A Proactive Complex Event Processing Method for Intelligent Transportation Systems

Yongheng Wang
College of Information Science and Engineering, Hunan University, Changsha, China
yh.wang.cn@gmail.com

***Abstract*—Recently the rapid development of Internet of Thing (IoT) brings new opportunity to Intelligent Transportation System (ITS). The key problem is how to process the huge events generated by IoT to support ITS. In this paper a proactive complex event processing method is proposed for congestion control in ITS. Based on complex event processing, this method predicts the traffic flow using a multi-layered Adaptive Dynamic Bayesian Network. When congestion is predicted, actions are chosen and executed based on concurrent-actions Markov Decision Processes (MDP) to avoid the future traffic congestion. The simulation experiments show that this method can reduce the traffic congestion obviously for typical transportation system.**

***Index Terms*—Internet of things; intelligent transportation Systems; Proactive Complex Event Processing; Predictive Analytics**

I. Introduction

Internet of Things (IoT) is an integrated part of future internet and could be defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network. The key issue in IoT application is how to process the huge events produced by IoT applications. Intelligent transport systems (ITS) are advanced applications aim to provide innovative services relating to different modes of transport and traffic management and enable various users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks. ITS is one of the application that can take advantage of IoT.

In IoT applications, event processing applications must process events that arrive from various kinds of sources such as sensors, RFID readers, GPS, etc. The events that generated by devices directly are primitive events. The semantic information inside primitive event is quite limited. In real application, we pay more attention to complex information such as business logic and rule. For example, each reading operation of the RFID reader at a garage generates a primitive event but complex event like "the car leaves the garage" is the kind of event that user really concern. IoT application systems convert business logic into complex events and then detect business logic based on detecting complex events. Complex Event Processing (CEP) [1] is used to process huge primitive events and get valuable information from them.

Complex event processing has been studied widely in active database. Most of the methods assume data is deterministic. However, in many real-time IOT applications, events are imprecise because of some factors such as the limitation of measuring accuracy, signal disturbance or privacy protection. The uncertainty is usually treated with probabilities. Therefore it is important to develop event processing engine that be able to deal with probability.

Most of the event processing method in IoT 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 [2]. For example, in ITS we can predict some congestion states and take some actions to avoid the congestion states. Predictive Analytics (PA) deals with the analysis of historical data to give predictions about a future event. PA applies several statistical and data mining techniques such as clustering, classification, regression and so on. CEP and PA is studied widely but there few papers about how to integrate CEP and PA to support proactive event-driven system.

In this paper, we propose a proactive complex event processing method for intelligent transportation systems. Based on probabilistic complex event processing, this method uses concurrent-actions Markov Decision Processes (MDP) to integrate CEP and PA. We also designed a multi-layered Adaptive Dynamic Bayesian Network (mADBN) model for predictive analytics.

II. Related Work

A. CEP and Probabilistic CEP

Complex event processing recognizes complex events based on a set/sequence of occurrences of single events by continuously monitoring the message flow, and then reacts to those detected situations. In their book, Etzion et

Manuscript received January 20, 2013; revised March 27, 2013.
This work was supported by the "complex event processing in large scale internet of things (K120326-11)" project of Changsha technological plan.

©2013 Engineering and Technology Publishing

doi: 10.12720/lnit.1.3.109-113

al. defined the basic concept and architecture of complex event processing [3]. Event Processing Agent (EPA) is a component that, given a set of input events, applies some logic to generate a set of complex events as output. Event Processing Network (EPN) is a network of a collection of EPAs, event producers, and event consumers linked by channels. The network is used to describe the event processing flow execution. The conceptual model of EPN based on this idea was further elaborated by Sharon and Etzion [4].

Most of the CEP methods in active databases and RFID applications use fixed data structure such as tree, directed graph and Petri-Net. SASE [5] is a high performance complex event detection method which uses Nondeterministic Finite Automaton (NFA) and Active Instance Stacks (AIS). Jagrati et al. proposed an improved NFA model to support more powerful query ability [6]. In Haopeng's work, SASE model is improved to support imprecise timestamps when processing complex events in stream [7].

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. Markov Logic Networks (MLNs) [8] have also been used to handle uncertainty in event pattern recognition.

Recently some work on detecting complex events in probabilistic event stream based on NFA is proposed. In the work of [9], a data structure called Chain Instance Queues (CIQ) is used to detect complex events satisfying query requirements with single scanning probabilistic stream. In another paper [10], 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.

B. PA and Proactive Event Processing

In the case of Predictive Analytics Methods based on Complex Event Data, one can make predictions about some attributes of the monitored system based on the previously monitored events. Such a prediction process can be divided into four steps [11]: (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.

Recently, some researchers used Bayesian Network (BN) in predictive analytics for traffic flow. In the work of Pascale et al., an adaptive Bayesian network was proposed in which the network topology changes following the non-stationary characteristics of traffic [12]. 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) [13]. 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 [14].

Proactive applications have been developed for several years. Some examples include proactive security systems, proactive routing in mobile ad-hoc wireless networks, and proactive SLA negotiation in service oriented systems. Engel et al. proposed a proactive event-driven computing framework based on CEP, PA and Markov Decision Processes (MDP) [2], [15]. They extended the event processing agent model to include two more type of agents: predictive agents that may derive future uncertain events based on prediction models, and proactive agents that compute the best proactive action that should be taken.

III. Background

A. System Architecture

The system architecture is shown in Fig. 1. Raw events are generated by devices such as RFID reader, radar, GPS, etc. we assume the events are imprecise because of the limitation of measuring accuracy and signal disturbance. The probabilistic raw event is processed by the Probabilistic Event Processing Network (PEPN) which is composed of many connected Probabilistic Event Processing Agents (PEPA). With PEPN, some complex events, such as the running path of a car, can be detected. The PA component with mADBN predicts the congestion states of the system from the complex events. Some complex events are saved into event database and the historical events can be used by the PA components for more accurate predict. The decision maker selects some actions according to the predicted states and assigns some proactive agents (PRA) to execute the actions.

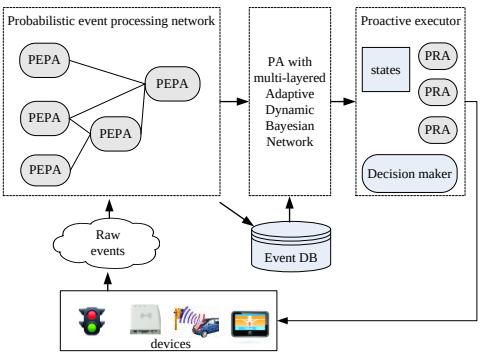

Figure 1. System architecture.

B. Event Model

Definition 1 (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>

©2013 Engineering and Technology Publishing

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 (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.

Event type is 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 vehicles in a specified area during specified time span. The SEQ event can be used to represent the running path of a vehicle. The detailed meaning of the patterns can be found in [3]. Those patterns can be composed to create hierarchical complex patterns.

C. Probabilistic Complex Event Processing

In a large ITS application, we may need to process distributed event streams. In this paper, we assume the event instances from different event stream are independent. For SEQ events in a single stream, some of the primitive events in the sequence have Markov property which means the probability of next event is only related to the current event in the sequence but has nothing to do with previous events. For example, the location probability of a car at time *i*+1 depends on the location at time i and the information detected by some reader at time *i*+1. Like the work of [10], we use Condition Probability Table (CPT) to save and sort the condition probability. The probability of the sequence consists of these Markovian events can be calculated by Bayesian formula. Beside these Markovian events, we assume there might be some primitive events that are independent with others. Therefore, we can calculate the probability of a SEQ event based on definition 3.

Definition 3 (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 can be computed as follows:

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

where the s_i means the Markov chain in set S and e_{i1} means the first event in Markov chain s_i. The conditional probability $\Pr(e_{m+1}|e_m)$ can be found in the condition probability table.

We extended the SASE [6] to support probabilistic CEP by adding probability for each event in the AIS. When searching the path in the AIS, use equation (1) to calculate the probability of the path. Detailed method and algorithm can be found in papers of the same author.

IV. PROACTIVE CEP

A. Predictive Analytics Method

The mADBN model is shown in Fig. 2. It contains a state plane and a set of location planes. Each plane is an adaptive dynamic Bayesian network with two dimensions: time and space. Bayesian networks are directed acyclic graphs whose nodes represent random variables and edges the conditional dependences among them. In the state plane, nodes denote the traffic flow observed in different time instants and/or spatial locations (road junctions), edges their probabilistic relations. The term "dynamic" in mADBN means we are modeling a dynamic system. As we can see in Fig. 2, the state of (i,t) is related to a set of states before time t. The term "adaptive" means the graph structure is created based on analysis of historical data. Each location plane represents the change of a vehicle's location (running path). The arrow means the transaction probability from one place (road junction) to another and the arrow with solid line means the real path.

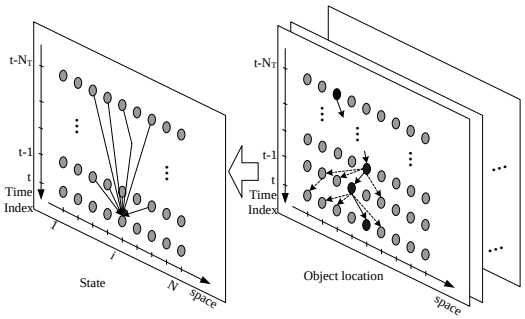

Figure 2. mADBN model.

The structure of the state plane can be created based on analysis of the object location planes. For example, in the state plane of Fig. 2, we need to select the nodes before time t which can affect the state of node (t,i) which is shown in algorithm 1. In this algorithm, the function Dest(pl) gets all the destination nodes from all the paths in an object location plane pl. The IF array is an influence factor array which stores the influence factor of other nodes to node (i,t). The function count(n) returns the number of paths that pass node n. α is a discount factor which means the closer nodes in the path have higher affect factor values. Parallelization of this algorithm is easy since we can partition the object location planes into groups then calculate the IF array in parallel and merge the result at the end. θ is a threshold value.

Algorithm 1. Create BN structure for node in state plane
Input: U is the object location plane, i is the destination node
Output: N_i is the set of nodes that affect state of node i

Method:
$pl_i \leftarrow \{pl | pl \in U \wedge i \in Dest(pl)\}$
foreach $p \in pl_i$
  for k=1 to |p|
    IF[p(k)]+=$\alpha^{|p-k|}$/count(p(k))
  end for
end for
$N_i \leftarrow \{n | n \in N \wedge IF(n) > \theta\}$
return N_i

Let $f_{i,t}$ represent the traffic flow state of (i,t) and pa(i,t) represent the parent nodes of (i,t) which is the output of algorithm 1. N_P denotes the number of nodes in pa(i,t). The set of flow states for pa(i,t) is $F_{pa(i,t)}=\{f_{j,s} : (j,s) \in pa(i,t)\}$. According to the BN theory, the joint distribution of all nodes in the flow states network can be expressed as:

$$p(F) = \prod_{i,t} p(f_{i,t} | F_{pa(i,t)}) \quad (2)$$

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

$$p(f_{i,t} | F_{pa(i,t)}) = p(f_{i,t}, F_{pa(i,t)}) / F_{pa(i,t)} \quad (3)$$

Like [12], [13], we model the joint distribution $p(f_{i,t}, F_{pa(i,t)})$ with GMM as:

$$p(f_{i,t}, F_{pa(i,t)}) = \sum_{m=1}^{M} \alpha_m g_m(f_{i,t}, F_{pa(i,t)} | \mu_m, C_m) \quad (4)$$

where M is the number of nodes and $g_m(|\mu_m, C_m)$ 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. Parameters $\{\alpha_m, \mu_m, C_m\}_{m=1}^{M}$ can be inferred from the historical data using the EM algorithm. Once $p(f_{i,t}, F_{pa(i,t)})$ has been obtained, the conditional distribution $p(f_{i,t}| F_{pa(i,t)})$ can be derived and the estimate $\hat{f}_{i,t}$ can be calculated from $F_{pa(i,t)}$ using the minimum mean square error (MMSE) method.

B. Decision Making with MDP

Definition 4 (Markov Decision Process): A Markov Decision Process (MDP) is a tuple $\langle S, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. S is a set of states and $\mathcal{A}$ is a set of actions that may be executed at any state. $\mathcal{R}: S \rightarrow \mathcal{R}$ is a reward function and $\mathcal{P}: S \times \mathcal{A} \times S \rightarrow [0, 1]$ is a transition function with $\mathcal{P}(s, \alpha, s')$ capturing the probability of reaching s' by applying action $\alpha \in \mathcal{A}$ in state s. For each $(s, \alpha) \in S \times \mathcal{A}$, $\sum_{s' \in S} P(s, \alpha, s') = 1$.

The optimization purpose is to find a policy $\pi: S \rightarrow \mathcal{A}$ which is defined as a function that maps every state $s \in S$ to an action. The quality of a policy is the expected sum of future rewards. Future rewards are discounted by a discount factor γ to ensure that the expected sum of rewards converges to a finite value. The discount factor $\gamma < 1$ indicates that future time steps worth less than the current time step. The quality of a policy can be calculated recursively using the Bellman Equation:

$$v(s) = R(s) + \gamma \max_{\alpha \in A} \sum_{s' \in S} P(s, \alpha, s') v(s') \quad (5)$$

This equation means the value of being at state s is the reward at the current state, plus the (discounted) expectation over the value of the next state. Common solution scheme for MDPs is called value iteration, in which the solution is calculated by iteratively going through all the states and updating their values according to the current values of their neighboring states.

Assume there are N nodes (road junctions) in the ITS network: $J=\{j_1, j_2, \ldots, j_N\}$. We partition the traffic flow value in mADBN model into K levels and represent it as congestion levels $CS=\{c_1, c_2, \ldots, c_K\}$. Then the state in MDP can be represented as $s_t = <s_{t1}, s_{t2}, \ldots, s_{tN}>$ where $s_{ti} \in S$ is the congestion state of node j_i at time t and $s_i \in$ CS.

In order to create the policy set, we partition the traffic network into M subgroups according to congestion nodes which is shown in Fig. 3. One or more nodes which have high congestion level (estimated by PA component) are selected as the center of the group. The nodes have distance smaller than a threshold value D from the center are partitioned into the group. A policy $\pi_{it}=\{a_{it1}, a_{it2}, \ldots, a_{itM}\}$ where a_{itk} is an action aim to reduce the congestion of group k at time t by guiding some vehicles to change their paths. In this paper we assume the distance between any two group centers is large enough which means the sub-actions are independent and they can be executed parallel.

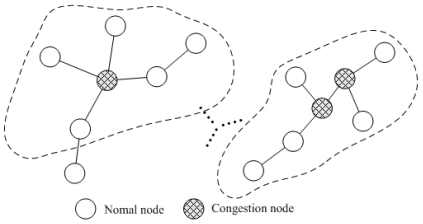

Figure 3. Network partition.

When a sub-action is executed, the traffic flow change of node i can be calculated as following:

$$\Delta_i = S_i - \sum_{jk \in Pa\_out(j_i)} S_k \cdot \alpha_{ki} \cdot \beta_{ki} \cdot p(j_k) - \sum_{jm \in Pa\_in(j_i)} S_m \cdot \alpha_{im} \cdot \beta_{im} \cdot p(j_m) \quad (6)$$

where $Pa\_out(j_i)$ is the node set that try to reduce the flow of node i and $Pa\_in(j_i)$ is the node set that try to increase the flow of node i. α_{ki} is the proportion of S_k that flow to node i. β_{ki} is the proportion of S_k that affected by action a_i (notified by a_i to change path). $P(j_k)$ denotes the probability that vehicles in node k change their path according to action a_i. $P(j_k)$ can be learned from historical data. Since we assume sub-actions are independent, the total state transaction probability can be calculated based on P(j) and Δ_i equation (6).

We can define the reward functions for each sub-action independently and sum them up to get the reward of an action. We first define the mean flow and variance of a group g_i in equation (7) and (8).

$$\bar{g}_i = \sum_{j=1}^{|g_i|} s_{ij} \Big/ |g_i| \quad (7)$$

$$D(g_i) = \left[\sum_{j=1}^{|g_i|} (s_{ij} - \bar{g}_i)^2 \Big/ |g_i| \right]^{1/2} \quad (8)$$

©2013 Engineering and Technology Publishing

The goal of our method is to minimize the total congestion status. Therefore, we define the reward function of sub-action with Euclid distance and the total reward function is defined as follows:

$$\mathcal{R}(\mathrm{S,a,S'})=\sum_{i=1}^{|g|}\left[(\bar{g}_i-\bar{g}'_i)^2+(D(g_i)-D(g'_i))^2\right]^{1/2} \quad (9)$$

Finally, the optimal policy can be calculated by:

$$\pi^*(\mathrm{S_t})=\arg\max_a \sum_{S'} P(s,a,s')V(s') \quad (10)$$

V. Fundamental Evaluations

In this section, we illustrate the accuracy and performance of the proposed method by simulation experiments. The simulation is created based on SUMO project (http://sumo.sourceforge.net/).

In order to simplify the simulation experiment scenarios, we consider a traffic network with 200 junctions and 20000 vehicles. Each junction is connected to 2~4 edges. All vehicles have default path but they may change their path at some junctions with given probability. Eight junctions are designed as congestion nodes. Virtual RFID reader and GPS are used to get the position of the vehicles. A vehicle counter is placed at every junction to get the flow of vehicle. We first run the simulation for many times to get the historical data of the vehicle paths. In the first experiment we evaluated the accuracy of the PA component and the result for a typical node is shown in Fig. 4. From the figure we can see the accuracy of the PA component is good enough for congestion control.

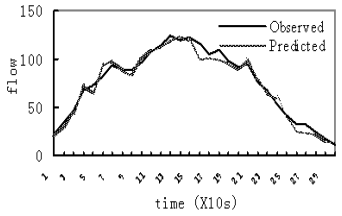

Figure 4. PA accuracy for a typical node.

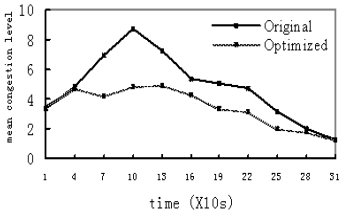

Figure 5. Mean congestion level over time.

In the next experiment, we evaluated the mean congestion level over time and the result is shown in Fig. 5. From the figure we can see the congestion level reduced when our method is used. The reason is that our method can predict the congestion and take some actions to avoid it. The reduction of congestion level is more obvious when the congestion level becomes high. The reason is that the system can redirect more vehicles to light loaded nodes in such circumstance.

VI. Discussion and Conclusion

In this paper, we proposed a proactive complex event processing method for ITS. MDP is combined with DBN-based PA to support proactive event processing. The fundamental evaluations show that this method works well for typical congestion problem.

The main limitation in our work is performance and scalability. When the number of vehicles and junctions becomes larger, the performance of algorithm reduces quickly. In the future we will work on high performance parallel algorithms under this framework.

References

[1] D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison Wesley, Boston, 2002.
[2] Y. Engel and O. Etzion, "Towards proactive event-driven computing," in *Proc. Fifth ACM International Conference on Distributed Event-Based Systems*, New York, 2011, pp.125-136.
[3] O. Etzion and P. Niblett, *Event Processing in Action*, Manning Publications, 2010.
[4] G. Sharon and O. Etzion, "Event-processing network model and implementation," *IBM System Journal*, vol. 47, no. 2, pp.321-344, 2008.
[5] E. Wu, Y. Diao, and S. Rizvi, "High-performance complex event processing over streams", in *Proc. ACM SIGMOD International Conference on Management of Data*, Chicago, IL, USA, June 27-29, 2006.
[6] J. Agrawal, Y. Diao, and D. Gyllstrom, "Efficient pattern matching over event streams," in *Proc. SIGMOD Conference*, 2008, pp.147-160.
[7] H. Zhang, Y. Diao, and N. Immerman, "Recognizing patterns in streams with imprecise timestamps", *PVLDB* vol. 3, no. 1, pp. 244-255, 2010.
[8] M. Richardson, P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp.107-136, 2006.
[9] C. Xu, S. Lin, and W. Lei, "Complex event detection in probabilistic stream," in *Proc. 12th International Asia-Pacific Web Conference*,2010, pp.361-363.
[10] H. Kawashima, H. Kitagawa, and X. Li, "Complex event processing over uncertain data streams," in *Proc. fifth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2010, pp 521-526.
[11] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using Bayesian networks," *Transportation Research Part B: Methodological*, vol. 42, pp.482-509, 2008.
[12] A. Pascale and M. Nicoli. "Adaptive Bayesian network for traffic flow prediction," in *Proc. IEEE Statistical Signal Processing Workshop*, 2011, pp. 177-180.
[13] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124-132, 2006.
[14] A. Hofleitner, R. Herring, and P. Abbeel, "Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp.1679-1693, December 2012.
[15] Y. Engel, O. Etzion and Z. Feldman, "A basic model for proactive event-driven computing," in *Proc. 6th ACM International Conference on Distributed Event-Based Systems*, 2012, pp.107-118.

Yongheng Wang was born in Hebei, China, in 1973. He received the Ph.D. degree in computer science from the National University of Defense Technology, Changsha, China, in 2006.
Since December 2008, he has been working as a teacher at the College of Information Science and Engineering, Hunan University. His research interests include data mining, event processing and big data.

©2013 Engineering and Technology Publishing