Traffic Prediction for Cognitive Networking in Multi-Channel Wireless Networks

Yao Liu, Bheemarjuna Reddy Tamma, B. S. Manoj, and Ramesh Rao
California Institute for Telecommunications and Information Technology – UC San Diego, USA
E-mail: {yal019, btamma, bsmanoj, rrao}@ucsd.edu

Abstract—One of the early applications of cognitive networking in wireless networks is the problem of dynamic channel selection in multi-channel wireless networks. Dynamic channel selection requires gathering past and current traffic across multiple channels and predict future traffic loads on each of the channels for deciding best channel for the Access Point (AP) to operate on for serving wireless clients. Traffic prediction can be performed by employing Multi-layer Feedforward Neural Network (MFNN) models for learning the effect of spatio-temporal-spectral parameters on traffic patterns and predicting future traffic loads on each of the channels. In this paper, we construct three kinds of traffic predictors that predict traffic at different time scales: MLP (Minute Level Prediction), MILP (Minute Interval Level Prediction), and HLP (Hourly Level Prediction) schemes and study their prediction accuracy in a campus wireless LAN environment. We found that one can simplify the cost of traffic prediction by carefully choosing input parameters of neural network model based on underlying traffic characteristics of environment under study. It is also observed that each location has its own unique traffic pattern, which makes it hard to re-use traffic predictor designed for an environment in a different environment.

I. INTRODUCTION

Cognitive networking involves developing wireless systems that will have deeper awareness about their own operations and the network environment, learn relationships among network parameters, network protocols, and the network environment, plan and make decisions in order to achieve local, end-to-end, and network-wide performance as well as resource management goals. In cognitive network paradigm, all network elements track the spatial, temporal, and spectral dynamics of their own behavior and the behavior associated with the environment, and report that information to a cognitive controller. The information so gathered is used by cognitive controller to learn, plan and act in a way that meets network or application requirements [1]. Cognitive networking differs from cognitive radios or cognitive radio networking [2], [3] in that the latter two apply cognition only to the physical layer to dynamically detect and use spectrum holes, focusing strictly on dynamic spectrum access, whereas, the objective of cognitive networks is to apply cognition to all layers of the network protocol stack for achieving network-wide performance goals.

One of the applications of cognitive networking in wireless networks is the problem of dynamic channel selection in IEEE 802.11 infrastructured networks. To solve this problem, the cognitive controller needs to gather past and current traffic across multiple channels and predict future traffic loads on each of the channels for deciding best channel for the access point (AP) to operate on for serving wireless clients. Sensing channels for traffic information collection is very challenging in multi-channel wireless networks. The 802.11b/g-based wireless networks operate on ISM band and have their transmissions overlap multiple channels. Even though orthogonal channels are typically used for configuring Access Points (APs), in some cases (e.g., non-802.11 sources such as Bluetooth, microwave ovens, and other noise sources render an orthogonal channel useless) other channels are also being used in the configuration of APs. In some scenarios APs belong to different wireless LANs co-exist on the same channel and compete for radio resources in the same geographic region. To gather traffic information in such network environments, the cognitive wireless networking system should have the capability of monitoring all wireless channels in a spatio-temporal fashion. In [4], we suggested packet sampling with single wireless interface which rotates its channel in a round-robin fashion for traffic monitoring in multi-channel wireless networks. We found that Systematic Timer-driven Time-based (STT) sampling strategy with a sampling period of 11 secs and a sampling duration of one sec allows accurate traffic collection across all 11 channels in IEEE 802.11 b/g based WLANs. In this work, we employ STT sampling strategy for collecting sampled traffic traces in a real campus wireless LAN environment.

Traffic prediction in multi-channel wireless networks involves using sampled traffic traces for learning the effect of spatio-temporal-spectral parameters on traffic patterns and predicting future traffic loads on each of the channels. With this predicted information, the decision engine module of cognitive controller could then see which channel is less loaded and reconfigure the AP’s wireless interface channel accordingly. In order to perform traffic prediction, in this work, we focus on designing neural network based traffic prediction frameworks, because they can model the complex relationship between multiple inputs and the output in a way similar to biological neural networks. Using such neural network frameworks, we could use our historical traffic data to predict network traffic on each of the channels in wireless LAN environments.

We design traffic prediction schemes that involve predicting future traffic at different time scales (i.e., Hourly, Five minute wise, and Minute wise). In cognitive networks, there exists many scenarios where the duration of traffic prediction is very important. Consider a situation where the space-time characteristics result in high traffic fluctuations and, therefore, the time scales of the neural network predictor are to be
fine tuned to achieve high accuracy in prediction. Another scenario where our study will be applicable is in fine tuning the traffic prediction scheme based on the applications’ channel switching constraints. For example, performance of certain application layer protocols can be affected by the prediction mechanism employed in a cognitive wireless networks. That is, the prediction mechanism may result in very rapid channel changes that, if used for carrying TCP traffic, can result in high degradation of the end-to-end throughput. While traffic prediction at hourly time scales may be more useful for dynamic channel selection, traffic prediction at minute wise time scales can be helpful in microscopic visualization of network traffic trends by network administrators for detecting any abnormal use of network resources and intrusion detection.

II. RELATED WORK

The authors of [5], dealing with the problem of network traffic prediction, evaluated the effectiveness of traditional prediction techniques such as Auto Regressive Integrated Moving Average (ARIMA) and fractional ARIMA, and compared them with neural networks, concluding that neural networks are more practical due to lower complexity and the ability to model non-linear relationships between inputs and outputs traffic prediction scheme. In [6], the authors compared the performance of linear regression models with that of neural network models for the purpose of building models of the performance in mobile ad hoc wireless networks as a function of external factors such as traffic load and configurable parameters such as the routing protocol being used; again, the authors concluded that neural network models are the best modeling choice for the considered scenario. In [7], we designed three different neural network based hourly traffic prediction schemes by slightly varying inputs of traffic predictors and studied their performance in a campus wireless LAN. In this paper, we design a set of neural network based traffic prediction schemes by slightly varying inputs of traffic predictors and studied their performance in a campus wireless LAN. We used the following parameters for the inputs/outputs of traffic prediction schemes.

III. TRAFFIC PREDICTION SCHEMES

In this work, we employ Multi-layer Feedforward Neural Networks (MFNNs) to construct three kinds of network traffic prediction schemes, namely MLP (Minute Level Prediction), MILP (Minute Interval Level Prediction), and HLP (Hourly Level Prediction) schemes. In MLP (HLP) scheme, we use inputs to predict mean value of future traffic (in Kbps) over next one minute (hour) interval. But in MILP, we aggregate traffic in every five minutes to be a minute interval and predict traffic for next five minute interval. Traffic load over an interval is calculated as the ratio of sum of sizes of packets exchanged over air and number of secs in that time interval. We made use of MATLAB neural network toolbox to implement traffic predictor schemes and study their prediction accuracy on traffic traces collected in a real campus wireless LAN environment. We constructed predictors as two-layer feedforward backpropagation networks with one hidden layer and one output layer. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between inputs and outputs. In our experiments, the training is done by using Levenberg-Marquardt algorithm [8] and a number of epochs of 100.

We used the following parameters for the inputs/outputs of traffic prediction schemes.

- **Channel**: operating channel which ranges from 1 to 11.
- **DayOfWeek**: ranging from 1 (Monday) to 7 (Sunday).
- **HourOfDay**: ranging from 1 to 24.
- **Traffic**(t-i): average traffic observed during time interval \((t - i - 1, t - i)\), \(i \geq 0\), measured in Kbps.
- **Traffic**(t+i): average traffic over next time interval \((t, t + j)\), \(j \geq 1\), measured in Kbps.

We call the first three parameters *environmental parameters* and the last three parameters *traffic parameters*. One of the objectives of our study is to figure out whether these environmental parameters are important for accurate traffic prediction. In order to carry out that study, we used two approaches: one is to check the weights of environmental parameters in neural network structure obtained after training phase, the other is to compare the prediction accuracy with and without environmental parameters. Table I shows weights of edges connecting input parameters with one of nodes in hidden layer of MLP scheme. We can observe that the magnitudes of weights of the environmental parameters are quite small compared to the traffic parameters. Hence, the traffic parameters are much more effective than the environmental parameters for traffic prediction. We also conducted experiments to study the effect of environmental parameters and the results are shown in Table II. We can observe that there is only very small improvement in accuracy of traffic prediction after adding the environmental parameters as the inputs of traffic prediction scheme. From Tables I and II, we conclude that the environmental parameters can hardly help in improving traffic prediction accuracy. For the purpose of reducing time complexity, we suggest using traffic parameters alone as the inputs to the neural network traffic predictors, ignoring environmental parameters.

<table>
<thead>
<tr>
<th>Sample of Input Parameter Weights in MLP Scheme.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Parameter Name</strong></td>
</tr>
<tr>
<td>Channel</td>
</tr>
<tr>
<td>DayOfWeek</td>
</tr>
<tr>
<td>HourOfDay</td>
</tr>
<tr>
<td>Traffic(t-2)</td>
</tr>
<tr>
<td>Traffic(t-1)</td>
</tr>
<tr>
<td>Traffic(t)</td>
</tr>
</tbody>
</table>

In this work, we take only traffic parameters and construct a couple of traffic prediction schemes at each time scale as shown in Table III. In our study, we also use WAS(3,1) scheme which basically estimates future minute traffic as weighted average of past three minutes traffic. Instead of using neural network model, WAS estimates next minute traffic as:

\[
\text{Traffic}(t + 1) = 0.2 \times \text{Traffic}(t - 2) + 0.4 \times \text{Traffic}(t - 1) + ...
\]
TABLE II
MLP SCHEMES WITH/WITHOUT ENVIRONMENTAL PARAMETERS.

<table>
<thead>
<tr>
<th>Network Inputs</th>
<th>MSE</th>
<th>Rel. Error</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic(t-2),...,Traffic(t)</td>
<td>158</td>
<td>0.063</td>
<td>0.93</td>
</tr>
<tr>
<td>Traffic(t-2),...,Traffic(t)+Env. prms</td>
<td>149</td>
<td>0.061</td>
<td>0.93</td>
</tr>
<tr>
<td>Traffic(t-14),...,Traffic(t)</td>
<td>134</td>
<td>0.050</td>
<td>0.94</td>
</tr>
<tr>
<td>Traffic(t-14),...,Traffic(t)+Env. prms</td>
<td>130</td>
<td>0.052</td>
<td>0.94</td>
</tr>
</tbody>
</table>

0.4 × Traffic(t). We consider the prediction of traffic only for the orthogonal channels, 1, 6, and 11, as only those channels contained significant traffic in our campus wireless LAN environment as can be seen from Figure 1 that shows the traffic behavior for working days across channel and hour dimensions.

TABLE III
VARIOUS MLP/MILP/HLP SCHEMES.

<table>
<thead>
<tr>
<th>Scheme Name</th>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP(3,1)</td>
<td>Past 3 Mnts Traffic</td>
<td>Next Mnt Traffic</td>
</tr>
<tr>
<td>MLP(15,1)</td>
<td>Past 15 Mnts Traffic</td>
<td>Next Mnt Traffic</td>
</tr>
<tr>
<td>WAS(3,1)</td>
<td>Past 3 Mnts Traffic</td>
<td>Next Mnt Traffic</td>
</tr>
<tr>
<td>MILP(15,1)</td>
<td>Past 15 Mnts Traffic</td>
<td>Next 5 Mnt Interval Traffic</td>
</tr>
<tr>
<td>MILP(3,1)</td>
<td>Past three 5 Mnt Intervals Traffic</td>
<td>Next 5 Mnt Interval Traffic</td>
</tr>
<tr>
<td>HLP(3,1)</td>
<td>Past 3 Hrs Traffic</td>
<td>Next Hr Traffic</td>
</tr>
<tr>
<td>HLP(5,1)</td>
<td>Past 5 Hrs Traffic</td>
<td>Next Hr Traffic</td>
</tr>
</tbody>
</table>

IV. ARCHITECTURE OF THE NETWORK TESTBED

Figure 2 shows the schematic diagram of the cognitive network testbed infrastructure that we used for our experiments. The main elements of our testbed include: (i) cognitive access points, a centralized cognitive controller module, and a centralized cognitive repository. The cognitive access points contain two important modules: sensing and serving modules. Traffic sensing module implements STT sampling [4] and gathers traffic from multiple channels. The serving module provides communication services to wireless clients that are connected with cognitive access point. The cognitive controller consists of a prediction engine and a decision engine. The prediction engine implements one of traffic prediction schemes and builds neural network structure by using historically available traffic traces for training phase. Based on the prediction made by the prediction engine for future time instant for all channels, the decision engine could then see which channel would be less crowded and reconfigures the serving wireless interface accordingly. The cognitive controller and database repository are consisted in a Dell PowerEdge 1900 server with 7.2 TB of storage. The programmable clients (PCs) play an important role of acting as clients to the cognitive access points.

Since traffic prediction schemes require large amount of historical data for training the neural network, we configured traffic sensor module of the cognitive access points to collect traffic samples for three months (January-March 2009) in the University of California San Diego campus. The cognitive access points, which are located in different buildings, face different traffic patterns. Here we divide the access points into two categories: namely HTV nodes and LTV nodes. HTV (High Traffic Variance) nodes are the access points that see traffic with very high variance. LTV (Low Traffic Variance) nodes are the access points that do not see wide fluctuations in the traffic. In our study, the nodes in UCSD library are HTV nodes, and the nodes at CALIT2 building are LTV nodes.

Fig. 1. Working days: Traffic Load vs Channel vs Time.

Fig. 2. Cognitive Network Testbed.

V. RESULTS AND OBSERVATIONS

The following metrics are used to evaluate prediction accuracy.

1) **MSE** measures error as the average of the squared difference between the predicted and actual target traffic values, with ideal performance yielding 0 MSE.

2) **Regression Coefficient**, also called as R-value, is a measure of how well the variation in the predicted values is explained by the actual traffic values. If this value is equal to 1, then there is perfect correlation between predicted and actual values.

3) **Relative Error (RE)** is a measure of the proportion that the predicted traffic drifts from the actual traffic.

4) **Channel Selection Accuracy (CSA)** is the percentage of right channel selection decision that our predictor
makes. We calculate the percentage in the following manner: the cognitive controller gets chooses the channel with the lowest predicted traffic value among all channels and compares that with the channel with the lowest actual traffic a posteriori. If they are the same, than we say the traffic predictor scheme makes a correct channel selection. We divide the number of correct channel selections by the total number of selections in order to get the CSA ratio.

Figure 3 shows MSE of the neural network predictor, MLP(3,1), starting at a large value and decreasing to a smaller value. In other words, it shows that the network is learning and converging at faster rate. Best validation performance (minimum MSE of 214) is obtained at epoch 10, which demonstrates the predictor’s quick learning capability.

**Problem with MSE as metric for comparing MFNNs:**

The use of MSE as the metric of prediction accuracy has the following problem. Consider two scenarios: in the first scenario, the actual traffic is 100 Kbps and the predicted traffic is 150 Kbps which results in an MSE value of 2500; and in the second scenario, the traffic is 1000 Kbps and the predicted traffic is 1050 Kbps where again the MSE value is 2500. Clearly the prediction is more accurate in the second scenario, however, we are unable to differentiate them with MSE metric. Therefore, instead of MSE, we use RE which is a better metric to measure the prediction accuracy. For these two scenarios, the REs are 0.5 and 0.05, respectively.

After defining the input and output of various prediction schemes from traffic traces present in cognitive repository, we perform cross-validation and report performance results. The cross-validation involves splitting traffic trace randomly into two subsets, 70% for generating MFNN model using MATLAB neural network toolbox and 30% for testing accuracy of MFNN model obtained. We repeat cross-validation process 10 times and report average values of metrics of interest.

### A. Minute Level Prediction Analysis

We consider three predictor architectures, MLP(3,1), MLP(15,1), and WAS(3,1), for studying accuracy of traffic prediction at minute level time scale. Figures 4, 5, 6, and 7 show variation of RE and R-value for traffic traces collected...
at two different locations in two different days. In Figures 4 and 5, we report results from LVT node location where as in Figures 6 and 7 we report results from HVT node location. As shown in figures, for all kinds of datasets, MFNN based predictors perform better than WAS prediction scheme. For LTV node, MLP(15,1) scheme is slightly better than MLP(3,1) scheme. We can claim the two schemes are equally good since for these two schemes, the relative error is below 0.1 and R-values are greater than 0.96. However, when we come to HTV node location, for which the traffic fluctuates much (i.e., the standard deviation of traffic is high), the situation is different. From Figure 6, we can see that MLP(15,1), which uses 15 inputs, performs much better than that using just three inputs, in terms of both RE and R-value metrics. Thus we can see more inputs are indeed helpful for accurately predicting traffic loads in locations with high traffic variance.

**Quantile Discritized Datasets:** In some situations, we may not want to predict absolute values of traffic, instead may be interested in traffic categories into which traffic will fall. For that, we apply quantile discretization on raw traffic datasets and study performance of MLP(3,1), MLP(15,1), and WAS(3,1). In quantile discretization, each bin (class) receives an equal number of traffic values. The data range of each bin varies according to the traffic values it contains. We apply 10% percentile discretization and assign each traffic class a number, from 1 to 10, and regard this number as the traffic value we need to predict. Figures 8 and 9 show RE and R-values for quantized traffic traces at HTV node location for two different days. Here also MLP(15,1) performs better among three schemes. However, from Figures 6 and 8, it is to be noted that REs of MLP schemes with quantized datasets dropped more than 50% for February 11 traffic. This is because quantile discretization packs extreme traffic loads which occur rarely in traffic trace into a very few categories thereby cuts down variation in traffic (compare STDs of February 11 and 14 before and after quantization).

![Figure 8](image-url)  
**Fig. 8.** Relative Prediction Errors of MLP schemes at HTV node location (quantized datasets).

![Figure 10](image-url)  
**Fig. 10.** RE, R-value, and CSA of MILP schemes at HTV node location.

Based on the results we obtained from raw and quantile datasets, we can conclude that when traffic does not fluctuate much, the MLP(3,1) scheme will perform as good as the MLP(15,1) scheme (here MLP(3,1) is a better choice since it has low complexity). However, for high fluctuating traffic scenarios like library location in our study, we can prefer to use 15 inputs to ensure high prediction accuracy.

**B. Minute Interval Level Prediction Analysis**

In this experiment, we study the effect of grouping of datasets on prediction accuracy. For five minute interval level prediction we want to predict the mean traffic of the next five minutes, so as to choose the channel with the lowest mean traffic. We design two kinds of prediction schemes, namely MILP(15,1) and MILP(3,1). The MILP(15,1) scheme uses previous 15 individual minutes’ traffic to predict mean traffic of next five minute interval while MILP(3,1) scheme groups every five minutes to an interval, and use three such intervals’ traffic to predict next one interval’s traffic. We have used traffic traces collected on February 11 for this study. From Figures 10 and 11, we can see that at HTV node location MILP(3,1) scheme performs better than MILP(15,1). This is because it has grouped the traffic values, which essentially cuts down the standard deviation of traffic dataset. The standard deviation of traffic dataset considered for MILP(15,1) is 64 Kbps, while that is 56 Kbps for MILP(3,1)’s traffic dataset. For LTV node location, grouping will not lead to a significant decrease in traffic standard deviation. In this case, more inputs will help in prediction and MILP(15,1) has slightly better performance, albeit with high complexity.

![Figure 9](image-url)  
**Fig. 9.** R-values of MLP schemes at HTV node location (quantized datasets).

**C. Hourly Level Prediction Analysis**

We consider two predictor architectures, HLP(3,1) and HLP(5,1), for studying accuracy of traffic prediction at hourly
level time scale. We have used traffic traces collected over one month period for this study. Figure 12 shows that HLP(5,1) that takes previous 5 hours traffic as its inputs will be more accurate than HLP(3,1) that uses previous 3 hours traffic. However, comparing these performance with MLP and MILP schemes, we can observe that the prediction errors of HLP schemes are higher. This is because hourly traffic values more separated in time and less correlated, which affects their prediction accuracy.

D. Effect of Space on Prediction Accuracy

In this section, we would like to study the effect of spatial dimension on prediction accuracy of MFNN based traffic prediction schemes. For instance, we have cognitive access points X and Y in two different locations, and we want to see whether we can use node X’s MFNN predictor model to predict node Y’s future traffic patterns. In this experiment, we use the LTV node #1’s traffic traces to train MLP(15,1) scheme and then use this predictor model to predict future traffic patterns of nodes located at other locations in same building (CALIT2) and different building (library). The prediction performance results are given in Table IV. The first record in the table corresponds to applying predictor to predict future traffic at the same (LTV node #1) location and hence shows good results. As we can see, using the network structure at a node to predict other nodes’ traffic will lead to bad prediction accuracy.

VI. Conclusions

In this paper, we found that when traffic does not fluctuate much, predictors that take traffic at previous three time instants (MLP(3,1) and HLP(3,1)) will perform as good as the predictors that depend on a lot of traffic from previous time instants (MLP(15,1) and HLP(5,1)). Hence we suggest using predictors with a few inputs for environments with low traffic variance as they have low space and time complexity. However, for high fluctuating traffic scenarios like library location in our study, we suggest using predictors with so many inputs to ensure high prediction accuracy. We also found that grouping of datasets helps a lot in improving prediction accuracy in high fluctuating traffic scenarios. Finally, we observed that each location has its own unique traffic pattern, which makes it hard to reuse traffic predictor designed for an environment in a different environment.

Even though cognitive networking paradigm is meant to make network management an autonomous activity and also improve network throughput and user satisfaction, frequent re-configurations of operating channel may have a negative effect on the performance seen by users connected to the Internet via cognitive APs. We will study the effect of frequency of channel switch algorithm (i.e., Hourly, Five minute wise, or Minute wise?) on the performance of application sessions in terms of throughput, number of retransmissions, and transfer times.

REFERENCES