Urban Traffic Congestion Prediction Based on Routes Information

Dan Pescaru*
* “Politehnica” University of Timisoara, Computers Department, Romania
e-mail: dan.pescaru@cs.upt.ro

Abstract—Traffic congestion represents an important problem in crowded urban areas. It leads to travel delays, increased fuel consumption and higher level of pollution. This paper proposes a technique for congestion prediction in urban traffic. It uses event based routes selection and relies on information collected by a sensor network. Simulation experiments with more than 50 traffic patterns over eight crowded intersections demonstrate promising results.

I. INTRODUCTION

One major direction for intelligent transportation systems (ITS) research in the last decade represents traffic and congestion forecasting [1]. In case of urban traffic, state of the art intelligent algorithms for intersection control rely on such prediction [2][3]. The main objective is to minimize the average waiting time. This should reduce congestions, resulting in less travel time, fuel consumption and pollution.

This paper presents an urban traffic congestion prediction technique based on routes information. A variety of different factors influence urban traffic congestion. They could be classified in two main categories as too much traffic for the available physical capacity and external events. The first category could be solved mainly through physical changes in the traffic network architecture. The second set of factors includes bad weather, crashes, weak intersections control, and special events. When these events occur, their main impact is to minimize the physical traffic capacity. The work presented in here is focused on the last category.

Reliable real-time traffic data from surveillance systems is essential for accurate prediction. A sensor network covering target area intersections collects data. A central point (base station) uses information to predict traffic based on routes probabilities. Probabilities are maintained in a knowledge database and updated periodically. They are also influenced by asynchronous external events like sport games, concerts or other crowded social events. That information is added by a human supervisor or automatically collected from specialized servers.

The central point gathers sensed intersection queue lengths and traffic flows and predicts the total queue length (TQL) in the considered area. It runs a toolkit containing three processing modules.

The first module implements a correlation-based feature subset selection method [4] to determine subsets of routes that are highly correlated with TQL. It evaluates the worth of possible subsets by considering the individual predictive ability of each feature along with the degree of redundancy between them.

The second module integrates collected data over preset intervals. Its goal is to update periodically the probability information for all routes maintained by the central point knowledge base.

The last module represents the main component running on the central point. It has the responsibility for TQL prediction. To achieve this goal it uses a radial basis function network RBFNetwork [5] as the estimator. If several intersection control schemas exist for considered area, central point keeps one network trained for each schema. In the case of adaptive algorithms, the central point will use a traffic simulator to train the network when a new situation is detected. However, the current implementation provides support only for non-adaptive control techniques.

The rest of this paper is organized as follows. Section two presents related work influencing presented solution. Next section describes solution architecture and implementation. Section four describes the environment used for approach validation. Section five presents the testing results. Finally, the last section concludes the work and suggests future improvements based on the testing results.

II. TRAFFIC PREDICTION

Short term forecasting of urban traffic represents a hot topic of research. Several methods and techniques have been proposed during the last years. Researchers have used a large variety of specifications to model traffic characteristics and produce short-term predictions. Models and tools used ranges from Kalman filtering to nonparametric statistical methods, sequential learning and neural networks.

A wide spread techniques derives from pure time-series models like Autoregressive Integrated Moving Average (ARIMA) [6][7], a fine-tuned version of random walk and random trend algorithms. The fact that ARIMA models are suitable only for linear prediction represents a major drawback.

In [8] Olszewski uses Markov chains to obtain the probability distribution of overflow queue. The algorithm estimates the mean queue length and its variance under different conditions such as stationary and non-stationary arrival processes, and variable service rates. Van Zuylen and Viti [9] also used Markov chains to model the dynamics of the probability distributions of the total queue and overflow queue.

In [10] Zhang and Liu investigate performances of least squares support vector machines for accurately predicting non-peak traffic. The method demonstrates good ability of converging rapidly and avoiding local minima.
Promising results were also proven by methods using advanced technologies like Vehicular Ad-hoc NETwork (VANET) [13]. Vehicles equipped with devices capable of short-range wireless connectivity can form a particular mobile ad-hoc network and transmit accurate data. Nagaraj and Kadam [14] investigated several statistical models in attempt of accurate routes predictions using VANET data. Since this technology is not present yet in real life traffic, we tried to avoid relying on it.

Many researchers appealed to theory of support vector regression to model traffic characteristics and predict traffic states [11][12]. In [15] Huisken and van Maarseveen evaluate several short-term congestion prediction methods like Multi-Linear Regression, Time Series Analysis, Fuzzy Logic, MLPs and RBF networks. They concluded that RBF demonstrates the best results in a large variety of cases. Based on this remark, we adopted RBF networks as a support method for the work presented here considering also its fast response time and low memory demands.

III. SYSTEM ARCHITECTURE

The system proposed in this paper incorporates a sensor network, a central point, a knowledge base and diverse software modules for knowledge management or traffic prediction.

The sensor network gathers traffic data from all intersections in the considered urban area. Information is represented by intersections queue lengths in number of cars and traffic flow in number of cars/hour. The architecture of sensor network depends on available infrastructure and on type of sensor used. A large variety of sensing models were proposed to achieve urban sensing systems. Viable alternatives are magnetic sensor networks, inductive loops, radar or video cameras. We consider in this work a network of video sensors included in a wired or wireless system as presented in [16].

As case study we consider eight crowded intersections from center of the city of Timisoara. The chosen sensor network architecture is presented in Fig. 1. A pair of video cameras managed by a sensor node covers all intersections and area entry points. Intersection nodes transmit sensed queue length at every second. Area entry nodes transmit traffic flow computed over a ten second interval. The central node, not showed in this figure, stores all data over one minute in a buffer and runs all modules on each time interval. Before of that it computes the TQL by summing the all intersections mean queues lengths over the one-minute interval.

The first central point module uses a correlation-based feature subset selection method to determine subsets of routes that are highly correlated with TQL. The correlation between the summed components and the outside variable TQL could be computed as [17]:

\[ r_{ce} = \frac{kr_{ij}}{\sqrt{k + k(k-1)r_{ii}^2}}. \] (1)

The result is the set of routes having highest influence on TQL grow. This set is updated periodically for regular periods when no special events occur. It is also associated with every event-based set of routes and stored along each individual set.

Information is stored also in a route database used to compute the routes probabilities. This information could be used in a future implementation for data mining route information when unexpected events occur.

The main module running on central point implements a radial basis function networks. The RBF network is self-adaptive exhibiting self-learning and self-organizing abilities. It is a multilayer neural network that proves high convergence speed and avoids local minimum problem.

In the RBF network a k-means clustering algorithm is used to provide the basis functions and learns a linear regression on top of that. Neurons in the hidden layer represent a set of functions used for the transformation of input vectors to the space of hidden neurons. The purpose is to provide an approximation of a nonlinear continuous function. At the end, symmetric multivariate Gaussians are fit to data from each cluster.

Fig. 2 depicts the architecture of the network used in presented case study. There are seven inputs represented by mean traffic flow at every area entry points. These values form the \( f_{in} \) input vector. It contains positive integer values as number of cars per minute. The output represents the predicted traffic queue length calculated using equation (2)

\[ TQL'(f_{in}) = \sum_{i=1}^{N} w_i \rho(i f_{in} - c_i), \quad i \in [1, 7]. \] (2)
where $N$ represents the number of neurons in the hidden layer, $c_i$ represents the center vector for unit $i$, and $w_i$ is the linear weight of the unit $i$. The norm considered is the Euclidean distance between vectors.

A Gaussian function $\rho$ local to the center vector $c_i$ plays the role of radial basis function

$$\rho(\|\text{flow}_{in} - c_i\|) = \exp\left[-\beta\|\text{flow}_{in} - c_i\|^2\right].$$

The parameter $\beta$ represents an optimization value determined in a manner that optimizes the fit between $\rho$ and the data.

IV. VALIDATION ENVIRONMENT

A. Validation Tools

To validate the approach presented in the previous section we build an environment based on Simulation of Urban MOBility (SUMO) traffic simulator [19]. SUMO is a state of the art open source microscopic road traffic simulator [20]. It was developed by the Institute of Transportation Systems at the German Aerospace Center and has the purpose of handling large road networks.

The simulator was used in two important stages of validation. First the RBF network needs simulation data for training. Then simulation on patterns adapted to special events allows us to test the accuracy of system prediction.

A Java module was implemented to generate the simulation data based on pre-recorded set of routes. Additional traffic flow information was recorded from considered intersection in various traffic conditions along one week.

System modules where implemented as Java packages using Weka 3.6.9 library [21]. Weka is a collection of machine learning algorithms for data mining tasks developed at the University of Waikato, New Zealand.

For correlation module we use the Weka package weka.attributeSelection.CfsSubsetEval, developed by Mark A. Hall [4]. As search method we use BestFirst. This method searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done.

To adapt the SUMO output for Weka handling we implement a translation module capable to generate standard Weka arff datasets from XML simulation output files. Traffic data and system output was recorded on a XML repository.

TQL prediction module was developed based on weka.classifiers.functions.RBFNetwork Weka package. The random seed to pass on the K-min parameter was set to 1. This value is used by the K-means to generate clusters. We do not specify a maximum number of iterations for logistic regression, therefore the algorithm will iterate until reach convergence. The convergence was obtained in less than one second. The minimum standard deviation for the clusters was set to 0.1. The ridge parameter for the logistic regression or linear regression was set to $1.0^8$. The K-means was set to generate six clusters in relation with the number of considered intersection. The prediction window is set to 15 minutes considering constant traffic flow on all entry points.

B. Validation Data

Simulation map was created starting from the physical environment depicted by Fig. 1. Road information was collected using OpenStreetMap service and converted to SUMO net file using netconvert component. Fig. 3 presents the original OpenStreetMap data and Fig. 4 shows the results loaded in SUMO graphical user

TABLE I. ROUTES COVERING SIMULATION AREA

<table>
<thead>
<tr>
<th>ID</th>
<th>Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>routeWW1</td>
<td>in1 n1 n3 n4 n6 n7 n8 n10 n11 in3</td>
</tr>
<tr>
<td>routeWW2</td>
<td>in3 n11 n9 n8 n7 n6 n5 n17 n16 n15 in4</td>
</tr>
<tr>
<td>routeWN1</td>
<td>in1 n1 n3 n4 n5 n17 n16 n15 in4</td>
</tr>
<tr>
<td>routeWN2</td>
<td>in1 n1 n3 n4 n5 n17 n16 n15 in4</td>
</tr>
<tr>
<td>routeWN3</td>
<td>in3 n11 n9 n8 n7 n6 n5 n17 n16 n15 in4</td>
</tr>
<tr>
<td>routeWN4</td>
<td>in3 n11 n9 n8 n7 n6 n5 n17 n18 n19 n24 n23 n22 in5</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>routeSW1</td>
<td>in8 n25 n26 n3 n2 in2</td>
</tr>
<tr>
<td>routeSW2</td>
<td>in8 n25 n26 n3 n4 n6 n7 n8 n10 n11 in3</td>
</tr>
</tbody>
</table>
The set of inner nodes is formed by 27 values as
\[ \text{nodes} = \{n_1, n_2, \ldots, n_{27}\}. \tag{5} \]

The corresponding set of routes covering the area, considering one-way routes and number of lanes, are presented in Table I. All 36 existing routes are organized from entry point to exit point in a clockwise manner starting with West and ending with South.

V. EXPERIMENTAL RESULTS

Experiments were conducted using the environment presented in the previous chapter. We start with generating traffic patterns for training RBF network. We use data on mean traffic flow collected in one week considering three relevant intervals for each entry point. Interval selected was:

- **Heavy traffic** – Peak hour on a weekday between 7:30-8:00 and between 16:30-17:00. These intervals correspond to normal business hours in Romania and represents regular travel between workers residence and place of work.
- **Medium traffic** – Weekday between 11:00-12:00. This interval corresponds to a traffic relaxation on considered area.
- **Low traffic** – Sunday between 7:00-8:00. This interval is characterized by accidental traffic occurrence.

Table II presents the mean traffic flow recorded for each area entry point. This information is used to generate traffic patterns as combination of various traffic flows on each intersection. To achieve good results in network training 50 distinct patterns were generated as random combination of mean traffic flow intervals over the eight considered intersections. Combinations of patterns representing all three types of traffic flow are relevant as we try to predict special situations when usual traffic patterns do not apply.

To generate these patterns we implement a tool using Java language. This tool takes as input a SUMO routing information file and a XML configuration file representing desired traffic flow on each entry point. Traffic is generated on the original routing file and a SUMO configuration file is also produced.

Simulation output for all 50 cases was filtered to retain only vehicles that are waiting on intersections. To include also the vehicles moving in the queues the threshold was set on 5 km/h. In the current implementation we do not consider the cars that are parking on considered area during simulation. This situation is covered by assumption that a comparable number of vehicles leave parking during this interval. Therefore, TQL value is computed from processed file as a sum of all stopped cars. All values are then gathered on a single **arff** file used for network training. The structure of this file is presented in Fig. 5.

The correlation module takes the **arff** file to generate the list of routes that has highest influence to TQL. After applying the best first algorithm with a maximum 5 levels of backtracking we obtain the result shown in Fig. 6.

The **arff** file is used also for RBF network training. On a dual core Intel Core i3-2120 3.3 MHz PC with 4 GB RAM the training phase takes 0.12s. The result of this phase is presented in Fig. 7. It demonstrates the high
correlation coefficient achieved (0.9842) and a reasonable mean absolute error (10.6629).

To evaluate the accuracy of implemented estimator we use a set of eight patterns corresponding to special events. Half of them have assigned heavy traffic from west-to-east, east-to-west, south-to-north and north-to-south routes respectively, and low traffic on the rest of the routes. They form the heavy/low group. The others have also heavy traffic assigned for specified routes and medium traffic on the others. They form heavy/med group.

Fig. 8 and Fig. 9 show the result of TQL variation after 15 min. of simulation and the corresponding TQL prediction. Pattern used was taken from heavy/low group and heavy/medium group respectively.

As observed from Fig. 8, the measured total queue length tends to stabilize after 12-15 minutes when the traffic is below medium intensity, as in case of high/low patterns. For high traffic presented in Fig. 9, total queue lengths curve are far to rich stability in the considered interval. However, prediction algorithm proves remarkable accuracy in both scenarios.

The results over all patterns are presented in Table III. Simulation mean TQL was calculated over 60 seconds before the 15 minutes time horizon, on the [840s .. 900s] interval.

TABLE III.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Simulation Mean TQL</th>
<th>Predicted TQL</th>
<th>Absolute Error</th>
<th>Relative Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHNLELSL</td>
<td>141.12</td>
<td>152.47</td>
<td>11.35</td>
<td>8.04%</td>
</tr>
<tr>
<td>WLNHELSL</td>
<td>135.59</td>
<td>138.92</td>
<td>3.33</td>
<td>2.46%</td>
</tr>
<tr>
<td>WLNLEHSL</td>
<td>160.71</td>
<td>152.64</td>
<td>-8.07</td>
<td>-5.02%</td>
</tr>
<tr>
<td>WLNLELSH</td>
<td>110.22</td>
<td>126.43</td>
<td>16.21</td>
<td>14.07%</td>
</tr>
<tr>
<td>WHNMEMSM</td>
<td>258.95</td>
<td>263.09</td>
<td>4.14</td>
<td>1.60%</td>
</tr>
<tr>
<td>WMNHEMSM</td>
<td>230.34</td>
<td>245.81</td>
<td>15.47</td>
<td>6.72%</td>
</tr>
<tr>
<td>WMNMEHSM</td>
<td>329.87</td>
<td>309.48</td>
<td>-20.39</td>
<td>-6.18%</td>
</tr>
<tr>
<td>WMNMEMSH</td>
<td>332.64</td>
<td>351.07</td>
<td>18.43</td>
<td>5.54%</td>
</tr>
<tr>
<td>Mean</td>
<td>Error</td>
<td>12.17</td>
<td>5.72%</td>
<td></td>
</tr>
</tbody>
</table>

 Considering problem complexity, the prediction algorithm demonstrates very good results in estimating TQL parameter. The mean relative error is less than 6% with a maximum value of less than 12%. These values are comparable with real traffic variation during the 60 seconds interval used for TQL measurement, which has in our experiments a mean of variation of 5.2%.

VI. CONCLUSIONS

This paper describes an urban traffic congestion prediction technique based on routes information. The technique is particularly useful when special events appear like sport games of concerts. Such events induce high traffic on a well-determined set of routes depending on the event location and roads configuration.

Artificial intelligence methods are used for predicting traffic parameters on a 15 minutes time window. Short-term traffic prediction serves as a basis for intelligent intersections control algorithms. The aim is to obtain more efficient use of existing road network system resulting in reduced traffic congestion, delays, pollution, and fuel consumption. Prediction algorithm uses data provided by a sensor network covering all intersections and entry points of the considered area.

Presented technique allows estimation of two important traffic parameters. The first is the total queue length in intersections from considered area. A radial basis function neural network trained with daily traffic data estimates this parameter. The second parameter is the set of roads that has highest influence in traffic congestion. This parameter is determined with a correlation-based feature subset method.

To prove the concept we have used two state of the art tools for traffic simulation and artificial intelligence algorithms, respectively. Simulations were conducted in SUMO, a well-known open source traffic simulator with a remarkable scalability. For estimation tasks we used Weka toolkit along with some additional modules implemented in Java.

Test where performed on a road map of the center of the city of Timisoara. Considered area includes eight
crowded intersections. RBF Network was trained with a set of 50 traffic scenarios based on measured intersections parameters. Eight traffic patterns where used to evaluate the estimation accuracy.

As presented in the experimental results section, the accuracy of the system is very good. It proves a remarkable low relative error in predicting total queue length over the considered area. The error is comparable with real traffic variation during the 60 seconds interval used for total queue length measurement. These results demonstrate the utility of proposed technique and open good perspectives for reliable intelligent intersection control implementation.

REFERENCES