Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, October 3-7, 2009

Prediction Aggregation of Remote Traffic Microwave Sensors Speed and Volume Data

John R. Junger, Joseph P. Havlicek, Ronald D. Barnes, and Monte P. Tull

***Abstract*— Short term traffic speed and volume prediction is an important component of well developed Intelligent Transportation Systems and Advanced Traveler Information Systems. In this paper, we examine the use of polled Remote Traffic Microwave Sensors as a data source for aggregate traffic predictors. Clock skew and data loss due to network transience pose significant challenges to integrating polled data into such a predictive system. To overcome these, we present a new interpolation and evaluation scheme for data regularization and predictor generation. A method for evaluating the validity of the test sets is proposed and illustrated in a case study using an aggregate predictor with real traffic sensor data acquired in Oklahoma City.**

I. Introduction

Traffic speed and volume prediction are important components of Advanced Traveler Information Systems (ATIS) and Intelligent Transportation Systems (ITS). Such predictions are useful for formulating travel-time predictions and in optimal route planning algorithms, and also enable dynamic traffic flow management. As a result of recent ITS research, traffic flow modeling and prediction has moved towards aggregated data systems [1]–[5].

Many machine learning paradigms and algorithms are available for traffic flow prediction, including neural networks (NN) [2], [3], [6]–[8], Support Vector Machines [7], Kalman and particle filters [5], and time series methods [9]. For most models of traffic flow prediction there are components that are relatively easy to predict and components which are relatively difficult to predict. In general, prediction performance degrades in proportion to how far into the future it is desired to predict the speed and volume; we refer to this as the prediction interval. The point at which this degradation is no longer acceptable to the application that uses the prediction is called the prediction horizon [4]. The finest time resolution with which a system can generate predictions also affects how the predictions can be used. Intuitively, for example, a metropolitan travel-time predictor would have poor performance if its most precise estimation of traffic flow was for a one-hour interval, since traffic flow during peak hours likely shows appreciable variance on that time scale. Since traffic flow prediction has many specific uses, it is beneficial to maximize the prediction horizon while simultaneously maximizing the fineness of the temporal granularity of the predictions.

Roadside detector networks are often widely dispersed geographically, and they can suffer from data loss because of compromised network reliability which can occur in any number of situations. Polled networks often have a semi-fixed sampling frequency giving time series data with an irregular support (*i.e.*, irregularly spaced sampling times). Data regularization by interpolation or extrapolation then becomes an important aspect of the design and implementation of predictor systems in order to shift the sample times onto a regular lattice. One of the fundamental decisions made during the data regularization process is at which point the sampling times of the interpolated data are skewed too significantly from the original sampling times at which data were actually measured. The First Moment Predictor (FMP) described below is an effective, new, and practical tool for making this design decision.

An FMP estimates the mean (location estimate) of a random variable from a limited set of historic data. Conceivably, there are many ways to generate an FMP; but a computationally simple one is given by the sample mean of the historical data under the naïve assumption that the theoretical mean is globally stationary. The benefits and drawbacks of this technique are considered below in Section II.

Aggregate predictors generate a prediction from two or more component predictors. Many of the predictors mentioned in the second paragraph of this section have been used to construct aggregate predictors. Aggregating prediction types, and specifically aggregating predictors that are well-adapted for different types of data sets, tends to yield tangible gains in prediction performance [1].

In this paper, we look specifically at the benefits of generating FMPs for data interpolation and the generation of training sets for neural network based speed and volume prediction algorithms. As an illustrative example, we also present a case study where we generate two FMPs. These two predictions are then used as inputs to a Neural Network which serves to aggregate recent historical data and arbitrates between the speed and volume estimates given by the FMPs and NN, all of which are generated using real traffic data acquired in Oklahoma City.

II. First Moment Predictor Design

A first moment predictor is characterized by two parameters. The *time interval* is the total period of time over which historical data will be considered in formulating the predictor. For example, if predictions are based on data

This work was supported in part by the Oklahoma Transportation Center under grant number OTCREOS7.1-05 and Oklahoma Department of Transportation through State Planning and Research (SPR) Item Number 2212. We want to thank all members of the University of Oklahoma ITS Lab for their development work which has made this research possible.

J. Junger, J. Havlicek, R. Barnes, and M. Tull are with the ITS Laboratory, School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK 73019, USA. {jjunger,joebob,ron,tull}@ou.edu

978-1-4244-5521-8/09/$26.00 ©2009 IEEE

acquired over the last year, then the time interval is one year. The *predictor period*, denoted T, specifies the temporal scope with which the historical data is considered in the prediction algorithm. For example, if traffic volume at 5:00 PM is estimated using historical data acquired at 5:00 PM every day for the last year, then the time interval is one year and the predictor period is one day. Alternatively, if traffic volume at 5:00 PM Tuesday is estimated using historical data acquired at 5:00 PM Tuesday for the last two years, then the time interval is two years and the predictor period is one week.

The temporal granularity is the nominal (time) sampling period of the detector which generates the historical data for the predictor. In terms of the granularity, the predictor period admits an interpretation as the number of sampling intervals that the prediction algorithm considers as distinct random variables. If d_t is the value of the time series acquired from the sensor at time t, then an FMP considers d_{t+kT}, $0 \geq k \in \mathbb{Z}$, as multiple realizations of a single random variable.

Let t be the present time and let the time interval be $[t_i, t_f]$; t_f determines how near to the present time data points will be considered in the estimation algorithm and t_i determines how far into the past they will be considered. If $t_f = t - \Delta_f$ and $t_i = t_f - \Delta_i$, $0 \leq \Delta_f, \Delta_i \in \mathbb{R}$, then the FMP is a moving average (MA) model and may be interpreted as a causal finite impulse response (FIR) digital filter. In this case, the prediction algorithm requires online calculations to be performed in real-time as the data are acquired. Alternatively, if t_f and t_i are *fixed* instants of time in the past, then the predictions can be precomputed and only a table lookup of these precomputed values is required to deliver the predictions in real-time. In this case, we refer to the time interval as *static*. Note that the most recently acquired data can not be considered by the prediction algorithm in the static time interval case.

Let $\hat{d}_t$ be the FMP prediction of d_t. In a practical network of polled sensors, individual data points may be lost for a variety of reasons ranging from hardware failures to buffer overruns to network outages. Let $\mathcal{A}_t$ be the set of time indices τ such that the datum d_τ is supposed to be included in the FMP prediction calculation and such that d_τ is in fact available:

$$\mathcal{A}_t = \{\tau = t + \xi : 0 \geq \xi \in \mathbb{R}, \exists\ d_\tau, \mathrm{mod}(\xi, T) = 0\}.$$

Then the straightforward FMP prediction for d_t is given by

$$\widehat{d_t} = |\mathcal{A}_t|^{-1} \sum_{\tau \in \mathcal{A}_t} d_\tau, \qquad (1)$$

where $|\mathcal{A}_t|$ is the cardinality of the set $\mathcal{A}_t$.

One of the primary disadvantages of an FMP with a static time interval is that it does not react to significant nonstationarities in the data. For example, a static time interval FMP cannot generally react to a lane closure that lasts for several predictor periods or to a work zone associated with a long-term construction project that begins after t_f.

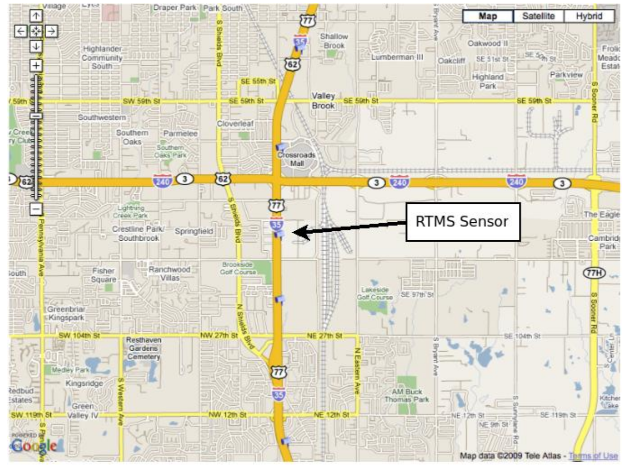


Fig. 1. Physical location of the RTMS sensor used for data acquisition.

III. Neural Network Design Considerations

It is commonly agreed that no one machine learning algorithm or neural network architecture can provide the best performance for predicting traffic speed and volume in all traffic environments [2]. Traffic flow is inherently noisy due to traffic clustering, traffic jams, and shock wave formation among other factors.

Neural networks (NNs) are well suited for modeling non-linear and discontinuous phenomena [10]. NNs have also been shown to perform well in the task of deciding which from among a set of multiple predictions to select at any given time. Thus, in the speed and volume prediction problem an NN may perform two roles [2]. Specifically, an NN might perform as both a primary predictor and as a consumer of primary predictions which are to be arbitrated. However, NNs require training (either online or off) and may also require online training in a deployed environment. This makes NNs a somewhat computationally expensive tool. Fortunately, adaptation cost may be small and it may be possible in many cases to limit the training so that the training cost need only be expended once.

NNs may have any number of layers and any number of neurons in each layer. Increasing these numbers provides a greater capability to model discontinuities and non-linear phenomena. However, since larger networks require longer training periods and increased computation to generate a prediction, there is a trade-off between network size and the expense associated with high computational complexity.

IV. Aggregated FMP Speed and Volume Prediction

A. Data Collection & Regularization

Timestamped speed and volume data were acquired from an EIS Remote Traffic Microwave Sensor (RTMS) located 0.7 miles south of the intersection of I-35 and I-240 in Oklahoma City, as shown in Fig. 1. The time interval was April 4, 2006 through January 10, 2007. The mean sampling period averaged across all ten months was 1.001 minutes. The Akima interpolation method was chosen because it contributes relatively little to high frequency noise, has an intuitive result, and is readily available on most platforms [11]. A "data metric" was also generated for the control of clock skew and missing data.

Let t_k be the time that the k'th data sample is acquired, let s_k be the speed measured at time t_k, and let v_k be the volume measured at t_k. Then $d_{t_k} = (t_k, s_k, v_k)$ and the set of all sampled data is given by $X = \{(t_k, s_k, v_k)\}_{k=1...N}$, where N is the number of data samples actually acquired. Interpolation is applied to X to obtain the interpolated data set $\widehat{X} = \{(\widehat{t}_k, \widehat{s}_k, \widehat{v}_k)\}_{k=1...K}$. Note that, since some data will generally be lost, $N \neq K$ in most cases. It is important to note that interpolation is used only for approximating missing data and not as a predictor itself.

The data metric is given by $\delta_k = \min_{1\leq j\leq N} |t_j - \widehat{t}_k|$. We use this metric to generate thresholds to determine which data points should be included in generation of the FMP as well as generation of the training sets for the neural networks. Thresholding the data metric limits the distance (time skew) between the support of the interpolated values $\widehat{X}$ and the temporally nearest datum actually acquired in X. One of the benefits of using the Akima interpolation is that it ensures that if the data metric is zero, then the interpolated data value is equal to the sampled data value with the same sampling time. Since the Akima interpolation scheme uses polynomials of order less than three it is clear that arbitrarily close support implies arbitrarily close value in the interpolant function. However, Akima interpolation is not the only data interpolation scheme with this property.

B. *FMP Implementation*

Two FMPs were developed for testing on the Oklahoma City traffic speed and volume data. The predictor period for the first FMP was one day, while it was seven days for the second FMP. These predictor periods were selected by examining the spectral energy density plots of the regularized data. As expected, the spectral energy was greatest for predictor periods of one day and one week. The one-day FMP also differentiated between weekday and weekend data due to dissimilarity. These facts are apparent in Fig. 3.

A static time interval was used to include the entire data set in the prediction algorithm. The data metric threshold was set at 10 minutes. This threshold value may seem high given the sampling frequency, but there were 154.94 data points averaged per table entry for the 1-Day Weekday predictor, 62.40 for the 1-Day Weekend predictor, and 31.05 data points for the 7-Day predictor. Noise introduced by the interpolation mechanism was reduced by averaging over such a large number of data points (see Fig. 3). As shown in Table I, the mean predictors developed here can be conveniently thought of as a table where two values are associated with a time index.

Determining the data-metric for the mean predictor is easier in general than determining the value for the NNs. Since averaging the data reduces noise, the data threshold tolerance is much higher for this type of predictor than for NNs. Fig. 2 is a plot of speed vs. volume for the two FMPs developed as described above. If the 1-day predictor is thought of as in Table I, then, since the sampling period is 1 minute, there are 1,440 table rows in the weekday predictor and 1,440 rows in the weekend predictors. Each row in both conceptual tables is plotted in Fig. 2(b). The 7-day predictor is similarly plotted, but has only one table with 10,080 rows. There is a noticeable lack in the available predictions of both predictors that are low in speed and volume simultaneously. Since simultaneously low data points are apparent in the real data as shown in Fig. 2(a), some level of data aggregation would be beneficial to a prediction system utilizing these tables.

TABLE I
SAMPLE OF ENTRIES IN 7-DAY PREDICTOR

Index	Speed	Volume
8487	64.95	14.72
8488	64.00	15.53
8489	64.32	16.07

C. *Aggregate Predictor*

Feed Forward NNs with one hidden layer were generated to arbitrate between the two FMP predictions and temporally local traffic flow information. Levenberg-Marquardt training was used to reduce the training time [12]. A log-sigmoid transfer function was used in the hidden layer and a linear transfer function was used in the output layer. Since a hidden layer that is larger than the input and output layers allows for greater generalization and relationship formation, and since there is only one output, $\lceil 1.5 \times (\text{number of inputs})\rceil$ was used to determine the number of nodes in the hidden layer. Separate NNs were developed for the prediction of speed and volume data. For this experiment the prediction intervals were selected such that $t \in \{1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60\}$.

The inputs for the NNs were the two FMP predictions, which we call α_t and β_t for the 7-Day and 1-Day predictors, respectively, as well as historic speed or volume data $\widehat{s}_1, \widehat{s}_2, \ldots, \widehat{s}_{K-1}, \widehat{s}_K$, where the subscript for $\widehat{s}$ is the number of minutes prior to the time of the prediction.

D. *Training Sets*

We use mean square error (MSE) to quantify the predictor performance. If $\widehat{p}_k$ is a prediction for some time k and p_k is the actual value for the time series at time k, then

$$\text{MSE} = \frac{\sum_{k=1}^{N} (\widehat{p}_k - p_k)^2}{N}. \quad (2)$$

Training sets for the neural network were generated from interpolated data as mentioned above. It may be acceptable that $\delta_k < 10$ for a mean predictor. However, such a high threshold would lead to poor performance in generating a data set as the interpolation scheme has introduced some noise. *As a principle, it should be the case that if the data set used to generate the FMP is a large, representative sample, and if its predictions are evaluated against a large representative*

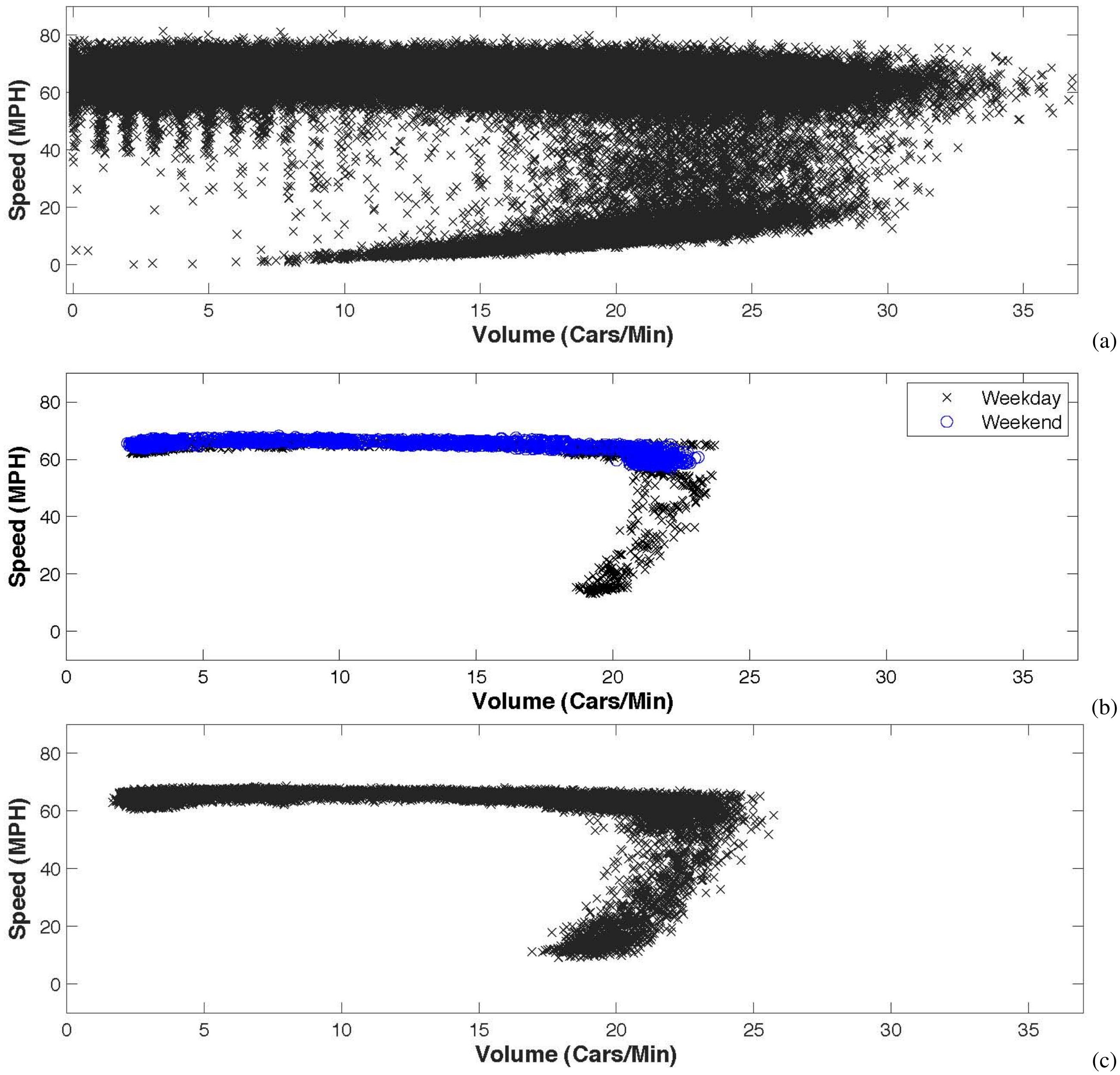


Fig. 2. Speed/Volume Plots: (a) Approx. 104,000 Data Points From RTMS sensors. (b) 1-Day Predictor. (c) 7-Day Predictor.

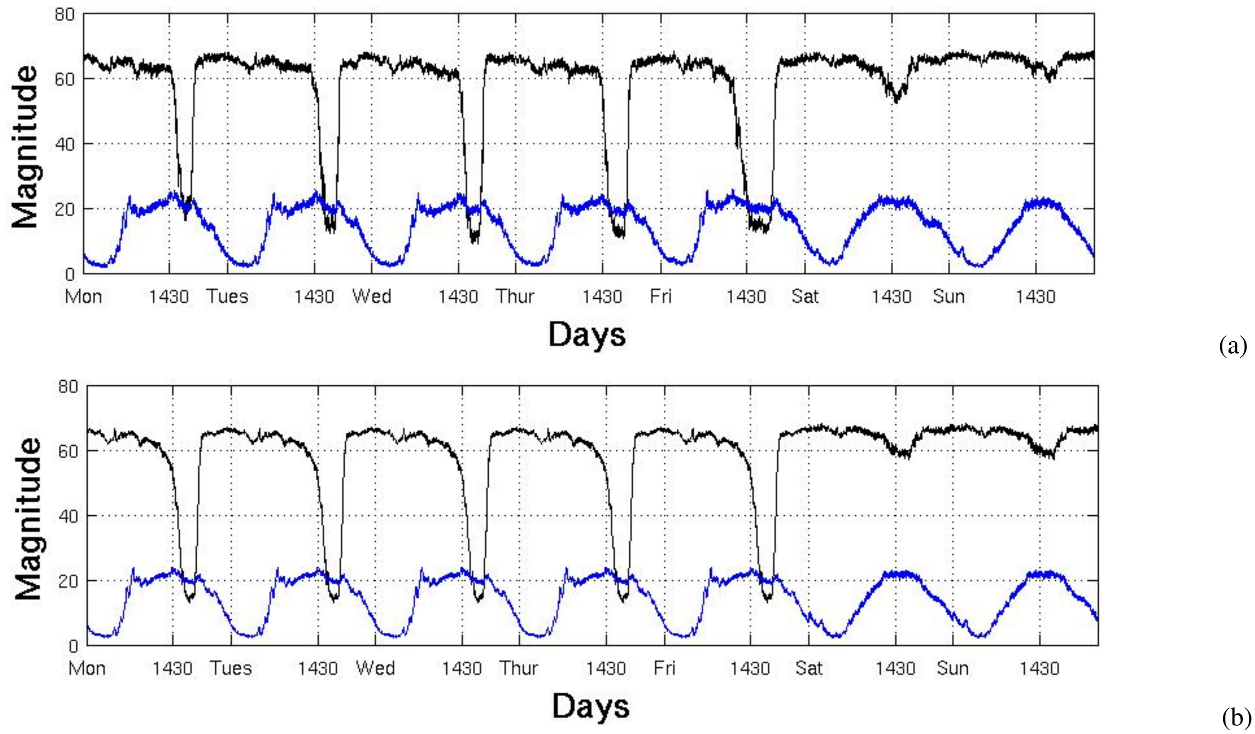
sample regardless of non-stationarities, then the MSE of that sample will not change for a given prediction horizon. That is, an FMP meeting these two conditions should perform equally well predicting any number of sampling intervals into the future. However, if the data threshold applied to the training set is too high, then prediction behavior of the FMP becomes noisy as demonstrated in Table II.

This observation can be used whenever one is uncertain about the nature of the sampling interval or needs to evaluate the training set generated by interpolated data for validity.

Since there is relatively little variance and since greater thresholds yield larger training sets, $\delta < 1.0$ was used to generate the training sets.

TABLE II
MSE FOR GIVEN δ FROM 7-DAY PREDICTOR
SPD. IN MPH^2 VOL. IN (VEHICLES/MIN)^2

	$\delta < 0.5$		$\delta < 1.0$		$\delta < 2.0$	
t	spd.	vol.	spd.	vol.	spd.	vol.
1	313.3	131.7	309.7	131.4	334.4	144.9
5	313.3	131.2	309.9	131.0	324.5	144.4
10	313.4	130.3	310.0	130.4	315.9	143.5
15	313.6	129.7	310.1	129.8	312.9	142.6
20	313.7	129.4	310.2	129.6	311.3	141.3
25	313.8	128.8	310.4	129.3	305.2	140.4



1

Fig. 3. Speed/Volume Plots: (a) 7-Day Predictor. (b)1-Day Weekday and Weekend Predictor Replicated to 7-Days. *Upper plot is Speed in (MPH), Lower Plot is Volume (Cars/Minute). 1430 is the 24-hour time of day giving a visual reference for time.*

E. Training

The data were divided into training (70%), validation (15%), and testing (15%) sets. The number of training epochs was limited to 1,000 and training terminated if further training provided worse performance on the validation set six times consecutively. The NN performance was evaluated by calculating the MSE on the testing set. The MSE convergence goal was set to machine precision. Each training set had 288,411 input/target sets.

F. Results

The MSEs for the NNs evaluated on the testing sets described in Section IV-E are plotted in Fig. 4. The MSE for both speed (in MPH^2) and volume (in (Vehicles/Min.)^2 are shown with the numbers of historical inputs varying from 0 to 10. It is worth noting that the improvement in using FMPs as the only input into the NNs as described in Section II was from on average 310.05 MPH^2 for the 7-day and 282.77 MPH^2 for the 1-day to 11.00 MPH^2 using the NN as an aggregator. For volume, the improvement was from 130.25 (Vehicles/Min)^2 for the 7-day to 11.02 (Vehicles/Min)^2.

In general, for both speed and volume, more inputs yield better predictions and smaller MSE for a given prediction interval, making the benefit of adding recent traffic information apparent. While typically more accurate in their predictions, NNs with greater numbers of inputs are computationally more expensive at prediction time. It should also be considered that some of the benefit may occur due to the increased size of the network rather then the increased information in estimation since the size the hidden layer and thus the NN increases with the size of the inputs. The designer should be able to choose a point where the accuracy required for the application is achieved while the number of inputs is minimized for computational efficiency.

The prediction horizon effect is demonstrated in Fig. 4(b). For numbers of inputs greater than 5, there is an upward slope from the 1 minute prediction interval to the 60 minute interval. It appears that this effect is nascent in the number of speed history inputs greater than 8, but as the speed data shows considerably less prediction horizon effect than the volume data, further research is required to verify this.

V. Conclusion & Future Work

FMPs provide a computationally simple and relatively effective stand alone prediction model for traffic flow. As they are implemented in our case study, a marked improvement through neural network refereed prediction aggregation is demonstrated. We proposed a technique for data regularization that provided the basis for both FMP and neural network training sets that yielded satisfactory results and provided useful information for the design of future aggregate predictor systems.

The design of the FMP as described in the case study is also an extremely effective tool for determining the data skew/data loss cutoff metric. The formulation and development of this tool to verify data set validity is straightforward to generate hence making it appealing.

Comparing the results of traffic flow prediction systems proves difficult in that studies use different temporal reso-

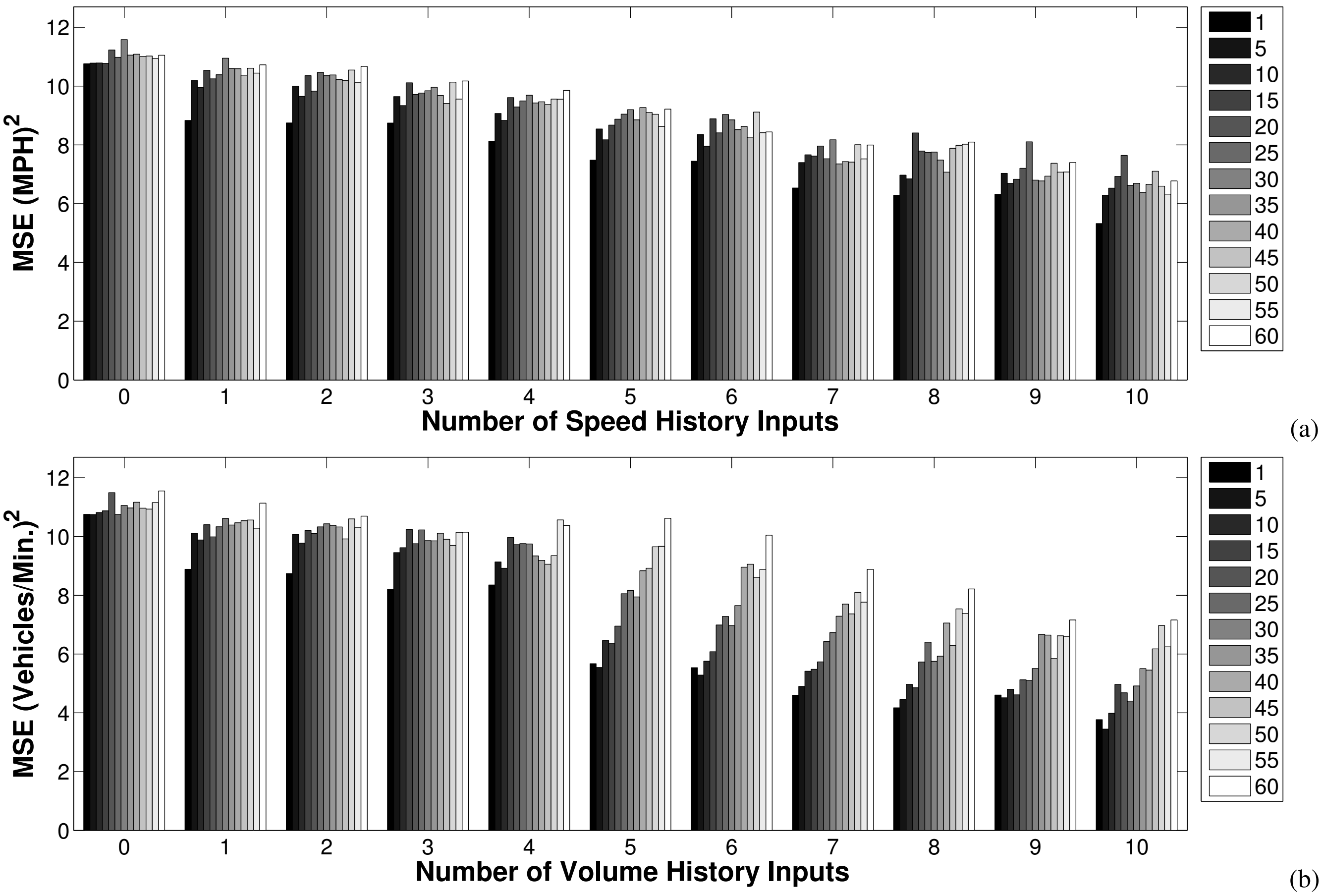


Fig. 4. MSE vs. Number of Inputs: (a) Speed Results (b) Volume Results

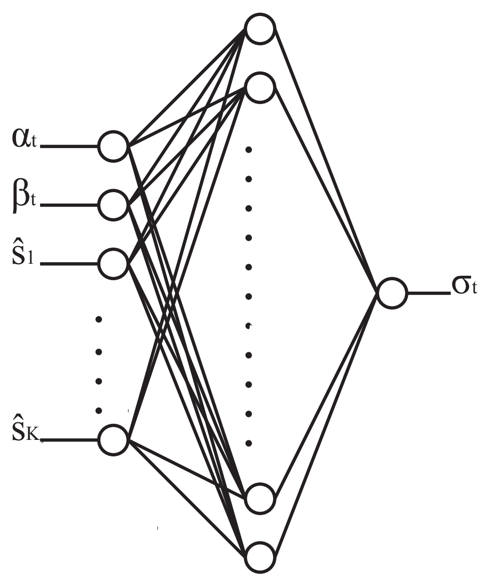


Fig. 5. Diagram of neural network for speed prediction. The neural networks for predicting volume are similar.

lutions, for example 15-minute in [8] or 1-hour as in [2], [9]. These termporal resolutions are dependent on the type of sensor being deployed. The sensing volume for RTMS or laser are relatively small, whereas the sensing volume for inductive loop is large by comparison. As these technologies become available for research new studies will be required to examine their use in ATIS systems.

It seems unlikely that a NN trained for one particular lane will yield effective results for lanes that do not exhibit similar traffic cycles and patterns. Development of a methodology to group lanes that can effectively use the same prediction engine would be necessary to develop a widely distributed prediction system. Most studies up to this point have used relatively few data source locations, e.g. one in [2], [7] and two in [9]. With the proliferation of distributed RTMS networks (Oklahoma has more than 75 sensors installed) and other sensor systems, further research into NN based prediction and extension of the use of a single prediction engine to many sensors or lanes would provide widely applicable information.

It is also likely that extending our study with a greater number of speed inputs would result in a more pronounced speed horizon effect. Further inquiry is required to explain the rate of development for this effect and how it differs between speed and volume data. This extension would also explore the upper limit to prediction system benefit gained by increasing the number of historic speed inputs, which is not apparent in this study.

Interpolation as described in the paper is an effective technique for approximating missing data and support regularization given that the consecutive data loss is limited to some small portion of the speed history. Not discussed in this paper, but potentially promising, is the idea of using the output of the predictive system to replace missing input to the same system. This offers one potential solution to the data loss problem. This speculative prediction is used in branch prediction algorithms in computer architecture [13]. Researching the use of known data replacement and replication techniques in these predictors also seems promising.

The predictor proposed in the case study here used a neural network to aggregate FMP predictions and historic traffic flow data and was shown to be an effective predictive system. The relative ease and configurability of this system makes it a feasible and desirable choice for deployment in real world systems.

REFERENCES

[1] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *Int'l J. of Forecasting*, vol. 5, no. 4, pp. 559–583, 1989.

[2] M. C. Tan, S. C. Wong, M. C. Xu, Z. R. Guan, and P. Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Trans. Intell. Transportation Syst.*, vol. 10, no. 4, pp. 60–69, March 2009.

[3] D. Park, L. R. Rilett, and G. Han, "Spectral basis neural networks for real-time travel time forecasting," *J. Transportation Engr.*, vol. 125, no. 6, pp. 515–523, Nov./Dec. 1999.

[4] Y. Yue, A. G. O. Yeh, and Y. Zhuang, "Prediction time horizon and effectiveness of real-time data on short-term traffic predictability," in *Proc. 2007 IEEE Intell. Transp. Syst. Conf.*, Sept./Oct. 2007.

[5] S. I.-J. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *J. Transportation Engr.*, vol. 129, no. 6, pp. 608–616, Nov./Dec. 2003.

[6] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *J. Transportation Engr.*, vol. 131, no. 10, pp. 771–779, October 2005.

[7] L. Vanajakshi and L. Rilett, "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed," in *Proc. 2004 IEEE Intell. Vehicles Symp.*, June 2004, pp. 194–199.

[8] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transportation Engr.*, vol. 132, no. 2, pp. 112–121, February 2006.

[9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *J. Transportation Engr.*, vol. 129, no. 6, pp. 664–672, Nov/Dec 2003.

[10] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. New York: Prentice Hall, 2008.

[11] H. Akima, "A new method of interpolation and smooth curve fitting based on local procedures," *J. ACM*, vol. 17, no. 4, pp. 589–602, October 1970.

[12] M. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, no. 6, pp. 989–993, Novemeber 1994.

[13] E. Hao, P. yung Chang, and Y. N. Patt, "The effect of speculatively updating branch history on branch prediction accuracy, revisited," in *Proc. 27th Int'l. Symp. Microarchitecture*, 1994, pp. 228–232.