick the parents for mating. The probability of crossover was set to 30%, and the probability of mutation was set to 1%. The concept of elitism was introduced which meant that the best 10% of the individuals in the previous generation were used to replace the worst 10% of the individuals in the present generation. For each individual, the squared error $\sum_{i=1}^{m} e_i^2$ is calculated from $m$ data points in the validation data set and the fitness value is given as:

$$F = \frac{m}{1 + \sum_{i=1}^{m} e_i^2}$$  \hspace{1cm} (9)

The GA evolution was stopped when there was insignificant change in the best value of fitness function for 50 consecutive generations or when the maximum number of generations (set to 300) has been reached.

The GA operations were implemented in Matlab, with the Matlab toolbox called LS-SVMLab (22) used to compute the solution for the LSSVM. Note that, the size of the matrix in (7) depends on the number of data points used in the training set.

RESULTS AND DISCUSSIONS

Three-hourly rolling horizon net flow data of port 2 to port 6 was used and forecasting for the months of January to June 2004 was performed. For the training or development or the MLP and LSSVM models, the previous month’s data was used as the validation set while the prior six month’s data was used as the training set. However, it is to be noted that, because of seasonal factors that affected the usage, months of December 2003 and January 2004 were excluded from the use for training and validation purposes. For the MLPs or LSSVMs, 720 models (= 144 models/port x 5 ports) were developed and applied for forecasting in the appropriate month, port and three-hourly period. The forecasted net flows were compared with the actual usage recorded by the ICVS system in the same port, month, and three-hourly period.

Figure 2 shows a typical plot of net flow generated by MLPs, LSSVMs with the radial basis kernel functions, and the actual net flow for a port in a day. Note that the forecasts for the 24 three-hourly period in a day were made by the different MLPs and LSSVMs developed for the specific time period of the day. It can be seen from the figure that both MLPs and LSSVMs
can capture the patterns of net flow over a day, but with a lag time of one period. Overall, the MLPs generate slightly closer forecasts to the real data compared to LSSVMs.

Figure 3 shows a typical plot of the actual and forecasted net flow over a one-week span. Again, the MLP models display slightly closer forecasts to the actual net flow compared to LSSVMs.

The average absolute error and maximum absolute error, from the forecasts of all the three-hourly periods within each month (hereafter referred to as monthly average absolute error and monthly maximum absolute error respectively), were generated for comparison purposes. Because of the voluminous amount of data generated in this evaluation, only the summarized version is presented in Table 2.

From Table 2, it can be observed that the monthly average absolute error for the LSSVMs with radial basis kernel function ranges between 0.42 – 0.83 veh/3-hr while the monthly maximum absolute error ranges between 1.71 – 3.21 veh/3-hr. In comparison, the MLPs return a lower monthly average absolute error range of 0.38 – 0.73 veh/3-hr and a monthly maximum absolute error range of 1.54 – 2.58 veh/3-hr. The monthly average absolute errors of the LSSVMs are slightly higher than that of the MLPs, but higher differences exist in the monthly maximum absolute errors. The results suggest that, on average, MLPs provide a more accurate forecast than LSSVMs. A possible explanation could be that MLPs are able to better capture the nonlinear relationship between the input data and net flow. Whereas in a LSSVM, attempt to fit the nonlinear patterns between the input and output data is by means of the kernel function. The kernel function first attempts to linearize the relationship between the input and output data by transforming the input vector from its original space into a higher dimensional space (see (8) and (9)). Linear regression is then performed between the output data and the “transformed” input data. Although the LSSVMs with radial basis kernel functions have produced reasonably good results in forecasting net flow, the radial basis kernel function may not be able to sufficiently transform the input pattern to achieve the same level of accuracy as the MLPs.

**SUMMARY**

This research has introduced the application of SVMs (more specifically, LSSVMs with radial basis kernel function) in forecasting new flow of vehicles at share-used vehicle ports at three-hourly rolling horizon intervals. This study has also demonstrated how GA can be used to calibrate parameters in the LSSVMs. Seven hundred and twenty LSSVM models have been developed to forecast net flow at a particular port and in a particular month, over a six month period. The accuracy of the forecasts, in terms of monthly average absolute error and monthly maximum absolute error, has been compared with that of the same number of MLP models which were trained and applied to the same data set used by the LSSVMs. Contrary to past research, this study has performed a more comprehensive evaluation by making use of data collected over a longer period (13 months, from May 2003 to June 2004) and from more ports (five ports). The results indicate that, although the current version of LSSVMs (with the radial basis kernel functions) is not as accurate the benchmark set by the MLPs, the slight difference in the accuracy is encouraging. With this promising initial results, the authors are currently investigating the performance of LSSVMs with different kernel functions, which has the potential to improve the accuracy of the forecast.
ACKNOWLEDGEMENT
This research was funded by Honda ICVS Singapore Pte Ltd. The authors would thank colleagues in Honda ICVS Singapore Pte Ltd for their assistance in data collection. The authors would like to acknowledge Professor Der-Horng Lee and Professor Qiang Meng for their contributions in earlier technical discussions and data analyses. The contents of this paper reflect the views of the authors and does not necessarily indicate acceptance by the sponsor.

REFERENCES
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE 1</td>
<td>Comparison between MLP and LSSVM</td>
</tr>
<tr>
<td>TABLE 2</td>
<td>Summary of Net Flow Forecasting Performance</td>
</tr>
</tbody>
</table>

**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1</td>
<td>Multi-layer perceptron neural network.</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>Forecasted and actual values of net flow at Port 2 in a typical day.</td>
</tr>
<tr>
<td>FIGURE 3</td>
<td>Forecasted and actual values of net flow at Port 2 in a typical week.</td>
</tr>
</tbody>
</table>
### TABLE 1  Comparison between MLP and LSSVM

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>LSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Similarities</strong></td>
<td>Do not explicitly define the non-linear regression function</td>
<td>Based on the principle of least-squares errors</td>
</tr>
<tr>
<td><strong>Differences</strong></td>
<td>The non-linear regression function is defined by the network structure, connection weights and activation functions</td>
<td>The non-linear regression function is defined by the hyperplane in the transformed space, with solution modified by the kernel function</td>
</tr>
<tr>
<td></td>
<td>Solution is to find the optimum number of hidden neurons, and all the weights and biases</td>
<td>Solution is to find the parameter values of the kernel function, support vectors and the Lagrange multipliers</td>
</tr>
<tr>
<td></td>
<td>Near-optimal solution</td>
<td>Globally optimal solution with respect to the kernel function</td>
</tr>
<tr>
<td></td>
<td>Solution by step-by-step gradient descent approach</td>
<td>Solution is solved by quadratic programming</td>
</tr>
</tbody>
</table>
## TABLE 2 Summary of Net Flow Forecasting Performance

<table>
<thead>
<tr>
<th>Month</th>
<th>Port</th>
<th>Average Absolute Error (veh/3-hr)</th>
<th>Average Maximum Error (veh/3-hr)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MLP</td>
<td>LSSVM</td>
<td>MLP</td>
</tr>
<tr>
<td>January</td>
<td>2</td>
<td>0.38</td>
<td>0.44</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.38</td>
<td>0.42</td>
<td>1.58</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.46</td>
<td>0.50</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.57</td>
<td>0.61</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.40</td>
<td>0.42</td>
<td>1.88</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.44</td>
<td>0.48</td>
<td>1.92</td>
</tr>
<tr>
<td>February</td>
<td>2</td>
<td>0.49</td>
<td>0.53</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.39</td>
<td>0.44</td>
<td>1.83</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.57</td>
<td>0.64</td>
<td>2.29</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.60</td>
<td>0.64</td>
<td>2.08</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.40</td>
<td>0.43</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.49</td>
<td>0.53</td>
<td>1.99</td>
</tr>
<tr>
<td>March</td>
<td>2</td>
<td>0.44</td>
<td>0.47</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.56</td>
<td>0.58</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.63</td>
<td>0.71</td>
<td>2.33</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.58</td>
<td>0.67</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.44</td>
<td>0.47</td>
<td>1.63</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.53</td>
<td>0.58</td>
<td>1.98</td>
</tr>
<tr>
<td>April</td>
<td>2</td>
<td>0.49</td>
<td>0.56</td>
<td>2.04</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.56</td>
<td>0.63</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.58</td>
<td>0.61</td>
<td>2.33</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.67</td>
<td>0.73</td>
<td>2.29</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.59</td>
<td>0.66</td>
<td>2.08</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.58</td>
<td>0.64</td>
<td>2.17</td>
</tr>
<tr>
<td>May</td>
<td>2</td>
<td>0.48</td>
<td>0.55</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.50</td>
<td>0.54</td>
<td>1.79</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.60</td>
<td>0.68</td>
<td>2.17</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.67</td>
<td>0.74</td>
<td>2.21</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.55</td>
<td>0.61</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.56</td>
<td>0.62</td>
<td>2.01</td>
</tr>
<tr>
<td>June</td>
<td>2</td>
<td>0.53</td>
<td>0.56</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.47</td>
<td>0.50</td>
<td>1.54</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.71</td>
<td>0.78</td>
<td>2.33</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.73</td>
<td>0.83</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.54</td>
<td>0.59</td>
<td>2.08</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.60</td>
<td>0.65</td>
<td>2.13</td>
</tr>
</tbody>
</table>
FIGURE 1 Multi-layer perceptron neural network.
FIGURE 2  Forecasted and actual values of net flow at Port 2 in a typical day.
FIGURE 3  Forecasted and actual values of net flow at Port 2 in a typical week.