Spatio-Temporal Event Detection Using Probabilistic Graphical Models (PGMs)

Azadeh Mousavi*, Matt Duckham†, Ramamohanarao Kotagiri ‡ and Abbas Rajabifard§
* Department of Infrastructure Engineering
University of Melbourne, Victoria 3010, Australia
Email: amousavi@student.unimelb.edu.au
†Department of Infrastructure Engineering, University of Melbourne
Email: mduckham@unimelb.edu.au
‡Department of Computing and Information Systems, University of Melbourne
Email: kotagiri@unimelb.edu.au
§Department of Infrastructure Engineering, University of Melbourne, Email: abbas.r@unimelb.edu.au

***Abstract*—Event detection concerns identifying occurrence of interesting events which are meaningful and understandable. In dynamic fields, as time passes the attribute of phenomenon varies in spatial locations. Detecting events in dynamic fields requires an approach to deal with the highly granular data arriving in real time. This paper proposes a spatiotemporal event detection algorithm in dynamic fields which are monitored by wireless sensor networks (WSNs). The algorithm provides a method using probabilistic graphical models (PGMs) in WSNs to cope with the uncertainty of sensor readings. The algorithm incorporates the ability of Markov chains in temporal dependency modelling and Markov random fields theory to model the spatial dependency of sensors in a distributed fashion. Experimental evaluation of the proposed algorithm demonstrates that the decentralized approach improves the F1-score to 82% and 29% better precision than simple threshold technique. In addition, the performance of the algorithm was evaluated and compared with respect to the scalability (in terms of communication complexity). In comparison with the centralized approach the decentralized algorithm can substantially improve the scalability of communication in wireless sensor networks.**

I. Introduction

The target of this research is finding the occurrence of spatiotemporal events in dynamic fields. Due to the complexity of dynamic fields, event detection in these fields is a challenging issue. When a stream of data arrives from these fields we need to deal with the uncertainty, heterogeneity, granularity, and dynamism that increases the complexity of data processing [1]. Hence, we need an analytical techniques dealing with the complexity of dynamic fields and extract interesting knowledge from a monitoring area.

A dynamic field can be monitored by wireless sensor networks (WSNs). Wireless sensor networks are one form of distributed systems. A WSN defined as a collection of small computing devices which are able to sense and report various parameters (e.g. temperature, humidity) from the environment [2]. Automatically detecting events in WSNs facilitates the event awareness and consequently critical decision making. However, it is necessary that the data be processed with the minimal communication cost in these networks. Indeed, energy resource constraints in the network affect the longevity of sensors and encourages local processing of data through a distributed fashion. In addition, the highly granular spatial and temporal data arriving form WSNs requires analytical techniques that consider this degree of granularity and uncertainty of data [3]. This approach should be able to ignore any arbitrary event and seeks only those that are interesting in terms of validity, novelty, usefulness, and understandability [4].

Data mining (DM) as one step of knowledge discovery from data (KDD) is a response to the necessity of discovering interesting events and patterns from enormous volumes of data. Reviewing DM techniques shows that each one has different ability in mining data [4]. Among them probabilistic graphical models (PGMs) has been selected for the spatial and temporal data modelling in this paper. Considering the uncertainty of the noisy data as well as simplifying the complexity of the real world are the main advantages of PGMs. Therefore, in this paper a decentralized spatiotemporal event detection algorithm based on PGMs has been proposed. The designed framework exploits Markov chains model in exploring the occurrence of temporal events through a learning based approach. The probability of occurrence of spatial events in cliques is explored by applying Markov random field theory. We will show that the decentralized algorithm achieves the same results in terms of accuracy than the centralized algorithm and more accurate results in comparison with lower computational complexity algorithm such as simple threshold algorithm which is a common technique. In addition, the experimental evaluation of proposed algorithm demonstrates that it can improve the scalability of communication in WSNs when compared with efficient centralized event detection alternatives. The robustness of the algorithm has been evaluated by adding random noises in the observation of sensors. The performance of algorithm has been evaluated by simulating a scenario of occurrence of spatiotemporal events in the sensor network. The rest of the paper will introduce the conceptual framework of the proposed algorithm and more details about constructing this algorithm. Finally, the efficiency and accuracy of the proposed algorithm will be discussed in experimental evaluation part.

978-1-4673-5895-8/13/$31.00

II. Related Works

Event detection as an important application of wireless sensor networks has been explored by variety of techniques. In simple threshold techniques the threshold values are defined by experts and the event occurs when the sensed values exceeds the threshold or locate within a range of threshold values. In this technique join query processing is used for the application of target detection. The REED algorithm [5] uses a join query between sensor data and static table in the network to detect spatial events. This algorithm filters sensor readings with a indexed table of predicates that encodes allowable ranges of variables such as temperature. In addition [6] uses self-join monitoring queries for continuous event detection in networks. In simple threshold techniques the task of detecting a complicated predicate needs multi-phase query processing which increases the amount of communication in WSNs. In addition, it involves high false alarm rates due to the sensor's failure in networks.

In contrast with the previous work where thresholds are defined by experts, thresholds can be defined through statistical approaches. Outlier detection techniques specifically have been used in the case of defining the critical deviation points in sensed values. A histogram based technique was applied to extract hints about the distribution of data and find distance based outliers [7]. In [8] an approximate model based on temporal correlation of sensor measurements was developed to identify either distance- or density-based outliers. In [9] the spatial and temporal correlation of sensors observation were modelled using geostatistics and time series analysis respectively to distinguish the abnormal observations that significantly deviate from the normal data distribution.

Pattern matching techniques have been exploited by researchers to solve the problem of events detection in WSNs. In pattern matching technique a specific pattern is defined by expert and then is checked with sensor readings in order to detect events. This technique has been applied centrally at the base station [10] or decentrally in sensor networks [11], [12]. Similar to the simple threshold technique in pattern matching techniques the pattern can be defined by users such as NED [12], a distributed pattern matching algorithm. The NED algorithm monitor events and extracts the pre-defined pattern through a learning process. In contrast with previous works that try to find pre-defined pattern in sensed values, instead following works try to find emergence of patterns through a learning process in training data. [13] is one of the typical examples uses extensible Markov model (EMM) [14], a time varying Markov chain model, in mining emerging events based on an ageing score of occurrences. [15] explores the ability of EMM in finding rare events. The distributed algorithm of EMM has not been defined and would have to deal with high communication overhead in the sensor network. Since EMM algorithm run the task of clustering for each time steps at the base station. Running the clustering algorithm for each time steps in the network increases the total amount of communication transmission.

In another effort using learning based approaches, probabilistic graphical models for distributed inference about the events were applied to do the task of event detection in WSNs. In this approach Bayesian networks [16] and Markov random fields (MRFs) are two common techniques that have been employed by different researchers. However, in Markov networks the symmetric relationship between adjacent nodes as an undirected graph is more appropriate than Bayesian networks. In pairwise Markov random fields nodes correspond to the observation of sensors as random variables and pairwise edges represent statistical dependency between corresponding nodes. This model has been used to study the behaviour of WSNs [17]. Markov random fields is executed in WSNs using the loopy belief propagation (LBP) algorithm. LBP is a well known algorithm that has been applied in networks in order to make probabilistic inference about the state of environment and occurrence of event [18]. LBP is a simple message passing algorithm into the neighbours which simplifies the inference through MRFs [19], [20]. This algorithm is able to capture the spatial and temporal correlation of sensor data and reduce the number of false detections that are caused in the case of failures in the network. Although LBP algorithm has achieved to the promising results, there is still a gap in modelling time-varying spatial phenomena. [21] has tried to incorporate PGMs such as conditional random fields in sensor networks, but the the model does not handle data in real time. Since all sensors in the network were programmed to collect samples and send to the sink nodes and then to the computer where the data were processed.

Overall, modelling the behaviour of systems in order to find the events has been tackled by different researchers. Considering the temporal correlation as well as the spatial correlation in the observation of sensors for behaviour modelling is the missing puzzle. Event detection through modelling the spatiotemporal behaviour of dynamic fields in a decentralized approach will bridge this gap and is the goal of this paper.

III. Methodology

A. Conceptual framework

This paper aims to detect the events hidden in spatiotemporal dynamic fields. So that a synchronized wireless sensor network consists of n sensor nodes is assumed which is able to observe environmental variables (e.g. temperature, humidity, etc.) at nearly equal times. In decentralized processing of sensed data, nodes can communicate directly with each other through wireless radio transmission. Each nodes can communicate with its immediate neighbours.

In this paper, a dynamic field is defined as a spatial scalar field consisting of continuous locations in which the attributes of locations varies over the time. Temperature is an example of a scalar attributes of a dynamic field which changes in the time and locations. By defining a dynamic field as a spatial scalar field we will be able to capture the spatiotemporal variability of this filed and facilitate the detection of events.

Definition 1. A spatial scalar field is a function from domain of S as spatial planar location into co-domain V as local values, $f : S \rightarrow V$. A dynamic spatial scalar field is a function of f from a temporal domain T into a spatial field S, $f : T \rightarrow (S \rightarrow V)$, say, $f : S \times T \rightarrow V$ [22].

We assume that spatiotemporal events would occur in a dynamic spatial scalar field. In real world situations, there is a spatial relationship between the locations in which sensors are deployed as well as temporal relationship within sensor's readings that causes spatiotemporal relationships. The meaningful changes of sensed values as a result of an event exhibits strong spatiotemporal correlation in dynamic fields while a failure in a particular sensor in network can not. Therefore, according to this assumption, a spatiotemporal event occurs when the normal situation of a system changes as a result of a natural reason. This explanation of event rejects sensors malfunction or any sudden deviations (i.e, noise) from the normal state of system.

B. Spatio-temporal dependency modelling in WSNs

The methodology of this paper in detecting the spatiotemporal event in WSNs through a decentralized approach has been discussed in this section. The proposed spatiotemporal algorithm in this paper uses advantages of probabilistic graphical models (PGMs) in dynamic modelling. Generally, probabilistic graphical models (PGMs) integrate probabilistic theory and graph theory in order to deal with the problem of uncertainty and complexity of modelling real world. Using probability theory provides probabilistic representation of data (e.g, joint probability distribution of random variables) as well as providing ways to exploit inference models (e.g, hidden Markov model). The basic probabilities are established through a learning method by combining expert knowledge and accumulated data [23]. The graph theoretic side of PGMs uses a graph-based representation as the basis to encode a complex distribution over high dimensional space in which graph's nodes are random variables and the edges correspond to the probabilistic interaction between them [23]. Among different models in PGMs the proposed algorithm in this paper uses Markov chains and Markov random fields theory in detecting spatiotemporal events.

Following, first the temporal modelling process using Markov chain model will be defined and then the process of exploring spatial events by Markov random fields model will be discussed. Finally, we will present the integrated spatial and temporal model that detects the occurrence of spatiotemporal events in the sensor network.

1) Temporal dependency Modelling: A first order Markov chain assumes a finite sequence of events over discrete time points, where the future behaviour of the process at any time, $P_{i,j}$, is based solely on the current state [14].

Definition 2. A Markov Chain is a graph with m vertices or states, $S = \{N_1, N_2, ..., N_m\}$, and directed arcs, $A = \{L_{i,j} | i \in 1, 2, ..., m, j \in 1, 2, ..., m\}$ and each arc $L_{i,j}$ is labelled with a transition probability $P_{i,j} = P(S_{t+1} = N_j | S_t = N_i)$.

In the process of temporal modelling a Markov chain graph is assumed as an overlay on the communication graph [1] in a WSN. In this step all sensors individually go through a Markov process and predict the future state (Figure 1).

TABLE I
CLASSIFICATION OF STATES

Cold	Mild	Hot
$< \bar{s} - 0.5\delta$	$\geq \bar{s} - 0.5\delta, \leq \bar{s} + 0.5\delta$	$< \bar{s} + 0.5\delta$
$< \bar{s} - \delta$	$\geq \bar{s} - \delta, \leq \bar{s} + \delta$	$< \bar{s} + \delta$
$< \bar{s} - 1.5\delta$	$\geq \bar{s} - 1.5\delta, \leq \bar{s} + 1.5\delta$	$< \bar{s} + 1.5\delta$
$< \bar{s} - 2\delta$	$\geq \bar{s} - 2\delta, \leq \bar{s} + 2\delta$	$< \bar{s} + 2\delta$

Definition 3. The communication graph is an undirected graph $G = (V, E)$, where V represents a set of sensors deployed on a dynamic fields and E, edges between sensors that represents one-hop communication link between nodes $(s_i, s_j) \in V$. For a sensor in communication graph a collection of immediate neighbours $\{s_i \in V | (s_i, s_j) \in E\}$ is defined as $nbr(s)$.

Basically, the structure of Markov chain graph is defined by states and transition probabilities based on expert views or through learning process from training data. In this paper transition probabilities of states (e.g, $P_{(C-H)}$) was calculated through a learning process by counting how many times each transition of states has been taken over all state transitions in training data. In addition, the states in Markov chain graph were determined through a statistical process. The Kolmogrov-Smirnov's significance value, a statistic testes the hypothesis that the data are normally distributed, demonstrated that sensed values of sensors have normal distribution. Hence a wide ranges of boundaries were considered for classification of states in three groups of cold, mild, and hot states. Table I shows the boundaries of this three states, where the $\bar{s}$ is the mean and δ is the standard deviation of sensed values. The reason why different classes of boundaries were selected for defining the states is that we want to show that the arbitrary boundaries do not affect the performance of algorithm.

As it has been illustrated in figure 1, the sensors individually go through Markov chains process. First they define their current states and then predict their next step state.

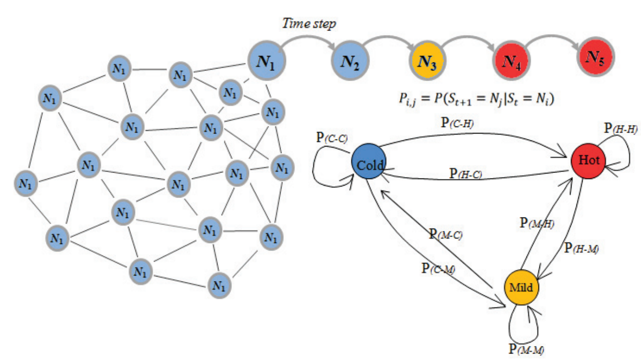

Figure 1. Markov process in the sensor network

2) Spatial dependency modelling: The spatial dependency of sensors has been explored by Markov random field (MRFs) theory. The definition of MRFs comes as follow.

Definition 4. Given an undirected graph $G = (V, E)$, a set of random variables $X = (X_v)_{v \in V}$, and $P(X = x)$ the probability of a particular states $x \in X$, then $P(X = x)$ for each nodes is conditionally independent of all other variables given its neighbours $X_u \perp\!\!\!\perp X_v | X_{V \setminus \{u,v\}} \quad \text{if } \{u, v\} \notin E$, the probability of $P(X = x)$ is taken with respect to a product measure is called a joint density. Joint density can be factorized over the cliques of $G : P(X = x) = \prod_{C \in \mathrm{cl}(G)} \phi_C(x_C)$ [24].

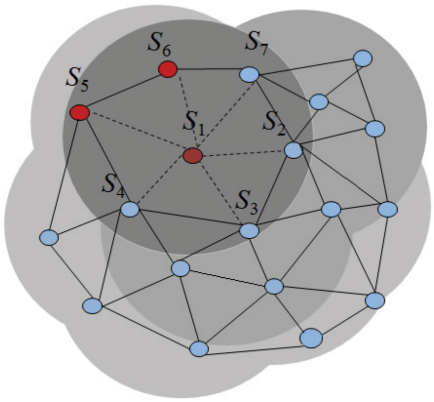

Figure 2. Clique in a wireless sensor network

Therefore, we assume that nodes are independent of the rest of nodes conditioned on the clique. In MRF, a clique is defined as a subset of the nodes in a graph such that there exists a link between all pairs of nodes in the subset (Figure 2). In this paper, the clique is considered as a subset of link-neighbours over the communication graph. Given a clique denoted by C and the set of variables in that clique by x_C, the joint distribution is written as a product of potential functions, $\psi_C(x_C)$, over the cliques of the graph $P(x) = \frac{1}{Z} \prod_c \psi_C(x_C)$. Here, the quantity Z, partition function, is a normalization constant and is given by $Z = \sum_x \prod_c \psi_C(xC)$. The quantity $P(x|X)$ denotes the probability of particular state that is observed by a sensor given the state of other sensors in the clique.

3) Determination of spatiotemporal events: The temporal and spatial dependency models are integrated to each other in order to detect the spatiotemporal events. In this process, first we identify the occurrence of temporal event then explore whether the temporal events are experienced in a spatial area.

A deviation from normal temporal behaviour of sensors would be interpreted as a noise (as a result of sensors malfunction) or as a temporal event. Since frequent occurrence of deviations rejects the possibility of noise; therefore, consecutive occurrence of deviations can be detected as temporal events. A temporal event is recorded when the predicted sensor's state through Markov chain process, $\hat{X}(s, t)$, does not meet the current state $X(s, t + 1)$ for a length of sequence. This concept has been defined in IDLE system state of the proposed algorithm (line 1-18 of algorithm 1).

That a temporal event is recorded by the nearby sensors is checked in cliques based on MRFs theory. Therefore, for the sensors which have experienced temporal events the probability of spatiotemporal event given the states of the other sensors in clique is estimated. In fact, when a sensor records unpredicted temporal state for a length of time and find itself in a clique in which the probability of unpredicted state is higher than normal state, so this event will be recorded as spatiotemporal event. This concept has been defined in TMPE system state of the proposed algorithm (line 19-29 of algorithm 1).

C. Decentralized spatiotemporal event detection algorithm

Up to this point this paper has presented a framework to detect spatiotemporal events. This section introduces a decentralized algorithm for implementing this framework. A reliable bi-directed communication graph $G = (V, E)$ and neighbourhood function $nbr : V \rightarrow 2^V$ are considered on a sensor network. The sensor function $s : V \times T \rightarrow R$ introduces the co-domain of real number R for the domain of sensor's location. Changes in attribute of location (e.g, changes in temperature) are defined as a time-varying sensor function. Thus for each time $t \in T$, $s(v, t)$ denotes the sensed value of sensors.

This algorithm uses the style of decentralized algorithm inspired by Nicola Santoro [25], [1]. There are four main components in this style which have been applied in designing the decentralized algorithm: restriction, system events, actions and states. The interaction between nodes in the network such as receipt of messages or spontaneous impulse is defined as system events. The procedures indicating how a node will respond to a particular event are specified by actions. Finally, states retain the knowledge about the previous interaction between nodes in order to respond to the subsequent events [1].

Two system states, IDLE and TMPE have been defined in this algorithm that switch into each other in response to the a particular event. Initially, all nodes simultaneously begin with IDLE system state and spontaneously read the attribute of the field $s(v, t)$ and determine their states. At each $t \in T$ nodes go through Markov chain process and predict the next step state ($Pr_{(c|X)}, Pr_{(m|X)}$, and $Pr_{(h|X)}$). The initial transition probability of states are estimated from a training data and recorded in a 3×3 matrix of M. The M matrix is introduced as an input to the algorithm. While the algorithm is executed, this matrix is updated and stored in UM matrix for each time step. At each time sensors compare the predicted state with UM matrix and decide whether any deviation occurs. Sensors which have recorded deviations for more than a length of sequences broadcast this message to their neighbours and then change their system state to TMPE (line 2-16 of algorithm 1). Sensors which have not detect any temporal event, as soon as receiving "temporal event" message from their neighbours send their sensed values to the neighbour node which has experienced temporal event (line 17-18 of algorithm 1).

Definition 5. Updated transition matrix, $UMmatrix$, is a matrix that at each time step is updated by multiplying the initial transition matrix, M and the $UMmatrix$ from the previous time step ($UM = M \times UM$).

Algorithm 1: Decentralized spatiotemporal event detection algorithm

Restrictions: Reliable, fully synchronous communication; bi-directed communication graph G=(V,E); sensor function $s : V \times T \rightarrow R$; identifier function $id : V \rightarrow N$; neighbourhood function *nbr*: V $\rightarrow 2^V$;

State Trans. Sys.: $(\{\text{IDLE}, \text{TMPE}, \}\{(\text{IDLE}, \text{TMPE}), (\text{TMPE}, \text{IDLE})\})$

Local data: Transition probability of states, $Pr(x^{t+1}|X^t)$; initial transition probability matrix $M_{(i,j)}$; counter of temporal exception Tp, initialized to $Tp := 0$; classify sensed values : $s \rightarrow X = \{c, m, h\}$; updated transition probability matrix $UM_{(i,j)}$; $MN_{(i,j)}$ matrix of pair-wise factors of states in a clique; set of states $X = \{c, m, h\}$ corresponds respectively to cold, mild, and hot; $P_{(x=X)}$ probability of being in cold, mild, or hot states given sensed values in clique; length of temporal exception sequence UDL.

Initialization: All nodes in state IDLE

IDLE

1: When time step elapsed
2: $State \leftarrow Clasify(s)$
3: **set** $UM := M \times UM$
4: **let** $a := 1$
5: **if** $X = m$ **then**
6: **let** $a := 2$
7: **if** $X = h$ **then**
8: **let** $a := 3$
9: **let** $Pr_{(c|X)} := UM_{(a,1)}$
10: **let** $Pr_{(m|X)} := UM_{(a,2)}$
11: **let** $Pr_{(h|X)} := UM_{(a,3)}$
12: **if** $X^{t+1} \neq Max(Pr(x^t))$ **then**
13: **set** $TP := Tp + 1$
14: **if** $TP \geq UDL$ **then**
15: **broadcast** $(msge, "temporalevent", id)$
16: **become** TMPE
17: *Receiving* $(msge, "temporalevent", id)$
18: **send** $(msge, "Data", s)$ to id

TMPE

19: *Receiving* $(msge, "temporalevent", s)$
20: **send** $(msge, "Data", s)$ to id
21: *Receiving* $(msge, "Data", s)$
22: **set** $MN := Matrix(C_{(v,V)})$
23: **set** $Z := \sum \prod_{v \in V} C_{(v,V)}$
24: **let** $P_{(c=X)} := \frac{1}{Z} \prod_{v \in V} C_{(v,V)}$
25: **let** $P_{(m=X)} := \frac{1}{Z} \prod_{v \in V} C_{(v,V)}$
26: **let** $P_{(h=X)} := \frac{1}{Z} \prod_{v \in V} C_{(v,V)}$
27: **if** $X = Max(P(x = X))$ **then**
28: Shout "Spatiotemporal event!"
29: **become** IDLE

Sensors in TMPE system state, when receive the message of temporal event from their neighbours send their sensed values to them (line 19-20 of algorithm 1).

In TMPE sensors which have recorded temporal events check whether they locate in a clique that experience spatiotemporal event. Pair-wise factors of decomposited joint distribution of sensor's states in cliques is estimated from the training data. The pair-wise factors are stored in MN matrix and introduces to the algorithm as an input.

Definition 6. $MNmatrix$ is a $m \times n$ matrix, where m is the number of thermal state's categories assignments (i.e, cold-cold, cold-mild, cold-hot, mild-cold, mild-mild, mild-hot, hot-cold, hot-mild, and hot-hot) and n is the number of linked neighbours in the clique.

In addition, nodes in TMPE system state when receive the message of data go though MRFs process and estimate the probability of being in each state's given the observation of sensors in the clique. When the sensor that has experienced the temporal event realize that is located in a clique in which the probability of its state is higher than the other states, then announces this event as spatiotemporal Event (line 21-29 of algorithm 1). After detecting spatiotemporal event, sensors switch to IDLE system state till the time elapse (line 1 of algorithm 1).

IV. Experiments and Analysis

A sensor network was simulated using NetLogo (http://ccl.northwestern.edu/netlogo/) and the decentralized event detection algorithm was implemented on the simulated nodes. In this experiment a WSN was simulated as an environment such that sensor nodes are agents in communication graph (constructed as unit distance graph, UDG), and edges connect all nodes as communication links .

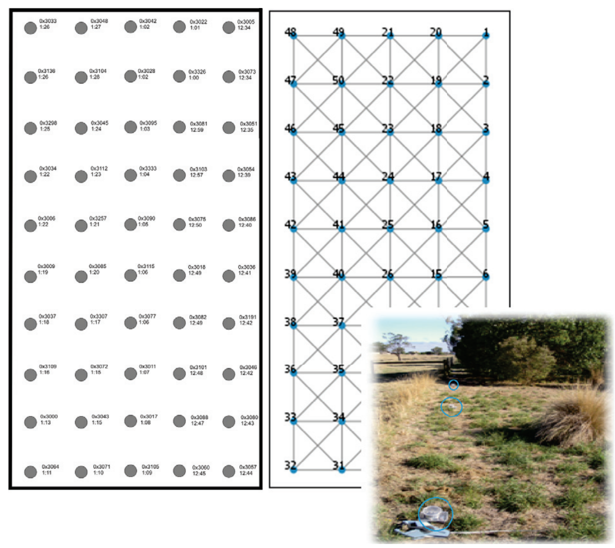

Figure 3. Communication graph and location of sensors in the field

The experiment was carried out on the real data collected from the network of 50 wireless sensors deployed on Cornet Bay in Victoria. The sensors detected the temperature of field ($50 \times 100m^2$) for the a period of six days from 6/6/2010 to 12/6/2010 with the interval of five minutes. Figure 3 shows the approximate location of these sensors and the communication link constructed as UDG graph in the network. The algorithm runs in parallel on all sensor nodes simultaneously and whenever they detect spatiotemporal events in the network then they announce it.

A. Evaluating the experiment

As it was mentioned earlier, energy resource is the main constraint in the WSNs. The resource is influenced by the rate of wireless communication through the message passing operation in sensor networks. Therefore, a critical indicator of the efficiency of an algorithm is the level of communication overhead being imposed to the network while the algorithm is executed in the network. Hence, evaluating the efficiency of algorithm in terms of message transmission will be considered as one part of algorithm evaluation. In addition, the robustness of proposed algorithm in terms of the accuracy of results is the other important issue. Evaluating the efficiency of the algorithm as well as verifying the accuracy of the results are two tasks of experiment evaluation in this paper.

1) Analysing the accuracy of algorithm: There are number of evaluation metrics are used to analyse the accuracy of algorithm based on confusion matrix. A confusion matrix visually tests the performance of algorithm. In this matrix each column represents the instances of prediction, while the rows represent the instances of actual class. Positive predictive value (PPV, also termed precision), recall, and F1-score are the accuracy metrics that are defined based on confusion matrix.

Since no interesting event has been detected by the algorithm on real data (Cornet Bay dataset), to test the accuracy of the proposed algorithm the occurrence of spatial events were simulated as underlying events on this dataset. In this simulation as time passes, hot spots appear, expand, merge, move, contract, and finally disappear in the field. These time-varying spatial events simulate the occurrence of a spatiotemporal event in the sensor network (Figure 4).

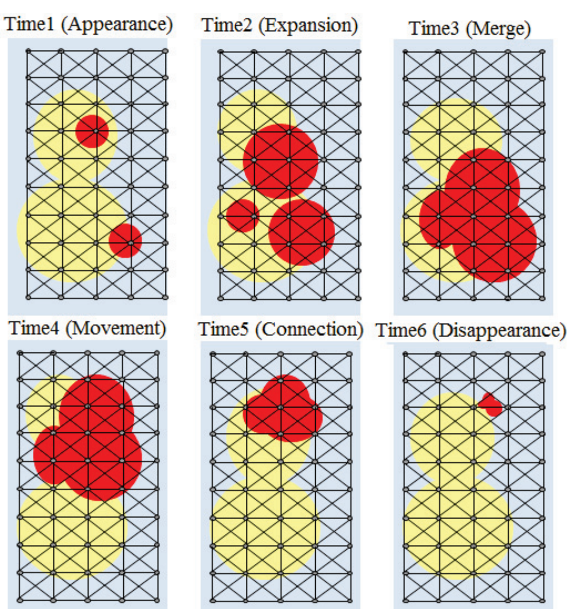

Figure 4. Simulation scenario

To check the robustness of the algorithm in case of noise some Gaussian noise ($\mu = 0, \delta = 2$) independent of the time of sensed values was added to the data. This experiment aims to test whether proposed algorithm is able to distinguish the occurrence of noise (i.e, random deviations) from meaningful events (i.e, spatially and temporally structured deviations).

TABLE II
ACCURACY METRICS

Accuracy metrics%	Proposed algorithm	Simple threshold
PPV	81.7	52.7
Recall	81.5	89.3
F1-score	82.1	72.2

If we define the number of sensors that predict the event correctly as Tp, Fp as the number of sensors that predict the event wrongly, and Fn as the number of sensors that are not predicted as event while they are, then the accuracy metrics are defined as:

$$\text{PPV}=\frac{Tp}{Fp+Tp},\ \text{Recall}=\frac{Tp}{Fn+Tp},\ \text{F1-Score}=2\times\frac{PPV\times recall}{PPV+recall}.$$

Column one of table II summarizes the average of accuracy metrics over 5 simulation runs of the proposed algorithm considering 4 classes of three states (table I). To test the performance of centralized approach, the observation of sensors at each time were forwarded to the base station through a bi-directed tree. The the algorithm was operated on the data accumulated at the base station (i.e, the root of the tree). The results demonstrated that the centralized and decentralized processing approaches achieved the same accuracy results.

The performance of proposed algorithm was compared with simple threshold techniques which are widely used for the problem of event detection in WSNs. The simple threshold algorithm has lower computational complexity rather than proposed algorithm in this paper. In simple threshold techniques the event is regarded to occur when sensor readings exceed to a range of pre-defined thresholds [5]. In our experiment the event is defined when the state of sensors changes from cold or mild states to hot state. Column two of table II summarizes the results of accuracy analysis over 5 simulation runs of the simple threshold technique considering 4 classes of three states (table I). Comparing the accuracy of proposed algorithm with threshold techniques shows that proposed algorithm has achieved to the higher precision and F1-Score. Although the simple threshold technique resulted higher level of recall, it does not show the better performance of this algorithm. Because this approach recognizes most of temporal deviations as event, while some of these deviations are noises and not necessarily meaningful events.

2) Analysing the efficiency of algorithm: The complexity analysis of models developed for WSNs is usually undertaken under high energy resource constraints of sensors in the network. Since these constraints can threaten the longevity of networks, a reasonable balance between accuracy and efficiency of algorithm is of more practical use.

The communication complexity of the developed algorithm in this research depends on the local transition of sensed values through the network. Detecting temporal events in the IDLE system state requires no communication overhead since the nodes run the algorithm individually. Whenever a temporal event is detected by sensors they broadcast this message and wait to receive the data. Figure 5 shows the total number of messages have been transmitted in different network sizes.

The graph shows a linear relationship with R^2 value of 0.59 which provides the relatively weak dependence between communication overhead and network size.

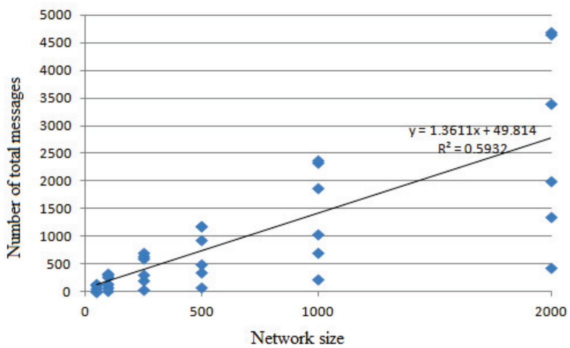

Figure 5. Analysing the scalability of proposed algorithm

The results indicate that the amount of transmitted messages does not directly depend on the number of sensors in the network but depends on the number of sensors in the regions in which events occur (Figure 6).

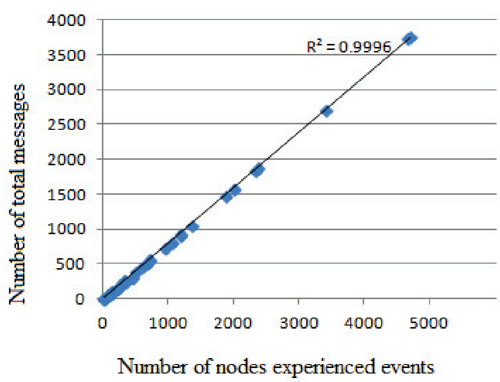

Figure 6. Transmitted messages by sensors involved in spatiotemporal events

Assuming V as the network size and V_e as the size of network which experience the spatiotemporal event then we expect $V_e \subseteq V$ and $(|V_e| < |V|)$ because at each time only some of the nodes experience spatiotemporal event. At each time nodes which have realized temporal events send one message to the neighbours, $O(1)$, and receive $O(deg(v))$ messages from the neighbours, where the $deg(v)$ is the degree of node in graph theory. Therefore, in the worst case the communication complexity of algorithm would be $O(V^2)$. However, the worst case occurs when all sensors in the network detect spatiotemporal event $(|V_e| = |V|)$. It can be assumed that the worse case occurs rarely in the network. Figure 7 illustrates the load balance of proposed algorithm. The histogram indicates a low number of messages has been sent by more number of nodes.

Centralized computation of the algorithm requires that all sensed values are transferred to the base station where they will be processed by the algorithm. To test the efficiency of decentralized approach relative to the centralized approach, a bi-directed rooted tree was constructed as an overlay on the communication graph. Every node's data was forwarded to the root where it was processed by the algorithm.

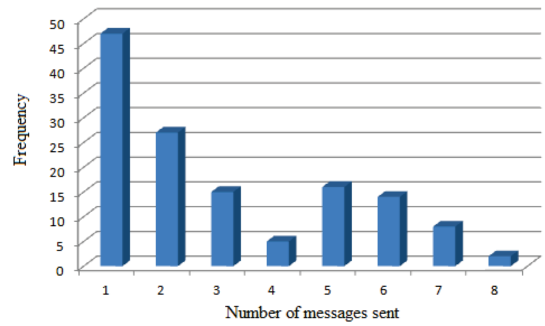

Figure 7. Load balance of proposed algorithm

The average number of total messages transmitted in the centralized and decentralized approach has been illustrated in figure 8. The results demonstrate that in comparison with the centralized approach the decentralized algorithm is relatively more efficient. However there is a limit of efficiency in decentralized computation of sensors. When the number of sensors that are involved in events increases then the amount of total message transmissions would be more than the centralized computation of events. Figure 8 compares the total amount of messages has been transmitted in the network (network size = 50) by two different centralized and decentralized approaches.

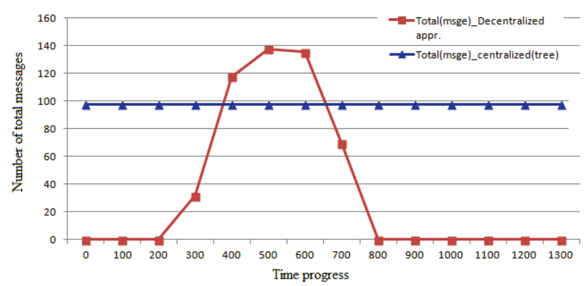

Figure 8. Comparison of the messages transition for centralized and decentralized approaches

In addition, when events occur frequently in the network then the number of message transmissions will increase. Because more number of sensors will be involved in the process of event detection. As it has been illustrated in figure 9, the straight line indicates the average number of messages were transmitted during the time in the centralized approach. Indeed, increases in the number of event occurrence causes more number of nodes be involved in event and more number of messages transmitted in the network. This may influence the efficiency of decentralized algorithm, however we can assume that occurrence of events with high frequency in big regions would rarely occur.

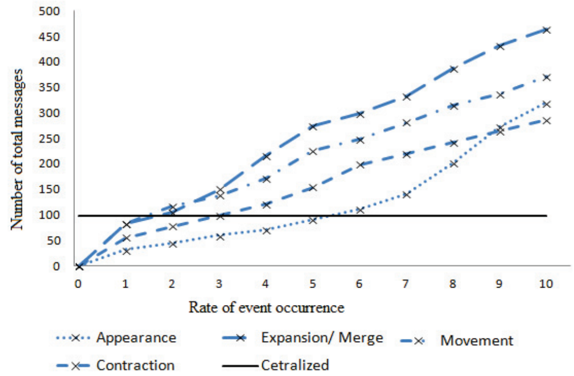

Figure 9. Limit of efficiency in proposed algorithm.

The computation complexity of this algorithm mainly depends on operating MRFs. Because as time progress temporal predication based on Markov chain model is operated in all sensors. Therefore, the main challenge in operating the algorithm would be estimating the potential function $\psi_C(x_C)$ and Z as the partition function in clique for those sensors that record temporal event . The overall computation complexity of this algorithm is $(|V_e|.|T_e|)$, where T_e is the number of times a sensor detect temporal event and V_e is the size of network experiences temporal events. The worse situation would occur when all sensors for the whole period of time sense temporal event and need to compute the possibility of event in clique based on MRF process, $O(|V|.|T|)$.

V. Conclusion

In this article the problem of finding spatiotemporal event in dynamic field which are monitored by WSNs has been investigated. The solution addresses the decentralized spatiotemporal event detection algorithm. This algorithm uses the advantages of PGMs in real world dynamic modelling. The solution has used the advantages of PGMs in dealing with the problem of uncertainty in sensor readings. The proposed decentralized algorithm exploits the ability of Markov chain model in temporal modelling and predicting the next step states of sensors. In addition, this research is applied MRFs theory in approximate spatial inference. MRFs process uses the observation of all sensor in clique to infer about the correct state. The key task in this process is that the observation of all sensors in clique are combined to cope with the uncertainty of sensor readings. The results demonstrate the better performance of this algorithm than simple threshold algorithm in terms of accuracy. Additionally, the decentralized algorithm demonstrates substantially better scalability than the centralized approach. Multi-sensor information fusion and effects of nearby sensors in neighbouring cliques will motivate us for the future work.

References

[1] M. Duckham, *Decentralized Spatial Computing: Foundations of geosensor networs*. Springer, 2012.

[2] F. Zhao and L. Guibas, *Wireless sensor networks: an information processing approach*. Morgan Kaufmann Pub, 2004.

[3] Y. Zhang, N. Meratnia, and P. Havinga, "Ensuring high sensor data quality through use of online outlier detection techniques," *International Journal of Sensor Networks*, vol. 7, no. 3, pp. 141–151, 2010.

[4] H. Miller and J. Han, *Geographic data mining and knowledge discovery*. CRC, 2009.

[5] D. Abadi, S. Madden, and W. Lindner, "Reed: Robust, efficient filtering and event detection in sensor networks," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 769–780.

[6] X. Yang, H. Lim, T. Özsu, and K. Tan, "In-network execution of monitoring queries in sensor networks," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 2007, pp. 521–532.

[7] B. Sheng, Q. Li, W. Mao, and W. Jin, "Outlier detection in sensor networks," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 219–228.

[8] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 187–198.

[9] Y. Zhang, N. Hamm, N. Meratnia, A. Stein, M. van de Voort, and P. Havinga, "Statistics-based outlier detection for wireless sensor networks," 2012.

[10] W. Xue, Q. Luo, L. Chen, and Y. Liu, "Contour map matching for event detection in sensor networks," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 145–156.

[11] F. Martincic and L. Schwiebert, "Distributed event detection in sensor networks," in *Systems and Networks Communications, 2006. ICSNC'06. International Conference on*. IEEE, 2006, pp. 43–43.

[12] G. Jin and S. Nittel, "Ned: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks," in *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*. IEEE, 2006, pp. 153–153.

[13] Y. Meng and M. Dunham, "Efficient mining of emerging events in a dynamic spatiotemporal environment," *Advances in Knowledge Discovery and Data Mining*, pp. 750–754, 2006.

[14] M. Dunham, Y. Meng, and J. Huang, "Extensible markov model," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE, 2004, pp. 371–374.

[15] Y. Meng, M. Dunham, F. Marchetti, and J. Huang, "Rare event detection in a spatiotemporal environment," in *Granular Computing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 629–634.

[16] X. Wang, J. Lizier, O. Obst, M. Prokopenko, and P. Wang, "Spatiotemporal anomaly detection in gas monitoring sensor networks," in *Proceedings of the 5th European conference on Wireless sensor networks*. Springer-Verlag, 2008, pp. 90–105.

[17] T. Wang and C. Yu, "Collaborative event region detection in wireless sensor networks using markov random fields," in *Wireless Communication Systems, 2005. 2nd International Symposium on*. IEEE, 2005, pp. 493–497.

[18] C. Crick and A. Pfeffer, "Loopy belief propagation as a basis for communication in sensor networks," in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 159–166.

[19] T. Anker, D. Dolev, and B. Hod, "Belief propagation in wireless sensor networks-a practical approach," *Wireless Algorithms, Systems, and Applications*, pp. 466–479, 2008.

[20] L. Chen, M. Wainwright, M. Cetin, and A. Willsky, "Multitarget-multisensor data association using the tree-reweighted max-product algorithm," in *Proceedings of SPIE*, vol. 5096, 2003, pp. 127–138.

[21] J. Yin, D. Hu, and Q. Yang, "Spatio-temporal event detection using dynamic conditional random fields," in *In Proceedings of IJCAI*, vol. 9, 2009.

[22] M. Duckham, S. Nittel, and M. Worboys, "Monitoring dynamic spatial fields using responsive geosensor networks," in *Proceedings of the 13th annual ACM international workshop on Geographic information systems*. ACM, 2005, pp. 51–60.

[23] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[24] R. Kindermann, J. Snell *et al.*, *Markov random fields and their applications*. American Mathematical Society Providence, RI, 1980.

[25] N. Santoro, *Design and analysis of distributed algorithms*. Wiley Online Library, 2007, vol. 509.