A TRAFFIC PREDICTION METHOD BASED ON COMPLEX EVENT PROCESSING AND ADAPTIVE BAYESIAN NETWORKS

Yongheng Wang, Lan Kuang

College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
wyh@hnu.edu.cn, kuanglan@hnu.edu.cn

ABSTRACT

Recently Complex Event Processing (CEP) is widely used in many areas to support real time event processing. In some applications events should be prevented proactively before they occur. In this paper we propose a high accuracy traffic prediction method based on complex event processing and Adaptive Bayesian networks. A learning algorithm using search-and-score is proposed to learn the Bayesian network structure. Bayesian Model Averaging is used to address the problem of model uncertainty. The experiments on traffic simulation show that this predictive complex event processing method has good accuracy and acceptable performance.
Keywords Bayesian Model Averaging, Bayesian Networks, complex event processing, Traffic prediction.

INTRODUCTION

Complex Event Processing (CEP) is used to collect events from multiple sources and process them into complex events. CEP is widely used in many areas such as Stock Market, network security and Internet of Things.

Most of the event processing method is reactive which means the action is triggered by the state change of the system. A proactive event processing system has the ability to mitigate or eliminate undesired future events, or to identify and take advantage of future opportunities, by applying prediction and automated decision making technologies [1]. Predictive complex event processing is an important part of proactive event processing system.

In this paper, we propose a traffic prediction method based on Predictive Complex Event Processing (PreCEP) and Bayesian networks. Based on a basic Bayesian networks prediction model, we use Bayesian model averaging to improve the accuracy of prediction. A context clustering based data partitioning and model selecting method are also proposed.

I. RELATED WORKS

A. Complex event processing

Complex event processing detects complex events based on a set/sequence of occurrences of single events by continuously monitoring the event stream, and then reacts to those detected situations. Etzion [2] and Luckman [3] defined the basic concept and architecture of complex event processing. Event Processing Network (EPN) is a network of a collection of Event Processing Agents (EPAs), event producers, and event consumers linked by channels.

The key ideas of complex event detection have four steps. (1) Primitive events are extracted from large volume data. (2) Event correlation or event aggregation is detected to create business event with event operators according to specific rules. (3) Primitive or composite events are processed to extract their time, causal, hierarchical and other semantic relationships. (4) Response is sent to the actionable business information because of guaranteed delivery of events to the subscribers.

CEP engine needs to process streams of events with time stamp and, therefore, numerous event pattern recognition methods are based on sequential variants of probabilistic graphical models, such as Hidden Markov Models, Dynamic Bayesian Networks and Conditional Random Fields . Recently some work on detecting complex events in probabilistic event stream based on Nondeterministic Finite Automaton (NFA) is proposed. Xu et al. proposed a data structure called Chain Instance Queues (CIQ) to detect complex events satisfying query requirements with single scanning probabilistic stream [12]. Conditional Probability Indexing-Tree (CPI-Tree) is defined to store conditional probabilities of Bayesian network to improve the performance. In the work of Kawashima et al., an optimized method is proposed to not only calculate the probability of outputs of compound events but also obtain the value of confidence of the complex pattern given by user against uncertain raw input data stream generated by distrustful network devices [13].

B. Predictive analytics

Predictive analytics (PA) is the technology which deals with the analysis of historical data to give predictions about future events. Such a prediction process can be divided into four steps: (1) collect and preprocess raw data; (2) transform preprocessed data into a form that can be easily handled by the (selected) machine learning method; (3) create the learning model (training) using the transformed data; (4) report predictions to the user using the previously created learning model. Future events will be predictable by using recent data based on the learning model trained for the previously monitored events.

Bayesian networks are widely used in PA. In the work of Castillo et al., Bayesian network is used to predict both the level of total mean flow and the origin-destinations pair flows [4]. Pascale et al. proposed an adaptive Bayesian network in which the network topology can be changed according to the non-stationary characteristics of traffic [5]. Compared to our work, their methods are not designed for CEP.

In the work of Sun et al., traffic flows among adjacent road links in a transportation network were modeled as a Bayesian network. The joint probability distribution between the cause nodes and the effect node in a constructed Bayesian network was described as a Gaussian Mixture Model (GMM) [14]. Hofleitner et al. used Dynamic Bayesian Network (DBN) and they introduced a model based on hydrodynamic traffic theory to learn the density of vehicles on arterial road segments, illustrating the distribution of delay within a road segment [15].

Recently some research work about predictive complex event processing are proposed [6,7]. But most of the work discussed the conceptual framework without detailed implementation technologies.

II. THE BASIC PREDICTION METHOD

A. The Event Model

Definition 1 (probabilistic primitive event): A primitive event in a stream means an atomic occurrence of interest in time. A probabilistic primitive event is represented as <A, T, Pr> where A is the set of attributes and T is the timestamp that the event occurs. Pr is the concrete probability value used to present the occurrence probability of the event. The probability value represents the possibility that one event is converted accurately from truthful data of nature to digital data used for computing in electronic devices.

Definition 2 (probabilistic complex event): Complex event is a combination of primitive events or complex events by some rule. A probabilistic complex event is represented as <E, R, Ts, Pr> where E represents the elements that compose the complex event, R represents the rule of the combination, Ts represents the time span of the complex event and Pr is the probability value.

Definition 3 (event type): A specification for a set of event objects that have the same semantic intent and the same structure. Every event object is considered to be an instance of an event type. An event type can represent either primitive events deriving from a producer or complex events produced by an event processing agent.

The main complex event patterns in our work include ALL, ANY, COUNT, SEQ, etc. In this paper, the COUNT event can be used to represent the number of objects in a specified area during specified time span. The SEQ event can be used to represent the moving path of an object. Those patterns can be composed to create hierarchical complex patterns.

In many applications we usually need to process distributed event streams. In this paper, we assume the event instances from different event stream are independent. In a single event stream, some of the primitive events in SEQ pattern have Markov property which means the occurrence probability of next event is only depended on the current event in the sequence but has nothing to do with historical events. For example, the location of a vehicle at time i+1 is related to its location at time i but has nothing to do with its location before time i. An event sequence with Markov property is called a Markov chain. Like the work of Kawashima et al. [10], we process condition probability using the

Condition Probability Table (CPT). Beside these Markovian events, there might be some primitive events that are independent of each other. Therefore, we can use definition 3 to calculate the probability of SEQ event.

Definition 4 (probability calculation of SEQ event in single stream): In a SEQ event E = SEQ(e_1, e_2,…e_n), the primitive events are partitioned into two sets S and T. Set T contains the independent primitive events while set S contains one or more Markov chain. The probability of E can be calculated according to equation (1):

$$\Pr(E) = \prod_{e_j \in T} \Pr(e_j) \bullet \prod_{s_i \in S} (\Pr(e_{i1}) \prod_{m=1}^{|s_i|-1} \Pr(e_{m+1} \mid e_m)) \tag{1}$$

In equation (1), s_i represents one of the Markov chain in set S and e_{i1} represents the first event in s_i. The conditional probability $\Pr(e_{m+1}|e_m)$ can be found in the CPT.

We extended the SASE method [16] to support probabilistic CEP by adding probability for each event in the AIS. When searching the path in the AIS, equation (1) can be used to calculate the probability of the path.

B. The Basic Prediction Method

The system architecture is shown in fig.1. The basic CEP engine uses probabilistic EPN which is composed of probabilistic EPA (PEPA). The prediction model is trained with historical data and it is used to predict future events based on the recent output of basic CEP engine. In this paper, we address the prediction problem for moving objects which can be vehicles, pedestrian, etc.

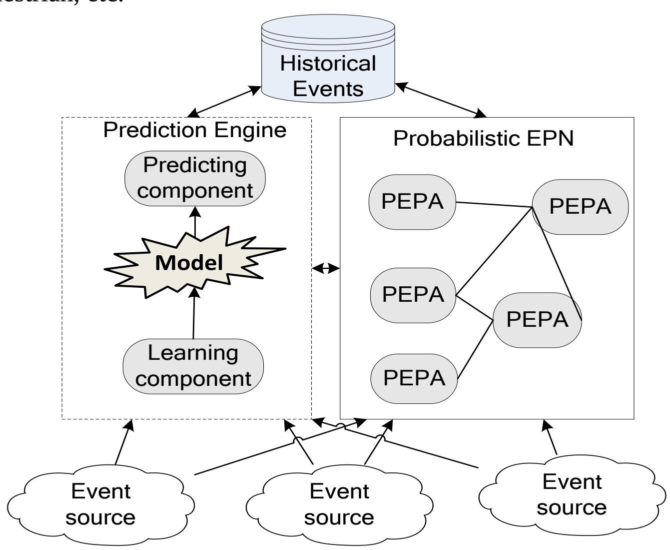

Fig. 1 System architecture

As shown in fig.2, the adaptive Bayesian network model has two dimensions: time and position. The term "adaptive" means the network structure is learned from historical data. Assume there are N positions and objects move among these positions. The state of (i,t) is related to a set of states before time t (parent nodes). Let $s_{i,t}$ represent the state of (i,t) and pa(i,t) represent the parent nodes of (i,t). N_P denotes the number of nodes in pa(i,t). The set of states for pa(i,t) is $S_{pa(i,t)}=\{s_{j,s} : (j,s) \in pa(i,t)\}$. According to the BN theory, the joint distribution of all nodes in the network is:

$$p(S) = \prod_{i,t} p(s_{i,t} \mid S_{pa(i,t)}) \tag{2}$$

The conditional probability $p(s_{i,t}| S_{pa(i,t)})$ can be calculates as:

$$p(s_{i,t} \mid S_{pa(i,t)}) = p(s_{i,t}, S_{pa(i,t)}) / S_{pa(i,t)} \tag{3}$$

Like the work of [5], the joint distribution $p(s_{i,t} , S_{pa(i,t)})$ was modelled with Gaussian Mixture Model:

$$p\left(s_{i,t}, S_{pa(i,t)}\right) = \sum_{m=1}^{M} \alpha_m g_m\left(s_{i,t}, S_{pa(i,t)} \middle| \mu_m, C_m\right) \tag{4}$$

where M is the number of nodes and $g_m(\cdot|\mu_m,C_m)$ is the m-th Gaussian distribution with $(N_P + 1) \times 1$ vector of mean values μ_m and $(N_P + 1) \times (N_P + 1)$ covariance matrix C_m. Using EM algorithm [9] we can infer the parameters $\{\alpha_m, \mu_m, C_m\}_{m=1}^{M}$ from historical data. Then the conditional distribution $p(f_{i,t}| F_{pa(i,t)})$ can be derived from $p(s_{i,t}, S_{pa(i,t)})$ and the estimate $\hat{s}_{i,t}$ can be calculated from $S_{pa(i,t)}$ with minimum mean square error (MMSE) method.

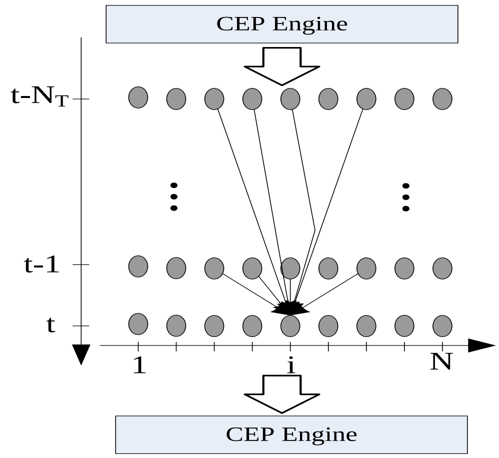

Fig. 2 The Bayesian networks

III. BAYESIAN NETWORK STRUCTURE LEARNING

The Bayesian network structure can be learned from the historical data. Given a node set V and a constraint set C, the Bayesian network structure learning problem is to find an optimized set of edges and an optimized set of distribution parameters.

A Condition Probability Table (CPT) can be created based on the historical paths of the moving objects. Candidates are selected based on global CPT and a search-and-score algorithm is used to learn the structure of Bayesian Network. The global CPT creating algorithm is shown in algorithm 1. In this algorithm, local CPT is first created based on equation (1) for each object plane and then all local CPTs are summed up and normalized to get the global CPT.

Algorithm 1. Generate global CPT
Method:

```
 begin
    for each plane p do
      for each node Np(i,t1)in p do
        for each node Np(j,t2) that t2<t1 do
            find all SEQ event that constructs path <Np(j,t2),
            Np(i,t1)>
            calculate Pr(Np(j,t2), Np(i,t1) based on equation (1)
            save Pr(Np(j,t2), Np(i,t1)) into local CPT
        end
      end
    end
  sum CPTs of all planes into globalCPT and normalize it
  return globalCPT
end
```

We use a structure learning algorithm based on search-and-score which is shown in algorithm 2. The main idea of this algorithm is to maximize the score function. Like the work of [10], we use BIC (Bayesian Information Criterion) score function.

$$SBIC(D,G,\Theta) = \log P(D \mid \hat{\Theta}, G) - \frac{d}{2}\log m + o(1) \tag{5}$$

where D is the data set, $\hat{\Theta}$ is the maximum likelihood distribution parameters for D, d is the number of edges, and m is the sample size per vertex.

This algorithm has two stages: expansion stage and contraction stage. During the expansion stage, edges that can increase the score function are added into the network. During the contraction stage, edges are removed if that does not decrease the score function. The candidate parent vertexes for a specific vertex are selected according to the CPT. α is a threshold value.

Algorithm 2. learn Bayesian network structure using search-and-score
Method:

```
for each vertex V(n,t)
   candidateParents←{V(m,t')|t'<t ∧ Pr(V(m,t'),V(n,t)) > α}
   for each candidate parent p(m,t') in candidateParents
       calculate the score after new edge <p(m,t'),V(n,t)> is added
      Keep the local structure maximizing the score
for each vertex V(n,t)
      for each parent p(m,t')
         calculate the score after edge <p(m,t'),V(n,t)> is removed
         Keep the local structure maximizing the score
```

IV. BAYESIAN MODEL AVERAGING

We found different models fit event data under different event context. To address the problem of uncertainty in model selection, we used Bayesian Model Averaging.

For data D and a set of K models, the model ensemble posterior distribution of a quantity Q (for instance the future model predictions using new input data) can be calculated as:

$$p(Q \mid D) = \sum_{k=1}^{K} p(Q \mid M_k, D)\, p(M_k \mid D) \tag{6}$$

where p(Q,M_k, D) is the posterior distribution of quantity under model M_k and data D. p(M_k|D) is the posterior model probability (model weight). This is a linear combination of the predictions made by each model separately where the weighting coefficients are given by the Posterior Model Probabilities (PMPs). The model weight is obtained by comparing the precision of different models. Assume model M_k has a vector of m_k parameters $\Theta_k=(\theta_{1k}, \theta_{2k},.., \theta_{mk})$ and D=($d_1,d_2,\ldots,d_n$) is a vector of n observation on the m_k parameters. According to Bayesian theory we get

$$p(\Theta_k \mid D, M_k) = \frac{p(D \mid \Theta_k, M_k)\, p(\Theta_k \mid M_k)}{p(D \mid M_k)} \tag{7}$$

where p(D|M_k) is model evidence (also called marginal likelihood function). p(D|M_k) can be calculated by:

$$p(D \mid M_k) = \int p(D \mid \theta_k, M_k)\, p(\theta_k \mid M_k)\, d\theta_k \tag{8}$$

The posterior probability of each model can be calculated by:

$$p(M_k \mid D) = \frac{p(D \mid M_k)\, p(M_k)}{p(D)} \tag{9}$$

where p(M_k) represents the prior distribution of model K. Usually if we have no preference for the models, we assume every model has the same prior probability. The p(D) is not related to models which means the quality of different model is mainly determined by p(D|M_k). Equation (9) can be transformed to the following:

$$p(M_k \mid D) = \frac{p(D \mid M_k)}{\sum_{i=1}^{K} p(D \mid M_i)} \tag{10}$$

Now the key problem is how to calculate the integration in equation (8). According to the work of Gelfand et al. [11], the model evidence p(D|M_k) can be estimated effectively using cross validation distribution as the following:

$$\hat{p}(D \mid M_k) = \prod_{i=1}^{n} p(y_i \mid M_k), (x_i, y_i) \in D \tag{11}$$

where $p(y_i|M_k)$ can be calculate by:

$$p(y_i \mid M_k) = \int p(y_i \mid \theta_k, M_k) p(\theta_k \mid M_k) d\theta_k \tag{12}$$

If the model is relatively complex, equation (12) is still not easy to be calculated directly. We use Markov chain Monte Carlo (MCMC) method to calculate it approximately. Through sampling, we can get a series of independent samples $\theta_k^{(t)}$： t=1,…,T of θ_k from the distribution $p(\theta_k |M_k)$ of θ_k. Then equation (12) can be approximated by:

$$\hat{p}(y_i \mid M_k) = \frac{1}{T} \sum_{t=1}^{T} p(y_i \mid \theta^{(t)}{}_k, M_k) \tag{13}$$

In order to find independent series of samples in MCMC method, we use a Markov chain $\theta_0, \theta_1, \ldots$, in which every value in the chain depends on its previous value only. When some condition is satisfied, whatever the original value of the series, after m iteration the series will converge to static distribution $p(\theta)$. Then the samples $\theta^{(t)}$: t=m+1,…,n generated by the following iterations can be used as independent samples of MCMC.

V. Experimental Evaluations

We developed a traffic simulation system based on SUMO to evaluate predictive complex event processing. The architecture of the system is shown in Fig.3. We can get raw events from the simulation system with the help of the TraCI interface of SUMO. We created a map of 55 junctions and placed 50000 vehicles into the map. In order to simulate real traffic system, a series of rules are defined. Each vehicle has a home location and an office location. A vehicle vi runs between home and office with probability αi. Vehicles go to other places such as supermarket, hospital, etc., with corresponding probabilities. We used 5 servers with 16GB memory as a cluster and the operating system is Ubuntu Server 12. One server is used for the traffic simulating system and others for PreCEP. The tasks of evaluating different models are partitioned into different servers to run parallel.

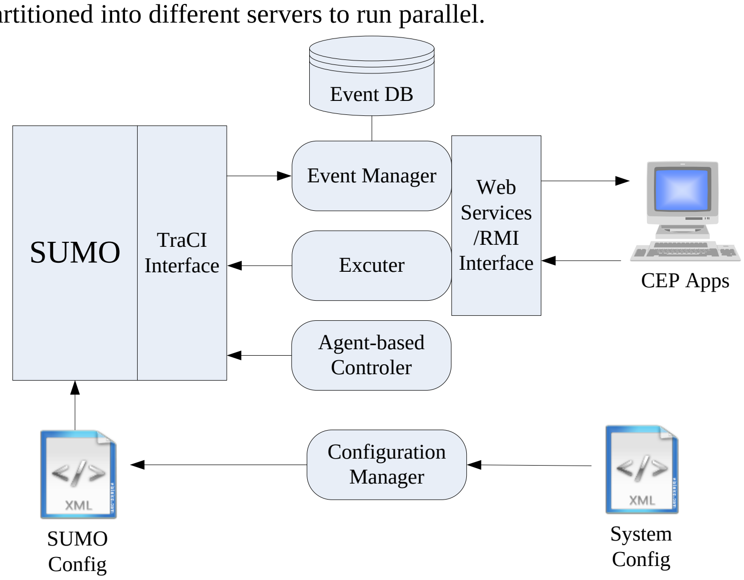

Fig. 3 The simulation system architecture

The accuracy of PreCEP for a typical node is shown in fig.4 and Table I. "SAB" means single adaptive Bayesian model and "BMA" means the Bayesian model averaging method. As we can see, through model averaging BMA gets better accuracy than single model.

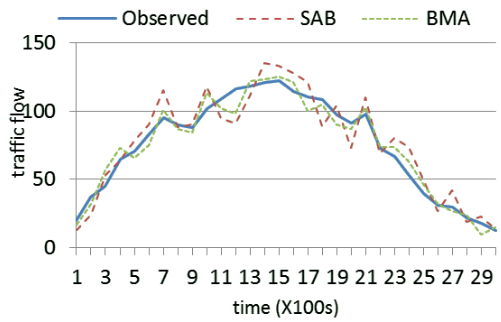

Fig. 4 accuracy for a typical node

TABLE I
DEVIATION FOR MODELS

	deviation	
	SAB	BMA
max	25	18
min	1	1
average	10.3	6.2

The accuracy of the models with the different sample data size is shown in fig. 5. The average deviation decreases when the sample data size increases but the decreasing almost stops when the sample data size reaches a certain value (about 800M in this experiment). The accuracy of BMA decreases more rapidly than that of SAB which means BMA needs more training data to get better accuracy. The performance of the three models with different training data size is shown in fig.6. The running time for both models increases when the training data size increases. But the running time of BMA increases more rapidly because the calculation becomes more complex. That means BMA gets better accuracy than SAB but worse performance. From all the experiments we can see PreCEP with BMA gets good prediction accuracy but it needs more training data and running time.

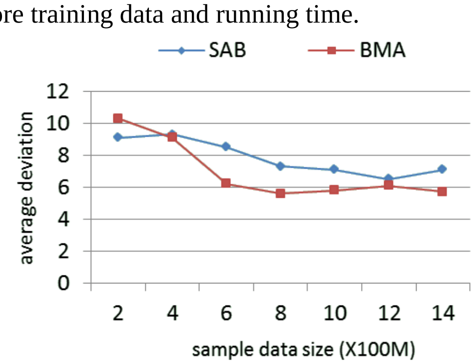

Fig. 5 accuracy for different sample data size

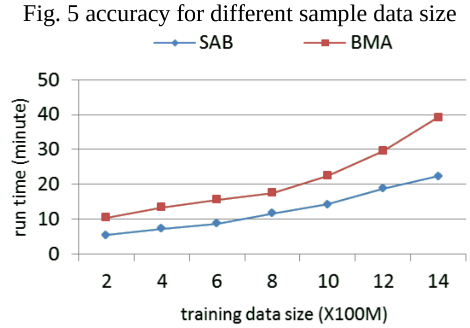

Fig. 6 performance for different training data size

VI. CONCLUSIONS

In this paper we propose a high accuracy predictive complex event processing method using Bayesian networks, model averaging and dynamic model selection. The experimental evaluations show that this method has better accuracy than traditional methods and acceptable performance when working in large-scale IoT applications. The performance of PreCEP still needs to be improved. Currently the parallel method only works when learning the structure of models and training models for different context. In the future we want to develop new algorithms to support massive historical data and complex training process.

ACKNOWLEDGMENT

The research work is supported by National Natural Science Foundation of China under Grant No. 61371116 and Hunan Provincial Natural Science Foundation under Grant No.13JJ3046.

REFERENCES

[1] Y. Engel, O. Etzion, "Towards proactive event-driven computing", Proceedings of the Fifth ACM International Conference on Distributed Event-Based Systems, DEBS 2011, New York, pp.125-136.

[2] Etzion, P. Niblett, "Event Processing in Action", Manning Publications, 2010.

[3] D. C. Luckham, "Event Processing for Business: Organizing the Real-Time Enterprise", Wiley Publications, 2011.

[4] E. Castillo, J. M. Menéndez and S. Sánchez-Cambronero, "Predicting traffic flow using Bayesian networks", Transportation Research Part B: Methodological. 42(2008), pp.482-509.

[5] Pascale, M. Nicoli, "Adaptive Bayesian network for traffic flow prediction", Proceedings of the Statistical Signal Processing Workshop (SSP), 2011 IEEE, pp.177-180.

[6] J. Lang and Z. Capík, "Prediction based on hybrid method in complex event processing", proceedings of the IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI2014), Jan. 2014, Herl'any, Slovakia.

[7] L. J. Fülöp, Á. Beszédes, G. Tóth, et al., "Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics", Proceedings of the Fifth Balkan Conference in Informatics, Sep. 2012, Novi Sad, Serbia.

[8] J.J. Verbeek, N. Vlassis nad B. Krose, "Efficient Greedy Learning of Gaussian Mixture Models", Neural Comput., vol.15, issue 2, February 2003, pp. 469-485.

[9] J.J. Verbeek, N. Vlassis nad B. Krose, "Efficient Greedy Learning of Gaussian Mixture Models", Neural Comput., vol.15, issue 2, February 2003, pp. 469-485.

[10] S. Samaranayake, S. Blandin, A. M. Bayen, "Learning the dependency structure of highway networks for traffic forecast", in Proc. 50th IEEE Conference on Decision and Control and European Control Conference , 2011, pp.5983-5988.

[11] A. Gelfand, D. Dey, 1994. "Bayesian model choice: asymptotics and exact calculations", J. Roy. Statist. Soc. Ser. B 56, 501–514.

[12] C. Xu, S. Lin, W. Lei, "Complex Event Detection in Probabilistic Stream", Proceedings of the 12th International Asia-Pacific Web Conference (APWeb 2010), pp.361-363.

[13] H. Kawashima, H. Kitagawa and X. Li, "Complex Event Processing over Uncertain Data Streams", Proceedings of the fifth international conference on P2P, parallel, grid, cloud and internet computing (2010), pp. 521-526.

[14] S. Sun, C. Zhang and G. Yu, A Bayesian Network Approach to Traffic Flow Forecasting, Intelligent Transportation Systems, 7(2006), pp.124-132.

[15] A. Hofleitner, R. Herring and P. Abbeel, "Learning the Dynamics of Arterial Traffic From Probe Data Using a Dynamic Bayesian Network", ITS(13), No. 4, December 2012, pp.1679-1693.

[16] J. Agrawal, Y. Diao and D. Gyllstrom, "Efficient pattern matching over event streams", in *Proc. SIGMOD Conference (2008)*, pp.147-160.