On-line prediction of border crossing traffic using an enhanced Spinning Network method

Lei Lin, Qian Wang, Shan Huang, Adel W. Sadek

Department of Civil, Structural and Environmental Engineering University at Buffalo, the State University of New York, Buffalo, NY 14260, USA
Terra Technology, Inc., 1515 E Woodfield Rd., Ste 104, Schaumburg, IL 60173, USA

ABSTRACT
This paper improves on the Spinning Network (SPN) method, a novel forecasting technique, inspired by human memory which was recently developed by Huang and Sadek (2009). The improvement centers on the use of the Dynamic Time Warping (DTW) algorithm to assess the similarity between two given time series, instead of using the Euclidean Distance as was the case with the original SPN. Following this, the enhanced method (i.e., hereafter referred to as the DTW–SPN) is used to predict hourly traffic volumes at the Peace Bridge, an international border crossing connecting Western New York State in the U.S. and Southern Ontario in Canada. The performance of the DTW–SPN is then compared to that of three other forecasting methods, namely: (1) the original SPN (referred to as the Euclidean–SPN); (2) the Seasonal Autoregressive Integrated Moving Average (SARIMA) method; and (3) Support Vector Regression (SVR). Both classified as well as non-classified datasets are utilized, with the classification made on the basis of the type of the day to which the data items belong (i.e. Mondays through Thursdays, Fridays, weekends, holidays, and game days). The results indicate that, in terms of the Mean Absolute Percent Error, the DTW–SPN performed the best for all data groups with the exception of the “game day” group, where SVR performed slightly better. From a computational efficiency standpoint, the SPN-type algorithms require runtime significantly lower than that for either SARIMA or SVR. The performance of the DTW–SPN was also quite acceptable even when the data was not classified, indicating the robustness of the proposed forecasting method in dealing with heterogeneous data.

1. Introduction

With increasing concerns about traffic congestion and interest in active traffic management, as a means to address that problem, approaches to providing accurate on-line and short-term forecasts of traffic volume have recently been attracting the attention of traffic researchers and operators alike (see Van Lint and Van Hinsbergen (2012) for a comprehensive review). This is because on-line short-term prediction, which focuses on road traffic condition changes in the near future (ranging from 5 min to about 1 h), is key to monitoring system performance and optimizing traffic operations decisions. While different classification schemes of traffic forecasting methods are possible, in this paper we broadly classify them into the following two groups: (1) statistical or time-series approaches; and (2) Artificial Intelligence (AI) based approaches.

With respect to the first group, the Box and Jenkins techniques (e.g., Autoregressive Integrated Moving Average (ARIMA) models) were firstly applied to the field of traffic forecasting by Ahmed and Cook (1979). Since then more and more
advanced techniques from that family have been applied to traffic volume prediction, such as the seasonal ARIMA models (SARIMA) (Williams and Hoel, 2003; Smith et al., 2002), the ARIMA models with intervention x-variables (ARIMAX) (Williams, 2001; Cools et al., 2009), and the combination of Kohonen self-organizing map with ARIMA models (Van Der Voort et al., 1996). In addition to the Box and Jenkins models, other multivariate time series techniques were exploited to increase prediction accuracy, including the state space model (Stathopoulos and Karlaftis, 2003) and the multivariate structural time series model (MST) (Ghosh et al., 2009). Kalman filtering theory was also utilized for short-term traffic forecasting. Examples include its initial examination by Okutani and Stephanedes (1984), the state space based of Stathopoulos and Karlaftis (2003), and the work by Xie et al. (2007) which used a Kalman filter with discrete wavelet decomposition for short term traffic prediction. Most recently, Min and Wynter (2011) adopted a multivariate spatial–temporal autoregressive model (MSTAR) to predict the network-wide speed and volume in real time.

On the AI side, among the most widely used methods are Neural Networks (NNs). Several NN topologies have been utilized in previous studies including the multilayer perceptron networks (MLP) (Smith and Demetsy, 1994; Chang and Su, 1995; Ledoux, 1997), radial basis function networks (Park et al., 1998), resource allocating networks (Chen and Grant-Muller, 2001), and wavelet networks (Chen et al., 2006; Xie and Zhang, 2006). NNs were also sometimes combined with other methods, such as fuzzy sets and genetic algorithms, to develop hybrid and more powerful predictions methods (e.g., Yin et al., 2002; Vlahogianni et al., 2005, and Wei and Chen, 2012). Besides NNs, other AI methods were recently proposed for short-term traffic prediction. Dimitrion et al. (2008), for example, proposed an adaptive hybrid fuzzy rule-based system approach to predict traffic flow in urban arterial networks. Support Vector Regression (SVR) has also recently been exploited for short-term traffic flow prediction. Specifically, Zhang and Xie (2008) compared a v-support vector machine (v-SVM) model with a NN model and concluded that the former performed better. Other examples of applying SVR to traffic prediction include Castro-Neto et al. (2009) and Hong et al. (2011).

Besides the time series models and AI models surveyed above, Huang and Sadek (2009) recently developed a novel forecasting method that attempts to mimic some of the key features of human memory. The method is called the Spinning Network (SPN) method because, as will be explained later in the paper, the method is based on a set of consecutive rings which hold the data items and continuously spin as they receive new data. Huang and Sadek (2009) also tested the SPN on a 5-min traffic volume dataset from the Hampton Roads area Virginia, and showed that the method yielded superior predictive accuracy in comparison to the back propagation NN and the k-nearest neighbor algorithm. More importantly, the SPN only consumed a fraction of the time required by either the nearest neighbor algorithm or the back propagation NN.

In this paper, the focus is on additional testing and improvement of the SPN algorithm on a more challenging short-term traffic forecast dataset than the test dataset utilized in our original paper (Huang and Sadek, 2009). Specifically, the dataset utilized herein is an hourly traffic volume dataset that comes from one of Northern America’s busiest border crossings, namely the Peace Bridge, connecting Western New York in the US and Southern Ontario in Canada. Compared to traffic volume datasets from typical locations, such as the Virginia dataset used in the original study (Huang and Sadek, 2009), border crossing traffic has several unique features (e.g., significant differences between weekday and weekend traffic, sensitivity to special events such as sporting events and national and religious holidays, etc.) and is thus much harder to predict. In fact, the highly non-linear trends in the Peace Bridge hourly volume dataset required introducing new features to the original SPN algorithm, as well as considering classifying the dataset into more homogeneous groups, as will be described in more detail later. The performance of the SPN algorithm was also compared to both the SARIMA model (as a representative of the statistical time series forecasting approach) and SVR (as a representative of the AI-based approach).

The remainder of the paper is organized as follows. Section 2 provides background information about: (1) the original SPN method developed by Huang and Sadek (2009); (2) SARIMA and SVR, the two models used in this study as the benchmarks; and (3) a handful of previous studies that focused on predicting border crossing volume and delays. Section 3 describes the modeling dataset and the statistical tests performed to characterize the dataset and to assess the difficulty level of predicting the time series in comparison to the Virginia dataset utilized in our previous study (Huang and Sadek, 2009). Section 3 also discusses how the dataset was classified into groups to facilitate the forecasting process. Section 4 describes the details of the methodology followed in this study, including a discussion of how the original SPN was enhanced to improve its accuracy in predicting the border crossing traffic volumes. The results from comparing the performance of the both the enhanced and the original SPN to the performance of SARIMA and SVR are presented in Section 5. Finally, the main conclusions from the study are summarized in Section 6, along with a few recommendations for future research.

2. Background

2.1. SPN method

The SPN method is a novel forecasting algorithm originally proposed by Huang and Sadek (2009). The method is inspired by the functionality of human memory in sensing, processing, and predicting the states of the surrounding environment, and attempts to mimic some aspects of human memory including: (1) the fuzzy nature of the information retrieved; (2) the instinctive association of ideas; and (3) the fact that the quality of the information retrieved is a function of the time and effort invested. While the method shares some features with the nearest neighbor approach, one of its key advantages is...
its dramatic computational efficiency compared to other forecasting algorithms including the nearest neighbor algorithm itself.

As shown in Fig. 1., the SPN consists of a set of consecutive rings on which the data items are stored and processed. Data items constitute the fundamental elements in the SPN. They denote the information that a “person” receives or recalls, and can be a vector, a matrix or an image. In the case of traffic volume prediction, a data item would take the form of a vector consisting of the current hourly traffic volume and the volumes collected in previous hours. Each ring has a fixed capacity that determines how many data items it can store. It also has the functionality of merging or consolidating similar data items and forwarding them to the next ring. This merge function has the benefit of saving space on the rings (i.e., reducing the number of data items stored and searched for) and of increasing the information content of data items, since the information content of a merged data item represents an integration of the information contained within the elements that got merged into it.

Each ring is exposed to three windows as also shown in Fig. 1., representing the stages of receiving, consolidating, and outputting information, namely the input window, the To-Next-Ring (TNR) window, and the output window. Specifically, the input window accommodates an arriving data item by either placing it in one of its vacant cells if any, or merging it with the most similar data item in the window in case there is no vacant cell. The TNR window consolidates or merges similar data items and forwards the merged data item onto the next ring. Finally, the output window scans its cells and picks the data item most similar to the new data item entering the SPN as the output (which in this case would represent the predicted hourly traffic volume for the next hour). When outputs are generated from all the rings, they are evaluated again, and the one that is most similar to the new data item is selected as the final prediction of the new data item. The rings spin continuously in a clockwise or counterclockwise direction so that the different parts of the ring are exposed to the input, TNR, and output windows, ensuring that a wide range of historical data items are examined.

As can be seen from the brief discussion above, the two key functions at the core of the SPN data processing are the “compare” function and the “merge” functions. The “compare” function in the original SPN algorithm used the Euclidean distance between two vectors to measure the degree of similarity between two data items. The “merge” function, on the other hand, combines similar data items by averaging their associated values. Given that the more a data item has been merged with others, the more stable and informative its value would be, the function also records the number of times a given item has been merged with others before the current merging, and uses that count as a weight when calculating the average.

### 2.2. SARIMA model

Williams and Hoel (2003) presented the theoretical basis for modeling univariate traffic condition data streams as Seasonal Autoregressive Integrated Moving Average (SARIMA) processes. In addition, Smith et al. (2002) showed that SARIMA performs better than the nearest neighbor algorithm on the single point short-term traffic flow forecasting problem, but is computationally more demanding. Given this, SARIMA was selected as one of the two benchmarking forecasting methods, used herein to evaluate the performance of the enhanced SPN. According to Box et al. (2011), a time series \( \{Z_t|t=1,2,...,k\} \) is generated by SARIMA\((p, d, q) \times (P, D, Q)_s\) if:

\[
\Phi_p(B^s)\varphi_p(B)^q \nabla d Z_t = \Theta_q(B^s)\theta_q(B)\alpha_t
\]

where

- \( B \) is the backshift operator defined by \( B^s W_t = W_{t-s} \);
- \( p, d, q, P, D, Q \) are parameters with integer values;
- \( \Phi_p(B) \) and \( \Phi_q(B) \) are \( p \times p \) lag polynomials;
- \( \varphi_p(B) \) and \( \varphi_q(B) \) are \( q \times q \) moving average polynomials.

![Fig. 1. Spinning Network (SPN) (revised based on Huang and Sadek, 2009).](image-url)
\( s \) represents the length of seasonal cycles;
\[
\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p
\]
is the nonseasonal autoregressive operator of \( p \) order;
\[
\phi_P(B^s) = 1 - \phi_1 B^s - \phi_2 B^{2s} - \cdots - \phi_P B^{ps}
\]
is the seasonal autoregressive operator of \( P \) order;
\[
\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q
\]
is the nonseasonal moving average operator of \( q \) order;
\[
\Theta_Q(B^s) = \Theta_1 B^s - \Theta_2 B^{2s} - \cdots - \Theta_Q B^{qs}
\]
is the seasonal moving average operator of \( Q \) order;
\[
\nabla_s^0 = (1 - B)^s
\]
is the seasonal differencing operator of order \( B \); \( P_d^0 = (1 - B)^d \) is the nonseasonal differencing operator of order \( d \); and \( q \)
is the estimated residual at time \( t \), which is assumed to be identically and normally distributed with mean as zero and variance as \( \sigma^2 \).

A limitation of SARIMA, however, is that the model assumes a linear correlation structure, and hence may not be able to capture the non-linearity inherent in complex real-world problems. AI-based algorithms on the other hand, appear to be better at handling non-linear trends in the data. Given this, we also compare the performance of SPN to SVR, which we briefly introduce in the next section.

2.3. Support Vector Regression model

The SVR model shares several advantages of the Support Vector Machine (SVM) concept, a popular machine learning method based on statistical learning theory proposed by Vapnik (1999). SVM embodies the structured risk minimization principle and attempts to minimize an upper bound of the generalization error (as opposed to the observation error which traditional regression analysis attempts to minimize). While the SVM concept was initially designed for solving classification problems, the introduction of Vapnik’s insensitive loss function in 1997 allowed its extension to nonlinear regression problems, resulting in the SVR method (Kim, 2003; Pai and Hong, 2005).

Consider the case of a set of data points \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \subset \mathcal{X} \times \mathbb{R} \), where \( \mathcal{X} \) denotes the space of the input patterns and \( m \) is the total number of training samples. In \( \varepsilon \)-SVR, the goal is to find a function \( f(x) \) that has at the most \( \varepsilon \) deviation for all data point and is as flat as possible. If we are to consider first the simplest case of a linear function, that function can be stated as \( f(x) = \langle \omega, x \rangle + b \) with \( \omega \in \mathcal{X}, b \in \mathbb{R} \), and where \( \langle \cdot, \cdot \rangle \) denotes the dot product in \( \mathcal{X} \), and \( b \) is a scalar threshold. In that context, a flat function can be achieved by minimizing \( \varepsilon \) or its norm. Furthermore, because it may not be feasible to achieve this, the slack variables, \( (\xi_i^+), (\xi_i^-) \), are introduced to allow for some flexibility in optimizing the objective of the regression (i.e., in solving the optimization problem). With this, the SVR problem can be formulated as shown in Eq. (2) below:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \omega^T \omega + C \sum_{i=1}^{m} (\xi_i^+ + \xi_i^-) \\
\text{Subject to} & \quad (\langle \omega, x_i \rangle + b) - y_i \leq \varepsilon + \xi_i^+,
\end{align*}
\]

where, \( \varepsilon \geq 0 \) is the insensitive loss function, representing the maximum deviation allowed; \( C > 0 \) is the penalty associated with excess deviation during the training; and \( \xi_i^+, \xi_i^- \) are the slack variables corresponding to the size of the positive and negative excess deviation beyond the \( \varepsilon \) threshold, respectively.

It should be noted that, in the process of solving this optimization problem, SVR achieves nonlinear regression by mapping the training samples into a high dimensional kernel induced feature space, followed by linear regression in that space. Since the kernel mapping is implicit (depends only on the dot product of the input data vectors), it is possible to map the data to a very high dimension, and still keep the computational cost low. A Radial Basis Function (RBF) \( K(x_i, x_j) = \exp (-\gamma |x_i - x_j|^2) \)
is one common kernel function. The parameters of the SVR models, namely the penalty factor \( C \) and the \( \gamma \) in the kernel function, are often optimized using the \( k \)-fold cross validation method (Stone, 1974; Chang and Lin, 2011).

2.4. Border crossing studies

While an extensive literature on the short-term traffic forecasting problem exists as surveyed above, only a handful of previous studies could be found which considered the specific case of modeling traffic flow conditions at border crossings. Moreover, most of those previous studies focused on modeling border delay for off-line planning applications, and not for on-line prediction which is the focus of this paper. Examples include a study by Paselk and Manerering (1994) that used duration models and a study by Lin and Lin (2001) that proposed a delay model for planning applications. More recently, Kam et al. (2005) described the development of a NN for predicting border crossing delays. However, the data used to develop the model came solely from a simulation model, and not from real-world observations as is the case in the current paper.
3. Data sets

This study used the hourly passenger car traffic volumes collected at the Peace Bridge, one of the busiest Niagara Frontier border crossings to test SPN, focusing on traffic entering the US from Canada. Traffic counts observed in 2009 and 2010 were downloaded from the Buffalo and Fort Erie Public Bridge Authority’s website (Buffalo and Fort Erie Public Bridge Authority, 2010). The quality of the data appeared to be excellent with very few hourly traffic counts missing (only 9 out of 8760 in 2009 and 7 out of 8760 points in 2010). Given that only a handful of data are missing, the study did not feel the need to utilize more elaborate imputation algorithms such as those described in (Smith et al., 2003). Instead, a simple method was used to fill the missing data, taking the average of the traffic counts for the hour before and the hour after the missing value.

3.1. Statistical characterization of the data set

In this section, we assess the statistical characteristics of the Peace Bridge data, as compared to the Virginia data used in our original SPN study (Huang and Sadek, 2009). The measures used include: (1) complexity measures, such as the approximate entropy measure, and (2) non-linearity measures, such as the time reversibility of surrogate data. These measures had been previously used by Lin et al. (2013) to help in data diagnosis and in evaluating the predictability of a given time series or data set.

3.1.1. Characterizing the complexity of the data sets

Approximate entropy (ApEn) is a measure for assessing the magnitude of irregularity or unpredictability of fluctuations in a time series. Specifically, the measure denotes the likelihood that fluctuation patterns of a series do not repeat over time. Small values of ApEn usually indicate a predictable dataset with repetitive patterns, whereas larger values of ApEn indicate more randomness. The first step in calculating the approximate entropy value is to make a time-delay reconstruction of the phase or state space of the time series in which to view the dynamics of the system (Jayawardena et al., 2002). Following this, the analysis needs to specify the threshold for the similarity criteria, r, which defines whether two patterns are similar. The details of how ApEn is calculated can be found in the reference (Ho et al., 1997).

The approximate entropy values for both the Peace Bridge and the Virginia data sets were calculated assuming that, if the Euclidean distance between two vectors is lower than 20% of the standard deviation of the data (the value for the threshold r) the two patterns are similar. The results were an approximate entropy value of 0.30 for the Peace Bridge, vs. a value of 0.21 for the Virginia dataset. Given this, it can be concluded that the Peace Bridge data is more unpredictable with fewer recurring or repetitive patterns, and hence is more challenging to predict than the Virginia data.

3.1.2. Characterizing the non-linearity of the data sets

For assessing the non-linearity of the datasets, the method of surrogate data was used. That method tests the nonlinearity of a time series by verifying whether the series is consistent with the null hypothesis of a linear Gaussian process (LGP). A process known as the iteratively Amplitude Adjusted Fourier Transform (IAAFT) is first used to generate surrogate datasets from the original time series (Schreiber and Schmitz, 2000). This process is repeated several times, and after each time, a measure known as the time reversibility, r, which measures the asymmetry of a series under time reversal, is calculated (Schreiber and Schmitz, 2000; Diks et al., 1995). The distribution of the values obtained for the time reversibility index, is then plotted. Simultaneously, the time reversibility value for the original time series, r0, is also calculated (this would be a single value). If the value of r0 is found to be within the distribution of r, it can be judged that the time series is linear; otherwise, it is nonlinear (Merkwirth et al., 2002).

Fig. 2 shows the results of carrying out the surrogate data test. In that test, 1000 replicates of surrogate data were generated, and for each surrogate data set, the time reversibility value, r, was calculated and a distribution of r was generated. The time reversibility value of the original time series, r0, was also calculated (this is the location of the straight vertical line in each of the plots in Fig. 2.). As can be seen, while the Virginia dataset appear to exhibit linear patterns (r0 lies within the distribution of r), the Peace Bridge dataset clearly exhibits nonlinearity. The Peace Bridge data set is therefore more challenging to predict, especially if the forecasting method is primarily designed to deal with a linear time series (e.g., SARIMA).

3.2. Data classification

Given that daily traffic patterns vary significantly by type of the day (i.e. weekday vs. weekend vs. a holiday, etc.), one approach to dealing with the complexity and non-linearity of the Peace Bridge data set, and hence to improve the accuracy of traffic volume prediction, is to develop separate prediction models for distinct patterns or day types based on a logical classification scheme. To do this, we first considered the group of what may be called “ordinary days”, defined as all weekdays excluding Fridays, holidays, and game days. We then calculated the mean hourly traffic volume for each hour of an “ordinary day” for the Peace Bridge dataset, and defined an interval of ±15% of the average hourly volume (15% was chosen based on what may be regarded as acceptable prediction accuracy for the models). The traffic patterns of the “special” days...
(i.e., holidays and game days) were then compared to the "ordinary days" to determine whether they differed enough to warrant having their own groups (i.e., whether they lied within the ±15% band or not).

Fig. 3a. and b shows how the traffic patterns during two holidays (one Canadian and one US) differ significantly from the pattern of an "ordinary day." As can be seen, the holiday hourly volumes clearly fall outside the 15% band. Fig. 3c. also shows that the traffic pattern on the day with a major sporting event (i.e., the Buffalo Sabres game in this case) experienced a pro-
nounced increase in traffic volume. Finally, Fig. 3d. confirms the expectation that Fridays and weekends have significantly different traffic patterns compared to Mondays, Tuesday, Wednesdays and Thursdays.

Based on the above observations, six different data groups were defined: (1) weekdays excluding Fridays (a total of 181 days in 2009 – only 2009 data were used for this group since they are sufficient enough for the analysis); (2) Fridays (35 days in 2009 and 37 days in 2010); (3) Saturdays (38 and 41 days in 2009 and 2010, respectively); (4) Sundays (a total of 83 days in 2009 and 2010); (5) game days (a total of 98 days in 2009 and 2010); and (6) holidays (a total of 42 days in 2009 and 2010).

The different forecasting methods considered in this study (i.e. SPN, SARIMA and SVR) were developed and tested using both the original unclassified and the classified datasets. For the unclassified case, one model was developed for the whole data set, whereas for the classified case, a separate model was developed for each of the first five groups listed in the previous paragraph (i.e. weekdays except Fridays, Fridays, Saturdays, Sundays, and game days). Given that the holidays’ group had rather small data size, it was not considered in this study. By comparing the performance of the forecasting methods on both the classified and unclassified data set, the study hoped to better understand the robustness of the different forecasting methods and the necessity of data grouping for each method. One final observation regarding the data that needs to be mentioned is that the data used in the study included instances of traffic flow under non-recurrent events (e.g. accidents, emergencies, inclement weather, etc.); no attempts were made to screen or exclude those data points. The following section will describe the process of model development and evaluation.

4. Model development
4.1. Enhanced SPN

Refinement of the original SPN algorithm was deemed necessary to allow it to handle the increased complexity and the non-linearity character of the Peace Bridge dataset, compared to the Virginia set used in the original study. As discussed before, both the “compare” and the “merge” functions of the SPN involve assessing the similarity between data items. The most straightforward way of evaluating the similarity of two data items is to calculate the Euclidean distance between them, and this was the approach implemented in the original SPN (referred to hereafter as the Euclidean–SPN). However, the Euclidean distance is a brittle distance measure that is incapable of dealing with elastic timing shifts in the time series data. For example, in the case of the two time-dependent sequences shown in Fig. 4., sequence Y shares similar patterns with sequence X although its peak timings (or peak spans) are shifted (or distorted). If the Euclidean distance is used as the similarity metric, sequences X and Y would be regarded as quite different, and the similarity between the two patterns would go unnoticed.

To solve this issue, an alternative algorithm, called Dynamic Time Warping (DTW), has recently been proposed as a similarity metric between time series sequences. DTW explores every possible time alignment to pair the elements of the two sequences, and then seeks the best pairing that returns the minimum distance. In comparison to the Euclidean distance based measure, it is much more robust and allows similar shapes to match even if they are out of phase in the time axis (Keogh and Ratanamahatana, 2005). The DTW distance is calculated in the following manner (Müller, 2007).

Given two time-dependent sequences $X = [x_1, x_2, x_3, \ldots, x_N]$ and $Y = [y_1, y_2, y_3, \ldots, y_M]$, an $N \times M$ accumulated distance matrix $V(N,M)$ is constructed, with its entries calculated as in Eq. (3). The last entry of the matrix, $v(N, M)$ records the minimum distance associated with the best time alignment between X and Y, and is defined as the DTW distance of the two sequence, i.e., $DTW(X, Y) = v(N, M)$. As can be seen, Euclidean distance is a special case of DTW when the fixed pairing $\{(x_1, y_1), (x_2, y_2), \ldots, (x_{\min(N,M)}, y_{\min(N,M)})\}$ is chosen for distance calculation.

$$
v_{i,j} = \begin{cases} 
\sum_{k=1}^{j} d(x_i, y_j), & i = 1, j \in [1, M] \\
\sum_{k=1}^{i} d(x_i, y_j), & i \in [1, N], j = 1 \\
d(x_i, y_j) + \min \{ v_{i-1,j}, v_{i-1,j-1}, v_{i,j-1} \}, & \text{others}
\end{cases}
$$

where

![Case A: shifted peak timings](image1)

![Case B: extended peak span](image2)

Fig. 4. Similarity between time-dependent sequences.
\(d(x_i, y_j)\) is the distance between \(x_i\) and \(y_j\), it could be \(d(x_i, y_j) = |x_i - y_j|\) or \(d(x_i, y_j) = (x_i - y_j)^2\).

Different from Huang and Sadek (2009) where the focus was on the Euclidean distance, this paper tested both the Euclidean distance and the DTW distance when developing the SPN model. The resulting models are labeled as the Euclidean–SPN (or Eu–SPN for short) and the DTW–SPN respectively, and are compared later to see which similarity measure performs better.

### 4.2. SPN parameters

As discussed in Section 2.1, the SPN has several parameters that may be tuned in order to improve performance. In this study, values of those parameters were determined primarily through experimentation. This involved changing the value of a given parameter until the value that yielded the best performance or the lowest Mean Absolute Percent Error (MAPE) was identified. The parameter tuning process and results are briefly discussed below.

#### 4.2.1. Data item length

The length of the data items refers to the number of elements included in the input vector (referred to as a data item in SPN). In the case of the border crossing traffic prediction, the best prediction results were obtained when the input vector included the current hourly traffic volume along with the hourly volumes for the previous 18 h. This meant that the total length of the SPN’s data items had to be set to 20, with the first 19 elements constituting the input information for the prediction problem (called the historical span) and the 20th element representing the predicted traffic volume for the next hour (i.e., the prediction span). In other words, the hourly data elements in the historical span of a data item are used to evaluate whether two data items are similar or not. If the two data items are deemed similar, the known volume of the future hour in one data item can be used to predict the 1-h future volume of the other.

#### 4.2.2. Number of rings and ring size

In the SPN, the rings hold the data items and serve, through the “merge” function, to consolidate similar data items, thereby making them more stable and informative. With respect to the number of rings, the SPN was found to perform the best on the border crossing dataset when the number of rings was set to four. Regarding the size of the rings, a general observation about the SPN is that the smaller the capacity of the rings the greater is the pressure to merge and consolidate data items, mainly because of the ring space constraint (i.e., each ring has a pre-defined capacity). In our study, after some experimentation, we set the size of the outer ring as 6000 slots or data items, with the size of each inner ring being 10 slots less than the size of the ring immediately preceding it.

#### 4.2.3. Input, TNR, and output windows’ sizes

As mentioned before, there are three windows in the SPN model, namely the input window, the To-Next-Ring (TNR) window, and the output window. To increase precision in this study, the size of the input and output windows was set to be equal to the full size of the ring. The size of the TNR window was arbitrarily set to 10% of the size of the ring.

#### 4.2.4. Spinning speed

Experimentation with different spinning speeds for the rings showed that the rings’ spinning speed did not have a major impact on the SPN performance (this in fact should be expected because the size of the input and output windows was set to be equal to 100% of the ring size). Given this, the spinning interval was arbitrarily chosen as 4 ms (millisecond) for the outermost ring, with the interval increasing by 1 ms for each inner ring.

#### 4.2.5. Threshold to Next Ring (TTNR)

The “Threshold to Next Ring (TTNR)” is a parameter associated with the TNR window. It denotes the minimum number of similar data items that need to be identified within the TNR window before a data “merge” can be conducted. This ensures that only common enough patterns are combined and forwarded to the next ring. A large value of TTNR would reduce the frequency of the “merge” operations (since merging would only happen in that case when a large number of similar data items are identified within the TNR window), and the outer ring would thus fill up quickly. In this study, the threshold was set to two data items.

#### 4.2.6. Distance tolerance

The “distance tolerance” is a parameter associated with the merge function, which specifies how close two data items need to be in order to be merged. That parameter was found to have a significant impact on the SPN performance. Large values would encourage unnecessary data merging while small values would prevent merging. After some experimentation, a value of the distance tolerance parameter equal to 60 vehicles/hour (vph) was found to achieve a good balance. Here, the 60 vph is around 15% of the average hourly traffic volume in the dataset.
4.3. SARIMA model

For benchmarking, the Statistical Package for Social Sciences (SPSS) was used to build SARIMA models, with a model developed for each of the five day groups defined before. The seasonal cycle for each model was set to 24, corresponding to 24 h in a day. Since the SARIMA models are typically used for off-line modeling, a simple procedure was developed to enable the SARIMA models for on-line prediction. This procedure fits a SARIMA model first using a training data set, and then continuously recalibrates the model when new data come in. When recalibrating the model, the most recent observation is added to the training data set, and the first or oldest data point in the training data is dropped to keep the computational burden manageable. This process, which was automated using a syntax file in SPSS, results in what may be viewed as a moving window that updates the training data timely for model calibration. Note that although there are more mathematically rigorous methods available to do so (e.g. a state-space representation of the SARIMA model coupled with a Kalman Filter (Shekhar and Williams, 2008)), the simple procedure described above was deemed adequate for the purposes of this study, and the required computation time is the minimum.

To determine the appropriate length of the training data set, different lengths were tried, and for each length, the prediction errors for a test dataset consisting of 37 days (about 20% of the whole time series available for the weekday group) were calculated. As the results show (Fig. 5.), for the weekday group, a length of a training dataset of 960 h (or 40 days) yielded the best performance in terms of the MAPE. The same procedure was applied to develop the SARIMA models for the other four classes (i.e. Fridays, Saturdays, Sundays, and game days) and the whole unclassified data.

4.4. SVR model

To serve as a second bench-marking model, a SVR model was developed. In that model, the value of the insensitive loss function ($\varepsilon$) was set as 0.01, and the Radial Basis Function (RBF) was chosen as the kernel function. As opposed to SARIMA, SVR can be easily adapted for on-line prediction. Specifically for the Peace Bridge data, the input data vector to the SVR model at a given time step, defined as $X(t) = [x(t), \ldots, x(t - B + 1)]^T$ with the length as $B$, is used to predict the next hourly traffic in the series, $X(t + 1)$. In other words, the model always uses the most recent $B$ – hourly traffic volumes to predict the next hour.

The optimal values for the cost factor $C$ and the gamma parameter of the RBF were determined using a training dataset. Moreover, to improve accuracy, the SVR model was re-calibrated (i.e., new values for $C$ and gamma were determined) after every $P$ hours ($P = 100$ h in this study), keeping the size of the training dataset fixed. The SVR model was recalibrated every 100 time steps (and not after each prediction as was the case with the SARIMA) mainly because the calibration process is computationally intensive and cannot be practically performed after each prediction. For a test period of 10 days in the weekday group, the best values for the length of the input vector ($B$) and the size of the training dataset ($O$) for calibrating $C$ and gamma were found to be equal to 6 and 1440, respectively (see Fig. 6.). For the other classified groups and the whole unclassified dataset, the same procedure is conducted.

![Fig. 5. MAPE of SARIMA with respect to training dataset length.](image)

![Fig. 6. MAPE of SVR with respect to input vector length $B$ and training dataset length $O$.](image)
5. Evaluation results

In this section, we compare the prediction performance of the four models (i.e. DTW–SPN, Euclidean–SPN, SARIMA, and SVR) on both the classified dataset (i.e. the one divided into five groups) and the unclassified set. The models were tested on the hourly volumes from 7:00 to 21:00 of each day because the night hours (22:00–6:00) are of little interest due to their low traffic volumes (e.g., less than 100 vph). The test dataset included 127 valid days (1905 h). For the classified case, the test set was itself broken into the corresponding groups as follows: Weekdays (Monday–Thursday) (525 h), Fridays (180 h), Saturdays (285 h), Sundays (345 h), and Game Days (570 h). The discussion of the results from testing on the classified datasets is presented first, followed by the results from testing on the unclassified or the whole data set.

5.1. Comparisons of the four models based on the classified dataset

Fig. 7 plots the MAPE of the four models when tested on the five groups. As can be seen, the DTW–SPN outperformed all other three methods for almost all the data groups. The only exception was the game day category where SVR performed the best closely followed by DTW–SPN. The superiority of DTW–SPN, compared to the other models, is also confirmed when one calculates the average MAPE for all the five groups. As can be seen also from Fig. 7, DTW–SPN had the lowest average MAPE at 9.84%, followed by SVR at 10.94%.

The Euclidean–SPN, on the other hand, did not perform that well and its MAPE was invariably higher than the two benchmarking algorithms (i.e., SARIMA and SVR). One reason behind this could be that the parameters of the SARIMA and SVR were re-calibrated and tweaked for each data group, whereas the SPN’s parameters were only calibrated once.

The results also appear to confirm the known strengths and weaknesses of traditional time series models such as SARIMA, compared to AI-based methods such as SVR. SARIMA is good at handling linear data sets such as weekday traffic that exhibits strong seasonality and trends, but is challenged when dealing with non-linear patterns (e.g., the game day group). SVR, on the other hand, appears to be capable of capturing the nonlinear patterns, particularly for the days that involve significant volume fluctuations such as game days.

The robustness of the four methods, or more specifically the capability to deal with the sudden changes in traffic volume levels, was also tested. To do this, we first identified what we refer to as, hours with abrupt traffic volume changes or hourly traffic volumes that are dramatically different from the volume in the previous hour. Specifically, the hours with an abrupt change were defined as those with volumes that are greater than 2.5 times (or lower than 0.4 times) of the preceding hourly volume. As found in Table 1, there are 36 abrupt points in total and 26 of them come from the Game Days group.

As can be seen from Table 1, SVR outperforms the others in estimating abrupt hourly traffic volumes, except for the Sunday group. DTW–SPN ranks as the best for the Sunday group, and is the second best for the other groups. In contrast, Euclidean–SPN and SARIMA are not good at all at predicting these abrupt traffic volumes. These results demonstrate the comparable ability of DTW–SPN in capturing abrupt changes in traffic volumes although SVR is still the best in general.

To assess the prediction performance of the models in more detail, the MAPEs of each model with respect to different levels of traffic volumes in the Friday group are shown in Fig. 8a. As can be seen, DTW–SPN performs the best for almost all traffic volume levels when the volume is greater than 250 vph. For low traffic volumes such as 151–200 vph and 201–250 vph, SARIMA or Euclidean–SPN tends to be the best. To provide a different view of the models performance, the predicted vs. actual hourly traffic volumes were plotted for a sample of 60 consecutive hours for the Friday group.
Consistent with the previous observations, DTW–SPN and SVR outperform Euclidean–SPN and SARIMA in estimating hourly traffic volumes.

Finally, the plots of the models’ predictions against the hourly traffic volume observations are shown in Fig. 9. As can be seen, the linear fitting curve associated with DTW–SPN has the highest $R^2$-square (0.89), implying that DTW–SPN is the best model with the highest prediction accuracy among the four. The SVR model comes as a close second with an $R^2$-square value of 0.86.

5.2. Comparison of the four models based on the non-classified dataset

Besides comparing the models’ performance on the classified dataset, their performance was also evaluated on the unclassified set that did not distinguish among the different day types. The results are shown in Table 2. As indicated by MAPE, the DTW–SPN model once again outperformed all other models, and had the lowest MAPE at 10.60%. The SVR came in second, followed by the SARIMA model and the Euclidean SPN. In thus appears that the use of the DTW distance measure (as opposed to the Euclidean distance) has significantly improved the performance of the SPN, reducing the MAPE from 16.49% to only 10.60%.

Table 2 also shows the MAPE for the 36 h with abrupt volume changes, as defined in Section 5.1 Consistent with the previous findings, SVR yielded the best performance, followed by DTW–SPN. Both methods were much more accurate than either SARIMA or Euclidean–SPN. This confirms the ability of SVR and DRW–SPN to deal with the non-linearity of the traffic volume time series.

Fig. 10 plots the four models’ predictions against the observed data. As can be seen from Fig. 10a, DTW–SPN outperforms the others for the majority of traffic volumes ranging from 100 vph to 800 vph, while SVR performs slightly better for higher volumes. In contrast, the Euclidean–SPN and SARIMA perform worse than both the DTW–SPN and SVR. Fig. 10b provides a zoom-in view of the models’ estimation performance in comparison to the actual observations for a sample data with 60 hourly traffic volumes. Consistent with the general case, DTW–SPN performs the best for most of the data entries, except for some hours such as the 55th hour.
Table 2
Prediction performance of SPN, SARIMA, and SVR for the unclassified data.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAPE (%) for the entire dataset (1905 h)</th>
<th>MAPE (%) for hours showing abrupt changes (36 h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTW–SPN</td>
<td>10.60</td>
<td>27.55</td>
</tr>
<tr>
<td>Euclidean–SPN</td>
<td>16.49</td>
<td>69.44</td>
</tr>
<tr>
<td>SARIMA</td>
<td>16.38</td>
<td>52.95</td>
</tr>
<tr>
<td>SVR</td>
<td>14.57</td>
<td>12.59</td>
</tr>
</tbody>
</table>

Fig. 9. Predictions of four models vs. actual volumes in the Friday group.

Fig. 10. Estimation performances of four models for the whole dataset.
Fig. 11 compares the four models’ predictions with the actual observations in the whole dataset (i.e., unclassified). As demonstrated by both the plots and the fitted regression models, DTW–SPN predictions appear to match the real-world observations the most: the fitting curve of the estimate-to-observation scatter plot has the slope which is closest to one; and the corresponding regression model has the highest $R^2$ as 0.89. Among the remaining three models, Euclidean–SPN and SARIMA have similar performance, and both of them are worse than SVR.

5.3. Impact of data classification

The comparison between the classified case and the unclassified case reveals the role of data classification in improving model performance (the classified case in Fig. 7. vs. the unclassified case in Table 2). When the dataset is not classified, the MAPEs of DTW–SPN, Euclidean–SPN, SARIMA, and SVR were 10.6%, 16.49%, 16.38%, and 14.57% respectively. In contrast, after classifying the data by the day type, the average MAPEs were reduced to 9.84%, 16.33%, 11.86% and 10.94%, respectively. This indicates that classifying the data based on similarities of exhibited patterns (e.g., by day groups) is generally helpful in improving the prediction accuracy of all four models.

However, the more interesting observation in that context is with respect to the magnitude in the improvement of the performance of the SPN, as compared to the other model types, when the data is classified, and what that reveals regarding the robustness of the paradigm. Specifically, for both the DTW–SPN and the Euclidean SPN, the improvement in the performance resulting from classifying the data into groups is significantly less (0.76% for DTW–SPN and 0.16% for SPN) than that for SARIMA (4.52%) or SVR (3.63%). This seems to point out to the superior classification ability inherent in the SPN algorithm itself, even without the external help from classifying the data into different day types. The grouping ability of the SPN is naturally the result of the multiple comparisons and merging processed involved in the different windows and rings of network, which allowed the paradigm to exhibit high predictive accuracy even without the external classification of the data. This additional advantage makes the SPN method more practical for real-world traffic volume prediction since no additional effort is required for pre data classification.

As a further illustration of this last point, Fig. 12 compares the four models’ performance for the two cases of the classified (solid line) and the unclassified (dashed line) against actual volume observations for a sample of 60 data points. As can be seen in Fig. 12, the improvement in the prediction performance of the SPN-based approaches (i.e., DTW–SPN or Euclidean–SPN) due to data classification was significantly less than the improvement for either SARIMA or SVR.

5.4. Running time comparison of the four models

In addition to prediction accuracy, running time is also considered as an important criterion for model comparison. For the SPN models, the computational burden mainly comes from the comparisons between the data items conducted on the three different windows of each ring, including: (1) the comparisons between existing data items in the input window of a
ring and a new data item entering the ring; (2) the comparisons between the existing data items in the output window of a ring and a new data item; and (3) the periodical comparisons among the existing data items on the TNR window before each data merging and forwarding. Given that the testing data length is $N_{te}$ and the sizes of the input window, output window, and TNR window are $w_1$, $w_2$, and $w_3$ respectively, the overall time complexity of the SPN, due to the comparison operations in the three windows of all the rings, can be represented as $O(N_{te} \times (w_1 + w_2 + w_3^3/P))$, with $P$ denoting the frequency of invoking the TNR process (i.e., the number of records in between two consecutive executions of the TNR process). If each comparison takes $T_{com}$, the total running time of SPN is $O(N_{te} \times (w_1 + w_2 + w_3^3/P)) \times T_{com}$. For the Euclidean–SPN model, the runtime of the compare function, $T_{com}$, is $O(B)$, where $B$ refers to the length of a data item. For the DTW–SPN, $T_{com}$ is larger and equal to the distance is calculated based on any random pairing of data elements between the two data items being evaluated. Therefore, the time complexity of the Euclidean–SPN model and the DTW–SPN model are $O(N_{te} \times (w_1 + w_2 + w_3^3/P)) / C3B$ and $O(N_{te} \times (w_1 + w_2 + w_3^3/P)) / C3B^2$ respectively.

For the SVR and SARIMA models, due to the moving window training strategy used for on-line prediction, their time complexity depends on both the training data length ($N_{tr}$) and the testing data length ($N_{te}$). For the SARIMA model, the time complexity associated with each model training is $O(m^3 N_{tr})$, with $m$ being the order of the model (Lu et al., 2010). Therefore, the overall time complexity of SARIMA is $O(m^3 N_{tr} + N_{te}/R) = O(m^3 N_{tr} + N_{te}/R)$, where $m$ refers to the order of the SARIMA model (the $D$ in Eq. (1)) and $R$ refers to the frequency of the re-calibration process (specifically the number of records or data points in between calibrations). In our study, given that the model is recalibrated at every prediction step, $R$ is equal to 1. Similarly, for the SVR model, the time complexity for each training phase is $O(N_{tr}^3)$ (Zhao and Sun, 2010), and the total running time is $O(N_{tr}^3 + N_{te}/R) = O(N_{tr}^3 + N_{te}/R)$ ($R = 100$ for the SVR model).

Generally speaking, the SVR model is the most time consuming prediction method due to the involvement of $N_{tr}^3$ and the large value of $N_{tr}$. The time complexity of the SPN model can be controlled by adjusting the size of the input, output, and TNR windows. Table 3 summarizes the general time complexity of the four models, and shows the specific running times obtained for the unclassified Peace Bridge data set on a computer with 4.00 GB RAM and Intel® Core(TM) 2 Duo CPU.

As can be seen, given the parameter settings described in the model development section, the Euclidean–SPN model is the fastest prediction method with just 1836 s, followed by the DTW–SPN model at 10,656 s. SARIMA required a total run time equal to 39,636 s, whereas SVR needed the longest running time. Based on this, it can be concluded that not only the DTW–SPN yielded the highest overall prediction accuracy (as discussed in Sections 5.1 and 5.2), but it is also significantly more computationally efficient compared to either SARIMA or SVR. Specifically, the runtime for DTW–SPN was about one quarter of the runtime for SARIMA and less than 1/15th of the runtime for SVR.
In terms of future research directions, the authors of this paper are currently working on integrating the border crossing traffic volume prediction methodologies developed herein with a queuing model to develop an overall decision support system for effective border crossing management which is being designed to predictive border crossing delay information. Besides this, the authors plan on conducting additional testing of both the original SPN and the enhanced DTW–SPN to other computer with 4.00 GB RAM and Intel® Core(TM) 2 Duo CPU.

### 6. Conclusions and future work

This study has developed an enhanced SPN which applies the DTW method to assess similarity among traffic volume data. The enhanced SPN was then used to predict hourly traffic volumes at the Peace Bridge international border crossing. The performance of the enhanced SPN (i.e. DTW–SPN) was then compared to three other forecasting methods, namely the original SPN algorithm (Euclidean–SPN) described in Huang and Sadek (2009), SARIMA, and SVR. When developing and comparing the models’ performance, the study considered two cases, classifying the dataset into groups by day type and using the original dataset without classification. Among the main conclusions and lessons learnt from the study are:

1. When the dataset was divided into day groups, DTW–SPN yielded the lowest MAPE for all data groups with the exception of the game day group where SVR performed slightly best. The DTW–SPN also had the overall best performance when the MAPE was averaged over all the groups.
2. DTW–SPN also performed the best when the whole dataset was used (i.e. the data was not broken into groups), and once again had the lowest MAPE. This demonstrates the robustness of the method and its ability to handle non-homogeneous time series to some extent.
3. Euclidean–SPN did not perform very well on the Peace Bridge dataset. One reason could be that its parameters were not re-calibrated for the different data groups. On the other hand, Euclidean–SPN was the most computationally efficient and required a fraction of the processing time needed by SARIMA or SVR.
4. DTW–SPN also appears to be significantly more computationally efficient compared to SARIMA or SVR. Specifically, for the case study considered, the total running time for DTW–SPN was about one quarter of the time for SARIMA and 1/15th the time for SVR.
5. DTW–SPN and SVR are capable of capturing abrupt changes of hourly traffic while Euclidean–SPN and SARIMA performed poorly for these abrupt points. This implies the robustness of DTW–SPN along with SVR.

In terms of being “application ready”, when comparing with SARIMA and SVR, our study shows that the SPN models appear to have several advantages such as computational efficiency, the ability to handle non-classified data sets, and not requiring training procedures. This means the SPN models can provide an efficient, straightforward and transferable method for short-term traffic volume prediction.

In terms of future research directions, the authors of this paper are currently working on integrating the border crossing traffic volume prediction methodologies developed herein with a queuing model to develop an overall decision support system for effective border crossing management which is being designed to predictive border crossing delay information. Besides this, the authors plan on conducting additional testing of both the original SPN and the enhanced DTW–SPN to other data sets to further confirm the prediction accuracy and advantages of the method. Additionally, because the SPN models require the specification of a number of parameters such as: (1) the data item length; (2) the number of rings and ring sizes; (3) the sizes of the input, output, and TNR windows; (4) the spinning speed; (5) the threshold to next ring, and (6) the distance tolerance, guidance is needed on how to best set these parameters. The current study seems to indicate that performance is most sensitive to the data item length and the distance tolerance. However, additional research is needed in order to confirm that.

### References


