1

Lightweight Event Detection Scheme using Distributed Hierarchical Graph Neuron in Wireless Sensor Networks

Asad I. Khan, Anang Hudaya Muhamad Amin
and Raja Azlina Raja Mahmood
Monash University
Australia

1. Introduction

Existing breakthrough in communication technologies have lead to the rapid growth of emerging networks in particular the wireless sensor networks (WSNs). These networks emerged from the confluence of wireless communication technology, extensive computational schemes, and advanced sensor technology. WSNs are created from a collection of self-organised wireless and battery-powered devices with sensing capabilities. The future of this kind of networks is promising, as been mentioned by (Stankovic, 2008), "The potential of these systems is nothing short of revolutionary. This technology will affect all aspects of our lives, bringing about substantial improvements in a broad spectrum of modern technologies ranging from healthcare to military surveillance".

The current scenario of WSN deployment is however is still far away from its fullest potential. To date, WSN has only been demonstrated for humble applications such as meter reading in buildings and basic form of ecological monitoring. In order to achieve its fullest potential, WSN requires an intelligent computational scheme which at present is still lacking.

Common approach implemented within existing WSN applications usually involve a number of processing steps including sensory data capture and conveyance of these data to a central entity known as the base station for further refinement and analysis. Consequently, this approach would lead to a system bottleneck, if it is scaled up for widespread use. Furthermore, processing delays would intermittently occur due to the high latency between data capture/aggregation and processing time. These limitations make WSN less suitable for real-time monitoring applications. We require a new approach for an improved data processing within WSN that has the abilities to process sensory data in situ within decentralised manner and to generate highly condensed and sophisticated outputs internally. These abilities will alleviate the bottleneck problem within WSN through on-site computations, and improve the detection performance by reducing the processing delays.

In this chapter, we will describe a lightweight and distributed event detection scheme within WSN infrastructure with one-shot learning pattern recognition capability. This scheme is comprised of a distributed associative memory algorithm known as Distributed Hierarchical Graph Neuron (DHGN) (Khan and Muhamad Amin, 2007). It has the potential to recognise and classify multi-dimensional sensory input for identifying natural or man-made phenomena through clustered and hierarchical graph-based representation of input patterns for use within fully decentralized networks. In addition, DHGN divide and distribute recognition tasks throughout the network in a fine-grained manner for minimise the energy use. Thus, such scheme is highly-suited for resource-constrained networks such as WSN. In addition, DHGN is capable of integrating vast networks of sensors into intelligent macroscopes to observing our surroundings. These will bring unprecedented capabilities within the reach that transform the way we deal with phenomena occurring over large distances and inaccessible regions.

The outline for this chapter is as follows. Section 2 provides an overview of WSN technology and its current research trends. Section 3 explains the current event detection schemes within WSN and some significant issues related to it. Details of our proposed DHGN distributed pattern recognition scheme will be further described in section 4. Section 5 reports on our case study on forest fire detection using DHGN-WSN scheme. Section 6 entails further discussion on our proposed scheme and future direction of this research. Finally, section 7 concludes the chapter.

2. WSN Overview

The advancement of Wireless Sensor Network (WSN) technology has been driven by a massive development of wireless technology and an increasing miniaturisation of RF devices and microelectromechanical systems (MEMS) (Hafez et al., 2005). WSN is a collection of battery-powered and tiny electronic devices known as sensor nodes that are being used to capture sensory data from its surrounding environment. In addition, these sensor nodes are responsible for reporting the sensory readings to a centralised node, known as the base station. That possesses several orders of magnitude more processing capability than the other sensor nodes (Akyildiz et al., 2002). WSN has been used in a wide range of applications including event detection, environmental monitoring, smart home applications, and inventory management.

2.1 WSN Architecture

Every wireless sensor node is equipped with its own onboard processing, limited wireless communication, sensing module, as well as lightweight storage facilities. Each sensor node is built up from a number of electronic components including sensor, microcontroller unit (MCU) for signal controlling and processing, RF transceiver for signal transmissions (Rx and Tx), antenna, and power supply unit. Fig. 1 shows this generic wireless sensor node architecture. Currently, there is a number of commercially available wireless sensor nodes of different types for applications. These include Berkeley Mica Mote (http://www.xbow.com) and UCLA iBadge (Park et al., 2002). The specifications of the Berkeley Mica Mote sensor node that is used in many surveillance networks (Lewis, 2004, Levis and Culler, 2002) are also listed in Table 1.

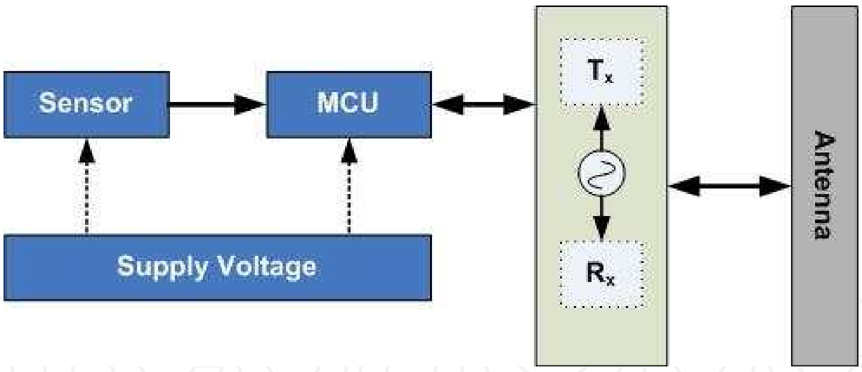


Fig. 1. Common components within a wireless sensor node

CPU:	8-bit 4 MHz
Memory :	128KB Flash and 4KB RAM
Communication :	916 MHz 40 Kbps Radio
Power :	2 AA Batteries

Table 1. Berkeley Mica Mote sensor node specifications

On a macro level, WSN is made up of of wireless sensor nodes that are linked together through a common entity known as the base station (also commonly known as sink). Due to limited power and processing capabilities, communications between sensor nodes and base station usually involve a series of data aggregation techniques to reduce the volume of traffic enroute to the base station. Data aggregation is a process which allows vast amounts of data to be communicated in an efficient manner through the use of data centric routing protocols and effective middleware. Protocols such as SPIN, LEACH, PEGASIS and SMECN are some of the known data aggregation solutions for WSNs. Fig. 2 shows a common infrastructure for WSN network.

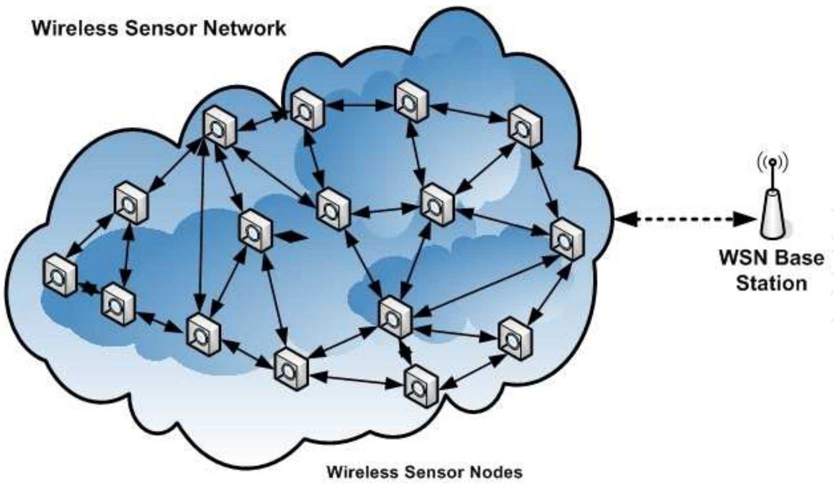


Fig. 2. Common infrastructure for WSN network

2.2 Advances in WSN technology

The advancement of WSN technology has also created a new network architecture known as wireless heterogeneous sensor network (WHSN) (Shih et al., 2006). WHSN is a new form of WSN network in which each sensor node may have a number of different sensors. An example for this kind of sensor node is the Crossbow's Mica2 mote (http://www.xbow.com) which commonly equipped with various sensors able to capture sensory readings such as temperature, humidity and light exposure. With WHSN, the capability of the sensor network to provide various range of sensor readings are extended. Consequently, this makes the network capable of detecting any occurrences of significant event by observing multiple parameters in comparison to a single parameter observation.

Another form of WSN network has recently being introduced, known as the mobile WSN (Benyuan et al., 2005, Wang et al., 2008). Mobile WSN derives its name from the presence of mobile sink or sensor nodes. It provides greater deployment flexibility in comparison to the static WSN architecture. Mobile WSNs have also been found to demonstrate enhanced performance over static WSNs (Munir et al., 2007).

2.3 WSN Deployment Issues

Issues with WSN deployment across wide area of applications encircled towards its resource-constrained characteristics, which include limited communication bandwidth, power, processing capability and memory capacity (Culler et al., 2004). In addition, any algorithm that entails computations, communications, or storage resources within a sensor node would lead to quick exhaustion of the limited battery power available per node. The limited energy and computational resources of sensors imply that the data processing and transmission must be kept to a minimum level in order to conserve energy.

In solving these issues, systems designers must be able to produce a well-managed design for WSN deployment in which it will provide long-term reliability to the network. An effective design should include some important principles such as data-centric mechanism, localised algorithms, and lightweight middleware. In this chapter, we propose a new design for WSN deployment for event detection which incorporates these principles for highly-reliable sensor networks.

3. WSN Event Detection

One of the primary purposes of the existence of WSN is to provide capabilities for monitoring, detecting, and reporting various significant occurrences of events in the sensory domain. An event can be defined as a behavioural change over time on a certain dynamic phenomenon (Guralnik and Srivastava, 1999). An example of event is the change in rainfall amount, ranging from light to heavy to extreme. The behavioural change mentioned here could be either a change involving single environmental parameter value or changes involving composite parameters. In explaining this, (Li et al., 2004) proposed an event hierarchy terminology that differentiates between atomic events and compound events. Atomic event can be determined based on an observation of a sensor, while compound event cannot be determined from a single observation. Rather, compound event is a collection of observations on different types of sensors. For instance, forest fire is a compound event in which observations could be made on four different parameters including temperature, relative humidity, wind speed, and rainfall.

In relation to event detection using WSN, the direction of research in this area is more towards developing energy-efficient, scalable, and reliable scheme to be used within this resource-constrained network.

Research in the area of event detection using WSN could commonly be classified into two groups: performance-specific research and application-specific research. The performance-specific research concerns with the efficiency of the event detection scheme. The main research goal is to develop an event detection scheme with minimum energy consumption and extended lifetime of the WSN network. Alternatively, application-specific research focuses on the development of event detection mechanism that provides accurate and reliable detection for predefined applications such as intrusion detection or phenomenon detection. The common goal of this research area is to obtain efficient mechanism for event detection that deploys specific data processing algorithm that is able to provide accurate and reliable detection using WSN network. This section will further describe the two common research areas of event detection.

3.1 Performance-specific Event Detection Schemes

Most of the recent works on performance-specific event detection schemes are focusing on efficient localisation method and routing mechanism that could be deployed within a WSN network. Localisation and routing are the two important factors in determining the optimum coverage and performance of WSN network. Furthermore, these works have also considered multiple event detection scenarios.

A collaborative event detection and tracking in wireless heterogeneous sensor networks (WHSN) has been proposed by (Shih et al., 2008). In their work, emphasis has been put into tracking procedure and localisation of sensors' attribute region for event detection. Event detection scheme known as CollECT (Collaborative Event Detection and Tracking) has been introduced. A collaboration of different types of sensor nodes is used for event detection and tracking. The three main procedures involved are vicinity triangulation, event determination, and border sensor node selection. The scheme allows event detection and tracking to be conducted simultaneously. However, the scheme requires significant distinction of sensor nodes and their attributes according to its sensing capability. Furthermore, it also requires extensive collaboration of sensor nodes to derive towards maximum accuracy in the event detection within WHSN.

(Banerjee et al., 2008) introduces multiple-event detection scheme with fault tolerant within WSN. They propose the use of polynomial-based scheme that addresses the problems of Event Region Detection (PERD). There are two important components involved, which are the event recognition and event report with boundary detection. For event recognition, (Banerjee et al., 2008) adopts min-max classification scheme which classifies event according to the sensor reading values. These values would then be transformed into polynomial coefficients and passed through a data aggregation scheme. The proposed event detection scheme has enabled a 33% savings in the communication overhead experienced by the network.

Another important contribution in this event detection with performance-specific research is on the works conducted by (Ai et al., 2009) in Authentic Delay Bounded Event Detection System (ADBEDS) for WHSN. ADBEDS implements iterative event detection scheme using event detection tree. This system does simultaneous event detection and packet routing. ADBEDS support singular and composite event monitoring. Important aspects within ADBEDS implementation include energy efficiency and authenticity within WSN deployment for event detection. ADBEDS implements user-specified bounded delay for event detection. Energy efficiency is achieved through sleep-awake alternation between sensor nodes.

3.2 Application-specific Event Detection Schemes

Application-specific schemes for event detection refer to the area of research involving development of application middleware for WSN. This middleware provides enhanced capability and accuracy for event detection using sensor networks. Several machine learning algorithms have been applied including Fuzzy-ART neural network, multi-layer perceptrons (MLPs), and Self-Organising Maps (SOMs).

The use of Adaptive Resonance Theory (ART) neural network for event tracking was introduced by (Kulakov and Davcev, 2005b). Further classification scheme for event detection within WSN has also been introduced in (Kulakov and Davcev, 2005a). In these works, the use of artificial neural networks (ANNs) in the form of ART network has been used as pattern classifier for event detection and classification. The scheme offers reduction in communication overhead with only cluster labels being sent to the sink, instead of the overall sensory data. However, the implementation of ART neural network incurs excessive iterative cycle to achieve optimum cluster matches.

The works by (Kulakov and Davcev, 2005b) on ART neural network for event tracking has also been further researched by (Li and Parker, 2008) in their works on intruder detection using a WSN with fuzzy-ART neural networks.

Self-organisation for event detection has also been a major focus in application specific research within WSN networks. (Elaine et al., 2003) propose a concept of distributed event classification through the use of Kohonen self-organising map (SOM) approach (Kohonen, 2001). The occurrence of events, which are signified by changes in sensor parameter values, could be mapped into clusters representation. The proposed scheme however, imposes significant iterative learning procedure and the classification process is carried out on each input unit, rather than collective input units.

3.3 Summary

Existing event detection schemes within wireless sensor network commonly involve centralised processing at the sink or base station. Efforts to minimise the tendency for this centralised or singular processing have been shown by the works of both performance and application-specific research works. However, a complete decentralisation processing for event detection has yet to be achieved. There are several factors related to this issue. These include complex learning algorithms for event detection and tightly-coupled schemes being deployed for event detection.

We can see by the works of (Kulakov and Davcev, 2005b) and (Elaine et al., 2003) that extensive learning procedures are required in order to derive clusters of events. Consequently, the inputs from the sensors would need to be processed separately and thus incur additional communication overhead for inter-nodes communication. In addition, the proposed schemes do not take into account the variable data processing latency for each sensor node, that is some nodes may require longer processing time than the others.

The works conducted by (Shih et al., 2008) and (Banerjee et al., 2008) offer significant contribution in the efficiency of communication schemes for event detection using WSN. However, the tendency for centralised processing is somewhat undeniable. Furthermore, approaches for distinguishing different roles of specific nodes within WSN are still within a scope of further discussion, due to the nature of WSN network which consists of uniformly-equivalent resource-constrained sensor nodes.

In this chapter, we propose a holistic solution for event detection using WSN. It incorporates a distributed pattern recognition scheme within WSN network and provides on-site and localised computation. We implement a single-cycle learning distributed pattern recognition algorithm known as Distributed Hierarchical Graph Neuron (DHGN) (Khan and Muhamad Amin, 2007). Within this scheme, a dimensionality reduction approach has been employed for minimising the need for complex computation, as well as the incurrence of communication overhead within the network. Our proposed scheme is also capable of providing scalable detection in which we are able to cater for the outgrowth of event classes. Furthermore, integration with computational grid for complex event analysis is viable through this scheme is. Finally, our proposed lightweight event detection scheme also equipped with a detailed workflow of the event detection process. The following sections will provide further descriptions of our proposed solution.

4. Distributed Hierarchical Graph Neuron (DHGN)

DHGN is a novel distributed associative memory (AM) algorithm for pattern recognition. The main idea behind this algorithm is that common pattern recognition approaches for various kinds of patterns would be able to be conducted within a body of a network. DHGN shifts the recognition algorithm paradigm from employing CPU-centric processing towards network-centric processing approach. It also adopts single-cycle learning with in-network processing capability for fast and accurate recognition scheme.

The basic foundation of DHGN algorithm is based upon the functionalities and capabilities of two other associative memory algorithms known as Graph Neuron (GN) (Khan, 2002) and Hierarchical Graph Neuron (HGN) (Nasution and Khan, 2008). It eliminates the crosstalk issue in GN implementation, as well as reduces the complexity of HGN algorithm by reducing the number of processors required for its execution. DHGN is also a lightweight pattern recogniser that supports adaptive granularity of the computational network, ranging from fine-grained networks such as WSN to coarse-grained networks including computational grid.

DHGN network consists of a collection of DHGN processing clusters (PCs) that are interconnected through an important processing entity known as Collective Recognition Unit (CRU), which is responsible for collection of recognition results from each DHGN PC. Fig. 3 shows the basic architecture for DHGN network.

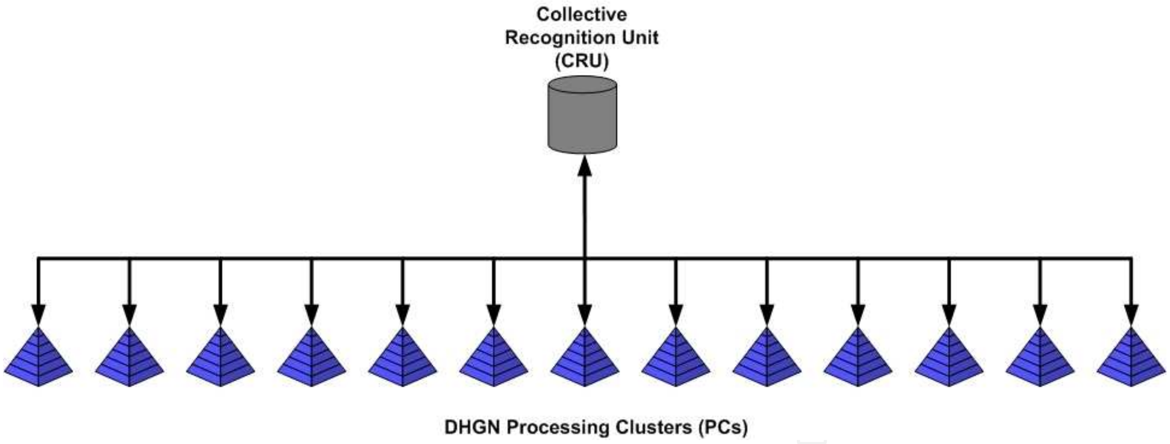


Fig. 3. DHGN network architecture

DHGN processing cluster (PC) is a structural formation of recognition entities called processing elements (PEs) as shown in Fig. 4. The formation is a pyramid-like composition where the base of the structure represents the input patterns. Pattern representation within DHGN network is in the form of *[value, position]* format. Fig. 5 shows how character pattern "AABCABC" is represented in DHGN algorithm.

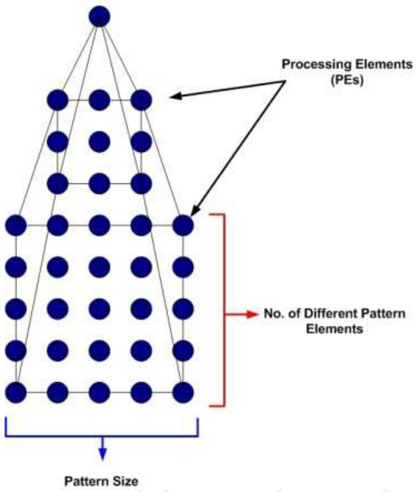


Fig. 4. DHGN processing cluster (PC) formation consists of a number of processing elements (PEs)

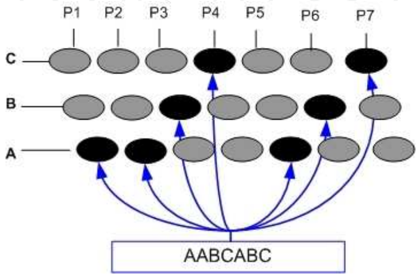


Fig. 5. Pattern representation within DHGN algorithm. Each element within a pattern is represented with *[value, position]* format,

Each row in this representation forms the pattern's possible values v, while each column represents the position of each value within the pattern, p. Therefore, the number of columns within this formation is equivalent to the size of the pattern. In this manner, each location-assigned PE will hold a single value. The formation of the input representation at the base of DHGN processing cluster could be derived from the number of PEs, n_{PE} at the base level of the PC, as shown in Equation (1):

$$n_{PE} = pv \tag{1}$$

4.1 DHGN Recognition Process

Recognition process within DHGN involves a single-cycle learning of patterns on a distributed processing manner. Unlike other pattern recognition algorithms such as Hopfield Neural Network (HNN) (Hopfield and Tank, 1985) and Kohonen SOM (Kohonen, 2001), DHGN employs in-network processing feature within the recognition process. This processing capability allows the recognition process to be performed by a collection of lightweight processors (referred to PEs). PE is an abstract representation of processor that could be in the form of a specific memory location or a single processing node.

At macro level, DHGN pattern recognition algorithm works by applying a divide-and-distribute approach to the input patterns. It involves a process of dividing a pattern into a number of subpatterns and the distribution of these subpatterns within the DHGN network as shown in Fig. 6.

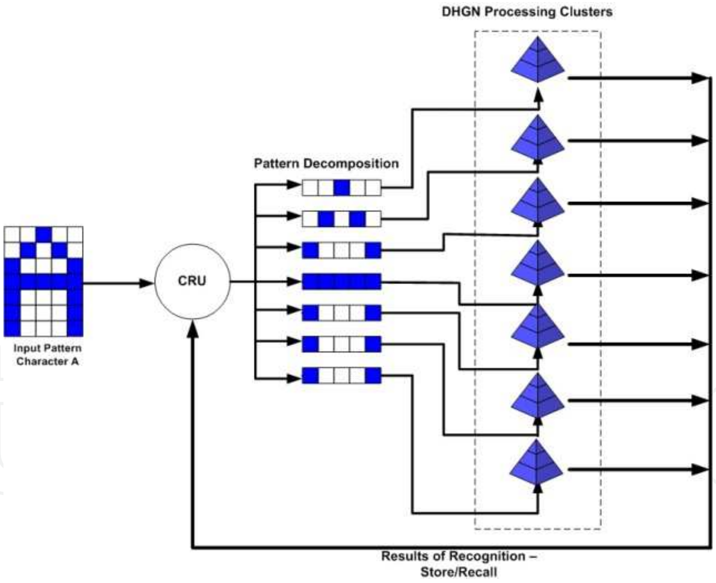


Fig. 6. Divide-and-distribute approach in DHGN distributed pattern recognition algorithm. Character Pattern 'A' is decomposed into similar size subpatterns

In this work, we have made an assumption that a pattern P is a series of data in the form of *[value, position]*, as shown in Equation (2):

$$P = \{v_1, v_2, \ldots, v_x\}, \quad x \in \mathbb{N}^+ \tag{2}$$

Where v represents element within a pattern and x represents the maximum length of the given pattern. For an equal distribution of subpatterns into DHGN network, the Collective Recognition Unit (CRU) firstly needs to determine the capacity of each processing cluster. The following equation shows the derivation of the size of subpattern for each processing cluster from the pattern size x and the number of processing clusters s_n available, assuming that each processing cluster has equal processing capacity:

$$s_{size} = \frac{x}{s_n} \tag{3}$$

Each DHGN processing cluster holds a number of processing elements (PEs). The number of PEs required, PE_n is directly related to the size of the subpattern, s_{size} and the number of possible values, v:

$$PE_n = v\left(\frac{s_{size}+1}{2}\right)^2 \tag{4}$$

Within each DHGN processing cluster, PEs could be categorised into three categories as shown in Table 2.

Type	Description
Base-Layer PE	Responsible for pattern initialisation. Pattern is introduced to DHGN PC at the base layer. Each PE holds a respective element value on specific location within the pattern structure.
Middle-Layer PE	Core processing PE. Responsible to keep track on any changes on the activated PEs at the base-layer and/or its lower middle-layer.
Top-Layer PE	Pre-decision making PE. Responsible for producing final index for a given pattern.

Table 2. Processing element (PE) categories

At micro level, DHGN adopts an adjacency comparison approach in its recognition procedures. This approach involves comparison of values between each processing elements (PEs). Each PE contains a memory-like structure known as bias array, which holds the information from its adjacent PE within the processing cluster. The information kept in this array is known as bias entry with the format *[index, value, position]*. Fig. 7 shows the representation of PE with bias array structure.

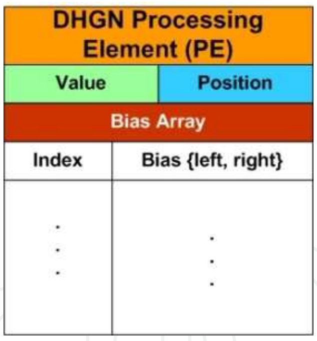


Fig. 7. Data structure for DHGN processing element (PE)

Fig. 8 shows inter-PE communication within a single DHGN processing cluster. The activation of base-layer PE involves matching process between PE's and the pattern element's *[value, position]*. Each activated PE will then initiate communication between its adjacent PEs and conducting bias array update. Consequently, each activated PE will send its recalled/stored index to the PE at the layer above it, with similar position, with exception of the PEs at the edges of the map.

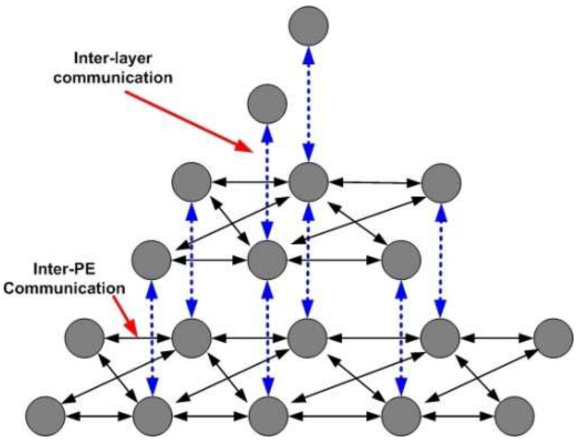


Fig. 8. Communications in DHGN processing cluster (PC)

Unlike other associative memory algorithms, DHGN learning mechanism does not involve iterative modification or adjustment of weight in determining the outcome of the recognition process. Therefore, fast recognition procedure could be obtained without affecting the accuracy of the scheme. Further literature on this adjacency comparison approach could be found in (Khan and Muhamad Amin, 2007, Muhamad Amin and Khan, 2008a, Muhamad Amin et al., 2008, Raja Mahmood et al., 2008).

4.2 Data Pre-processing using Dimensionality Reduction Technique

Event detection usually involves recognition of significant changes or abnormalities in sensory readings. In WHSN, specifically, sensory readings could be of different types and values, e.g. temperature, light intensity, and wind speed. In DHGN implementation, these data need to be pre-processed and transformed into an acceptable format, while maintaining the original values of the readings.

In order to achieve a standardised format for pattern input from various sensory readings, we propose the use of adaptive threshold binary signature scheme for dimensionality reduction and standardisation technique for multiple sensory data. This scheme has originally been developed by (Nascimento and Chitkara, 2002) in their studies on content-based image retrieval (CBIR). Binary signature is a compact representation form that capable of representing different types of data with different values using binary format.

Given a set of n sensory readings $S = \left(s_1, s_2, \ldots, s_n\right)$, each reading s_i would have its own set of k threshold values $P_{s_i} = \left(p_1, p_2, \ldots, p_k\right)$, representing different levels of acceptance. These values could also be in the form of acceptable range for the input. The following procedures show how the adaptive threshold binary signature scheme is being conducted:

a. For each sensor reading s_i, is discretised into j binary bins $\left(B^i = b_1^i\, b_2^i \ldots b_j^i\right)$ of equal or varying capacities. The number of bins used for each data is equivalent to the number of threshold values P_{s_i}. This bin is used to signify the presence of data which is equivalent to the threshold value or within a range of the specified p_i values using binary representation.

b. Each bin would correspond to each of the threshold values. Consider a simple data as shown in Table 3. If the temperature reading is between the range 20 – 25 degrees Celsius, the third bin would be activated. Thus, a signature for this reading is "01000".

c. The final format of the binary signature for all sensor readings would be a list of binary values that correspond to specific data, in the form of $S_{bin} = b_1^1 b_2^1 b_1^2 b_2^2 \ldots b_j^n$, where b_j^k represent the binary bin for k th sensor reading and j th threshold value.

Temperature Threshold Range (°C)	Binary Signature
0 – 20	10000
21 - 40	01000
41 – 60	00100
61 – 80	00010
81 - 100	00001

Table 3. Simple dataset with its respective binary signature

4.3 DHGN Integration for WSN

With distributed and lightweight features of DHGN, an event detection scheme for WSN network can be carried out at the sensor node level. It could act as a front-end middleware that could be deployed within each sensor nodes in the network, forming a network of event detectors. Hence, our proposed scheme minimises the processing load at the base station and provides near real-time detection capability. Preliminary work on DHGN integration for WSN has been conducted by (Muhamad Amin and Khan, 2008b). They have proposed two distinctive configurations for DHGN deployment within WSN.

In integrating DHGN within WSN for event detection, we have considered mapping each DHGN processing cluster into each sensor node. Our proposed scheme is composed of a collection of wireless sensor nodes and a sink. We consider a deployment of WSN in two-dimensional plane with w sensors, represented by a set $W = (w_1, w_2, \ldots, w_n)$, where w_i is the ith sensor. The placement for each of these sensors is uniformly located in a grid-like area, $A = (x \times y)$, where x represents the x-axis coordinate of the grid area and y represents the y-axis coordinate of the grid area. Each sensor node will be assigned to a specific grid area as shown in Fig. 9. The location of each sensor node is represented by the coordinates of its grid area (x_i, y_i).

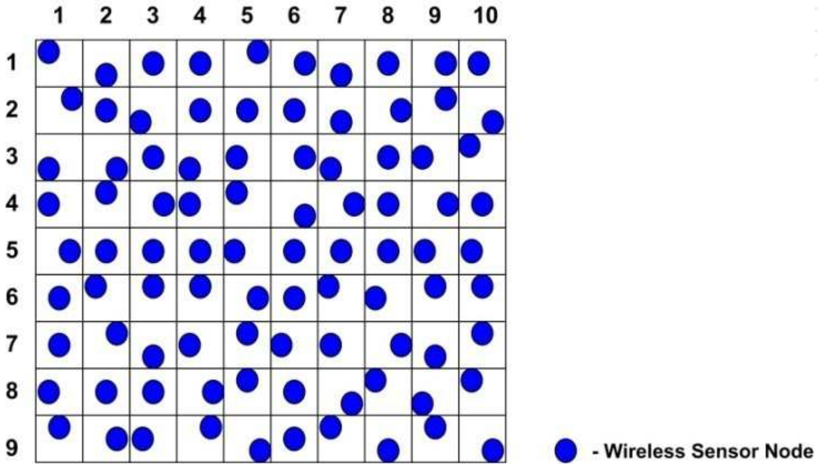


Fig. 9. Sensor nodes placement within a Cartesian grid. Each node is allocated to a specific grid area

For its communication model, we adopt a single-hop mechanism for data transmission from sensor node to the sink. We suggest the use of "autosend" approach as originally proposed by (Saha and Bajcsy, 2003) to minimise error due to the lost of packets during data transmission. Our proposed scheme does not involve massive transmission of sensor readings from sensor nodes to the sink, due to the ability for the front-end processing. Therefore, we believe that a single-hop mechanism is the most suitable approach for DHGN deployment. On the other hand, the communication between the sink and sensor nodes is done using broadcast method.

4.4 Event Classification using DHGN

DHGN distributed event detection scheme involves a bottom-up classification technique, in which the classification of events is determined from the sensory readings obtained through WSN. As been discussed before, our approach implements an adaptive threshold binary signature scheme for pattern pre-processing. These patterns would then be distributed to all the available DHGN processing clusters for recognition and classification purposes.

The recognition process involves finding dissimilarities of the input patterns from the previously stored patterns. Any dissimilar patterns will create a respond for further analysis, while similar patterns will be recalled. We conduct a supervised single-cycle learning approach within DHGN that employs recognition based on the stored patterns. The stored patterns in our proposed scheme include a set of ordinary events that could be translated into normal surrounding/environmental conditions. These patterns are derived from the results of an analysis conducted at the base station, based upon the continuous feedback from the sensor nodes. Fig. 10 shows our proposed workflow for event detection.

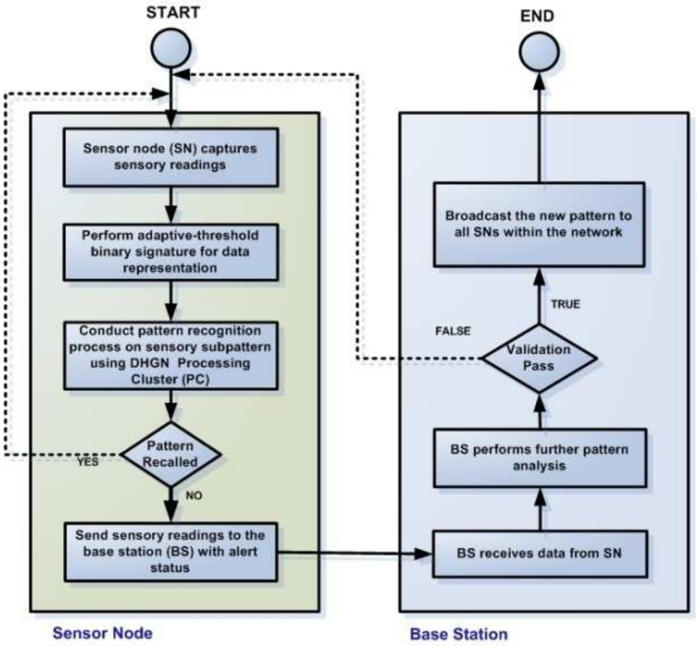

START
END
Sensor node (SN) captures sensory readings
Perform adaptive-threshold binary signature for data representation
Conduct pattern recognition process on sensory subpattern using DHGN Processing Cluster (PC)
Pattern Recalled
YES
NO
Send sensory readings to the base station (BS) with alert status
BS receives data from SN
BS performs further pattern analysis
Validation Pass
FALSE
TRUE
Broadcast the new pattern to all SNs within the network
Sensor Node
Base Station

Fig. 10. DHGN distributed pattern recognition process workflow

Our proposed event detection scheme incorporates two-level recognition: front-end recognition and back-end recognition. Front-end recognition involves the process of determining whether the sensor readings obtained by the sensor nodes could be classified as extraordinary event or simply a normal surrounding condition. On the other hand, the spatial occurrence detection is conducted by the back-end recognition. In this approach, we consider the use of signals sent by sensor nodes as possible patterns for detecting event occurrences at specific area or location. In this chapter, we will explain in more details on our front-end recognition scheme.

4.5 Performance Metrics

DHGN pattern recognition scheme is a lightweight, robust, distributed algorithm that could be deployed in resource-constrained networks including WSN and Mobile Ad Hoc Network (MANET). In this type of networks, memory utilisation and computational complexity of the proposed scheme are two factors need to be highly considered. The performance of the scheme largely relies on these major factors.

A. Memory utilisation

Memory utilisation estimation for DHGN algorithm involves the analysis of bias array capacity for all the PEs within the distributed architecture, as well as the storage capacity of the Collective Recognition Unit (CRU). In analysing the capacity of the bias array, we observe the size of the bias array, as different patterns are being stored. The number of possible pattern combinations increases exponentially with an increase in the pattern size. The impact of the pattern size on the bias array storage is an important factor in bias array complexity analysis. In this regard, the analysis is conducted by segregating the bias arrays according to the layers within a particular DHGN processing cluster.

The following equations show the bias array size estimation for binary patterns. This bias array size is determined using the number of bias entries recorded for each processing element (PE). In this analysis, we have considered a DHGN implementation for one-dimensional binary patterns; wherein a two dimensional pattern is represented as a string of bits.

Base Layer. For each non-edge PE, the maximum size of the bias array:

$$bs_{ne}^{l_0} = n_r^2 \tag{5}$$

Where n_r represents the number of rows (different elements) within the pattern. For each PE at the edge of the layer:

$$bs_e^{l_0} = n_r \tag{6}$$

The cumulative maximum size of bias arrays at the base layer in each DHGN processing cluster could be derived as shown in Equation (7):

$$bs_{total}^{l_0} = n_r(bs_{ne}^{l_0}(s_{size} - 2) + 2bs_e^{l_0}) \tag{7}$$

The maximum size of bias array, i.e. the total number of bias entries at the base layer is mostly determined by the number of possible combinations of values within a pattern.

Middle Layers. The maximum size of the bias array at a middle layer depends on the maximum size of the bias array at the layer below it. For non-edge PE in a middle layer, the maximum size of its bias array may be derived as follows:

$$bs_{ne}^{l_i} = bs_{ne}^{l_i - 1} * n_r^2 \tag{8}$$

For each PE at the edge, the maximum size of its bias array could be derived as the following:

$$bs_e^{l_i} = bs_e^{l_i - 1} * n_r \tag{9}$$

Therefore, the cumulative maximum size of bias arrays in a middle layer (of a processing cluster) could be estimated using the following equation:

$$bs_{total}^{l_i} = \sum_{i=1}^{l_{top}-1} n_r \left(bs_{ne}^{l_i} \left(s_{size} - (2i + 2) \right) + 2bs_e^{l_i} \right) \tag{10}$$

Top Layer. At the top layer, the maximum size of the bias array could be derived from the preceding level non-edge PE's maximum bias array size. Hence, the maximum size of the bias array of PE at the top level is:

$$bs_{all}^{l_{top}} = n_r * bs_{ne}^{l_{top}-1} \tag{11}$$

From these equations, the total maximum size of all the bias arrays within a single DHGN processing cluster could be deduced as shown in Equation (12):

$$bs_{total}^{DHGN} = bs_{total}^{l_0} + \sum_{i=1}^{l_{top}-1} bs_{total}^{l_i} + bs_{all}^{l_{top}} \tag{12}$$

From these equations, one could derive the fact that DHGN offers efficient memory utilisation due to its efficient storage/recall mechanism. Furthermore, it only uses small memory space to store the newly-discovered patterns, rather than storing all pattern inputs. Fig. 11 shows the comparison between the estimated memory capacities for DHGN processing cluster with increasing subpattern size against the maximum memory size for a typical physical sensor node (referring to Table 1).

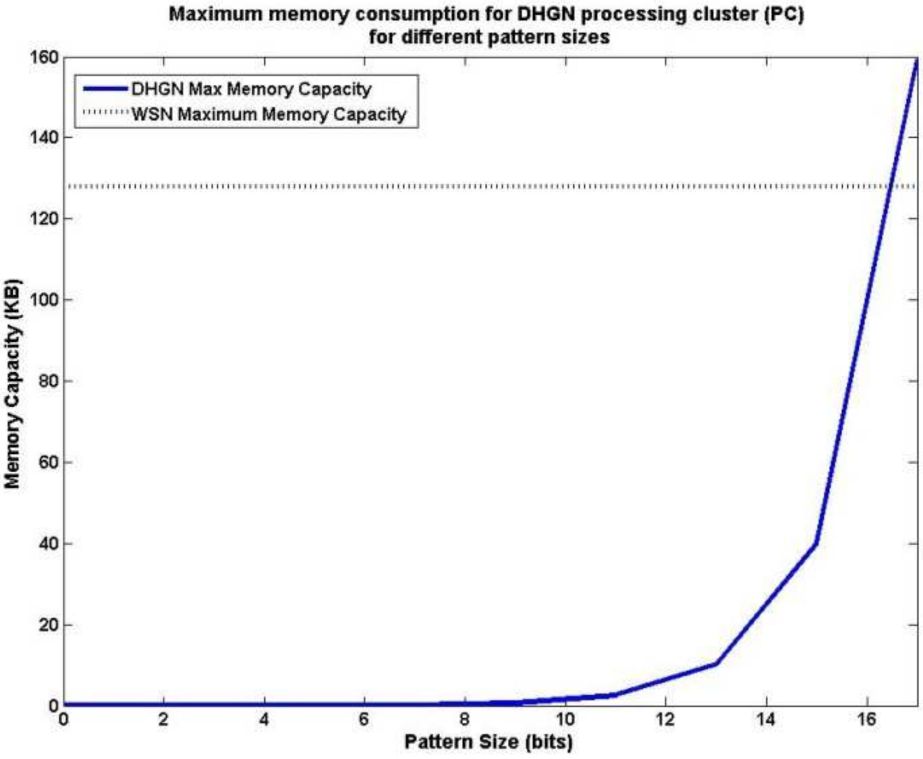


Fig. 11. Maximum memory consumption for each DHGN processing cluster (PC) for different pattern sizes. DHGN uses minimum memory space with small pattern size

As the size of subpattern increases, the requirement for memory space is considerably increases. It is best noted that small subpattern sizes only consume less than 1% of the total memory space available. Therefore, DHGN implementation is best to be deployed with small subpattern size.

B. Computational complexity

Computational complexity of DHGN distributed pattern recognition algorithm could be observed from its single-cycle learning approach. A comparison on computational complexity between DHGN and Kohonen's self-organising map (SOM) has been prescribed by (Raja Mahmood et al., 2008).

Within each DHGN processing cluster, the learning process consists of the following two steps: (i) submission of input vector x in orderly manner to the network array, and (ii) comparison between the subpattern with the bias index of the affected PE, and respond accordingly. There are two main processes in DHGN algorithm: (i) network initialisation, and (ii) recognition/classification. In the network initialisation stage, we are interested to find the number of created processors (PE) and the number of PEs that are initialised. In DHGN, the number of generated PEs is directly related to the input pattern's size. However, only the processors at the base layer of the hierarchy are initialised. Equation (13) shows the number of PEs in DHGN PE_{DHGN}, given the size of the pattern P_{size}, the size of each DHGN processing cluster N_{DHGN}, and the number of different elements within the pattern e:

$$PE_{DHGN} = e.\left(\frac{\frac{S}{N_{DHGN}}+1}{2}\right)^2 \qquad (13)$$

The computational complexity for the network initialisation stage, I_{DHGN} for n number of iterations, could be written as in Equation (14):

$$f\left(I_{DHGN}\right) = O(n) \qquad (14)$$

This equation proves that DHGN's initialization stage is a low-computational process. Fig. 12 shows the estimated time for this process. Similar speed assumption of 1 microsecond (µs) per instruction is applied in this analysis. It can be seen that the time taken in the initialization process of DHGN takes approximately only 0.2 seconds to initialize 20,000 nodes.

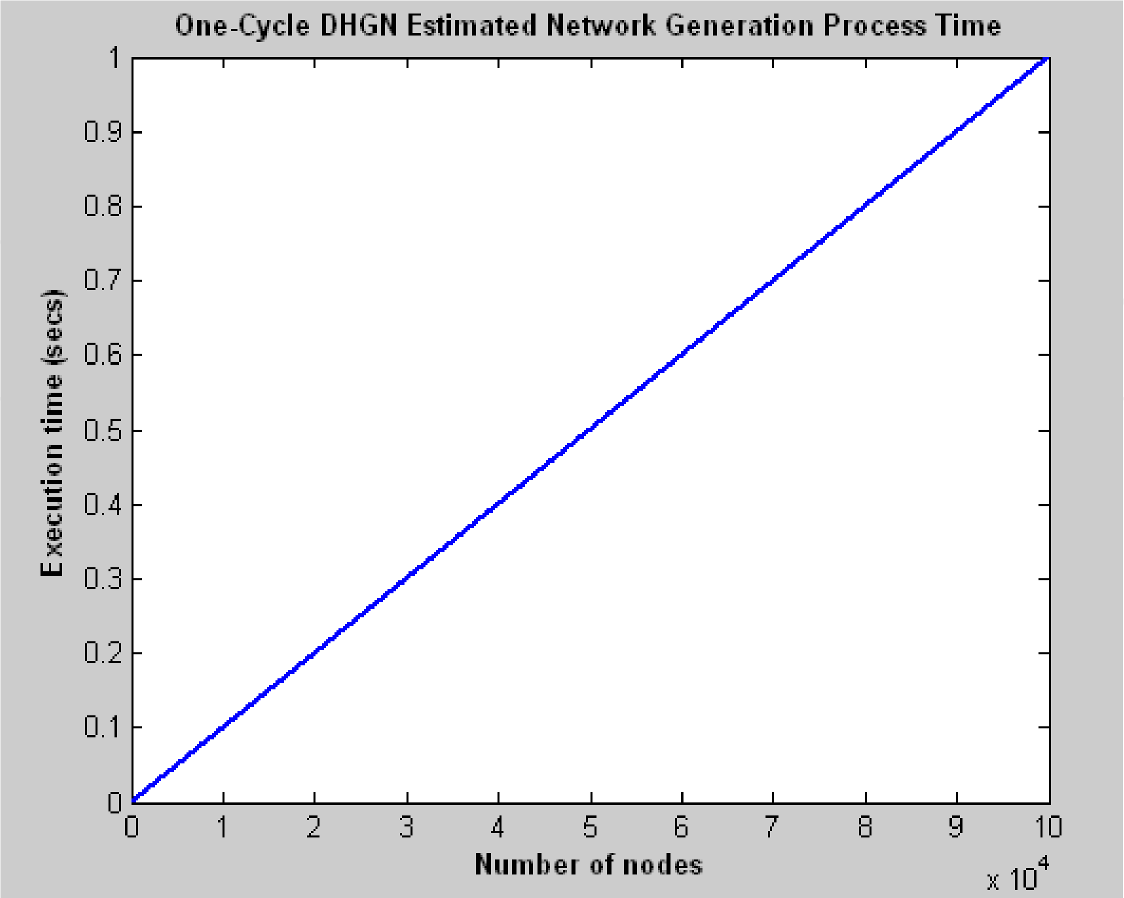


Fig. 12. Complexity performance of DHGN's network generation process (Adopted from (Raja Mahmood et al., 2008))

In the classification process, only few comparisons are made for each subpattern, i.e. comparing the input subpattern with the subpatterns of the respective bias index. The computational complexity for the classification process is somewhat similar to the network generation process, except an additional loop is required for the comparison purposes. The pseudo code of this process is as follows:

```
for each PE in the cluster
{
     recognition()
    {
          for each bias entry
          {
              check whether input index is similar to stored index
           }
     }
     classification()
}
```

From this pseudo code, the complexity of the classification process C_{DHGN} for n number of iterations could be written as the following equation:

$$f\left(C_{DHGN}\right) = O\left(n^2\right) \tag{15}$$

It can be seen from Equation (15) that DHGN's classification process requires low computational complexity. The time taken for classification by DHGN in a network of 50,000 nodes is less than 3 seconds, as shown in Fig. 13. This exponential effect is still low in comparison to other classification algorithms, including SOM (Raja Mahmood et al., 2008).

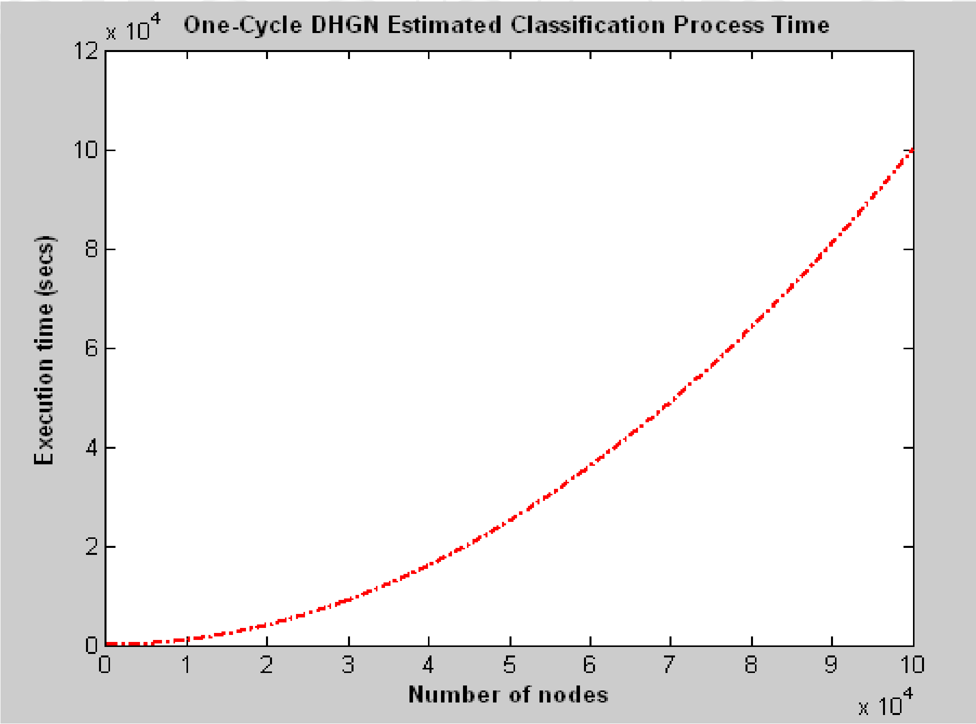


Fig. 13. Complexity performance of DHGN classification process (adopted from (Raja Mahmood et al., 2008))

In summary, we have shown in this chapter that our proposed scheme follows the requirements for effective classification scheme to be deployed over lightweight networks such as WSN. DHGN adopts a single-cycle learning approach with non-iterative procedures. Furthermore, our scheme implements an adjacency comparison approach, rather than iterative weight adjustment approach using Hebbian learning that has been adopted by numerous neural network classification schemes. In addition, DHGN performs recognition and classification processes with minimum memory utilisation, based upon the store/recall approach in pattern recognition.

5. Case Study: Forest Fire Detection using DHGN-WSN

In recent years, forest fire has become a phenomenon that largely affects both human and the environment. The damages incurred by this event cost millions of dollars in recovery. Current preventive measures seem to be limited, in terms of its capability and thus require active detection mechanism to provide early warnings for the occurrence of forest fire. In this chapter, we present a preliminary study on the adoption of DHGN distributed pattern recognition scheme for forest fire detection using WSN.

5.1 Existing Approaches

There are a number of distinct approaches that have been used in forest fire detection. These include the use of lookout towers using special devices such as Osborne fire finder (Fleming and Robertson, 2003) and video surveillance systems such as in the works of (Breejen et al., 1998).

There are also a few works on forest fire detection using WSN, including the works of (Hefeeda and Bagheri, 2007) in the implementation of forest fire detection using Fire Weather Index (FWI) and Fine Fuel Moisture Code (FFMC). In addition, the works of (Sahin, 2007) in forest fire detection suggested the use of animals as mobile biological sensors, which are equipped with wireless sensor nodes. On the other hand, (Zhang et al., 2008) proposed the use of ZigBee technique in WSN for forest fire detection.

Our main interest is in the works conducted by (Hefeeda and Bagheri, 2007) using FWI and FFMC for standard measurement for forest fire detection. FWI and FFMC have been introduced by Canadian Forest Service (CFS) and (De Groot et al., 2005). FWI is used to describe the spread and intensity of fires, while FFMC is used as a primary indicator for a potential forest fire. At this stage, our interest is mainly focuses on early detection for potential forest fire. Hence, our works basically concentrate on the use of FFMC values for fire detection.

5.2 Dimensionality Reduction on FFMC Values

The detection scheme proposed by (Hefeeda and Bagheri, 2007) involves centralised process of obtaining the FFMC values from the sensory readings. The readings obtained from the sensor nodes would be transmitted to the sink for FFMC value determination. FFMC value is derived from an extensive calculation involving environmental parameter values including temperature, relative humidity, precipitation, and wind speed. Our approach using DHGN recognition scheme is focusing on reducing the burden experienced by the back-end processing within the sink, by providing a front-end detection scheme that enables only valid (event-detected) readings that will be sent for further processing.

Table 4 shows the FFMC value versus ignition potential level. This FFMC value provides an indication of relative ease of ignition and flammability of fine fuels due to exposure to extreme heat. In general, fires usually begin to ignite at FFMC values around 70, and the highest probable value to be reached is 96 (De Groot et al., 2005).

Our DHGN implementation performs dimensionality reduction on the FFMC values, by combining the existing five ignition potential levels for ignition potential into two stages: High Risk and Low Risk, as shown in Table 5. Using this approach, we could determine the possibility of forest fire occurring, given certain values of sensory readings.

Ignition Potential	FFMC Value
Low	0 – 76
Moderate	77 – 84
High	85 – 88
Very High	89 – 91
Extreme	92+

Table 4. Ignition potential versus FFMC value

Ignition Potential	VVMC Value
Low Risk	0-84
High Risk	85+

Table 5. Modified FFMC classification for DHGN event detection scheme

5.3 Methodology

The implementation of DHGN for forest fire detection involves a series of steps that reduces the expensive computation of FFMC values at the base station. We have proposed a distributed detection scheme that enables each sensor node to perform a simple recognition process using DHGN to detect any abnormal readings obtained from its surroundings.

The first processing step in our recognition scheme is to reduce the sensory data dimension using adaptive threshold binary signature approach. In this approach, we assume that each sensor node is composed of multiple sensors including temperature, relative humidity, precipitation, and wind speed. The readings would be converted into binary string representation, using the conversion methods as discussed in Section 4.2.

The second step is the actual recognition process, in which the binary signature is treated as subpattern and being introduced into specific DHGN processing cluster within each of the sensor nodes. We assume that DHGN processing cluster in this context is taken place as a block of memory space that could be used for simple DHGN recognition process. In addition, we assume that each node is handling a subpattern (sensory readings) which collectively could become an overall pattern for the whole sensor nodes within the network. The recognition process is conducted by using reference patterns which consist of normal event subpattern/readings.

Once the sensor node detected an abnormal occurrence of subpattern (subpattern is not being recalled), it will send a signal to the base station for further analysis. This signal consists of all the sensory readings and event flag. The base station then compute the FFMC value of the readings. Continuous signals being sent to the base station could be interpreted as a high potential risk of forest fire and vice versa. Therefore, early process of prevention could be executed at a specific location within the area of the sensor nodes.

We have conducted a preliminary test on the accuracy of our scheme and a comparison with Kohonen’s self-organizing map (SOM). We have taken a forest fire data from (Cortez and Morais, 2007) and performed our DHGN simulation on computational grid environment for this dataset with 517 items. We have taken three distinctive readings from the dataset, which include temperature, relative humidity and wind speed. We have ignored the precipitation (rainfall) values for this dataset as it has shown minimal effect to the FFMC values. Table 6 shows the bits allocation for each of the readings. This bits allocation eventually will be represented as a binary signature. The results of this test are presented in the following subsection.

Data	Bit Allocation
Temperature	2 bits
Relative Humidity	3 bits
Wind Speed	2 bits

Table 6. Sensory data with allocated binary signature bits

5.4 Classification/Recognition Results

In this study, we have performed a supervised classification test using DHGN event detection scheme. We then compared our results with Kohonen's self-organizing map (SOM) classifier. We have used the SOM toolbox for Matlab that has been developed by (Vesanto et al., 2000). The results from this test have shown that our approach non only produces equivalent recall accuracy in comparison to the SOM classifier but also requires minimum training and training data.

The training data used in the experiments only signifies the normal event data (FFMC values lower than 84). With similar number of training data used, DHGN produces higher classification accuracy as compared to Kohonen SOM. Table 7 shows the training data that have been used in this classification test. The classification accuracy obtained using DHGN reaches up to 88.78%, while SOM is only able to achieve accuracy percentage of 5.61%. Fig. 14 shows the results of this classification test.

Data	1	2	3
FFMC Value	≤ 84	≤ 84	≤ 84
Temperature (ºC)	0-40 (10)	0-40 (10)	0-40 (10)
Relative Humidity (%)	> 70 (001)	≤ 40 (100)	> 70 (001)
Wind Speed (km/h)	≤ 3 (10)	≤ 3 (10)	> 3 (01)
Binary Signature	1000110	1010010	1000101

Table 7. Training data set in the form of specific threshold ranges used in classification test. Binary digits in brackets represent signature for the respective data range

We then extend our test on Kohonen SOM to observe the effect of the increased training data on its classification accuracy. We added abnormal event data (FFMC values higher than 84) in the training set. Fig. 15 presents the results of this extended test. Note that an increase in the number of training data improves the accuracy of SOM classifier.

The test reveals that DHGN offers higher accuracy with minimum training data, in comparison to SOM. Furthermore, our distributed approach requires no training iteration, as it adopts a single-cycle learning mechanism. Comparatively, SOM requires high training iteration in order to achieve high classification accuracy. Fig. 16 shows the number of iterations incurred for different number of training data being used in our test.

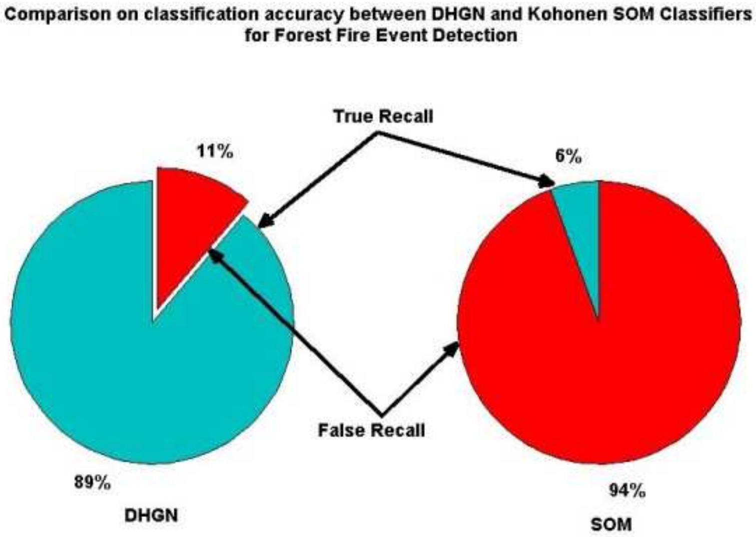


Fig. 14. Classification accuracy comparison between DHGN and Kohonen SOM classifiers for forest fire detection. We have used small number of training data (3 patterns) for each algorithm

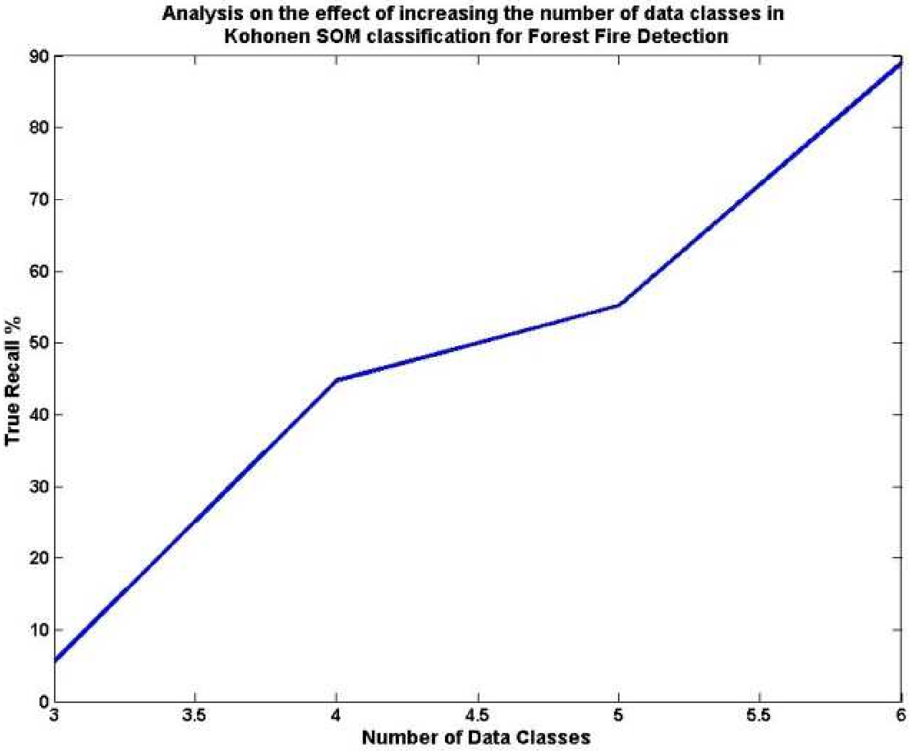


Fig. 15. Analyses of the effect of increasing number of training data set in Kohonen SOM for forest fire data Classification

The results of this test have also convinced us that our proposed scheme is capable of providing high classification accuracy while requiring minimal training effort. Thus, makes it highly suitable to be deployed over resource-constrained networks such as WSN.

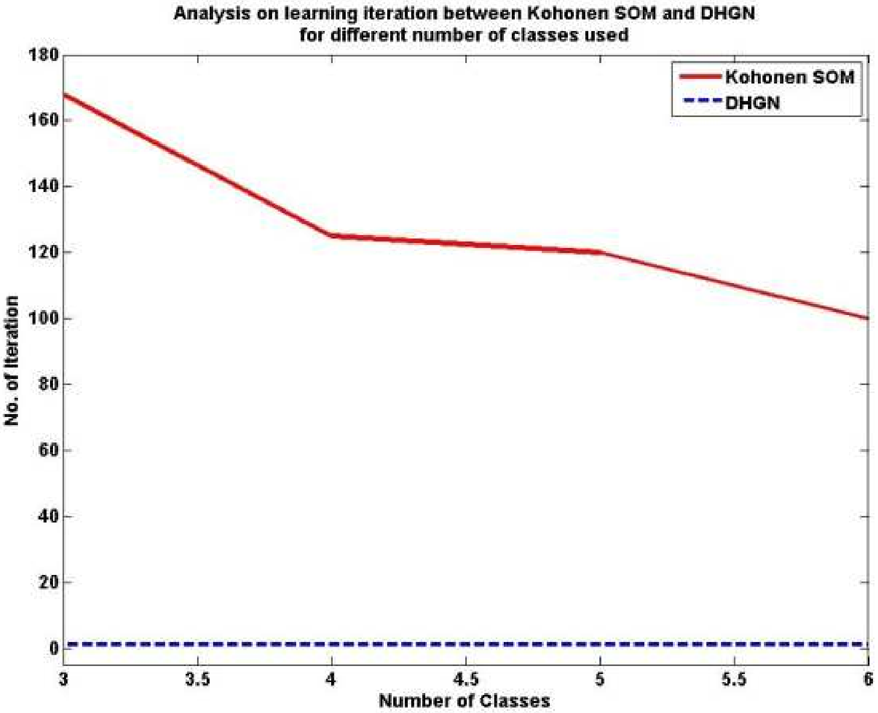


Fig. 16. Analysis of learning iteration between Kohonen SOM and DHGN for different number of classes used in training

5.5 Summary

We have presented our preliminary study on forest fire detection using DHGN distributed pattern recognition algorithm within WSN network. Our proposed implementation involves minimum modification towards existing WSN infrastructure. Furthermore, based on the classification test results, DHGN has shown to perform well with minimum training data and within a single-cycle learning mechanism. This makes our proposed approach more viable for WSN deployment in forest fire detection.

6. Discussion and Future Research

There are several benefits and advantages in our DHGN implementation for event detection within WSN network. Our approach offers low memory consumption for event data storage using simple bias array representation. Furthermore, this scheme only stores subpatterns/patterns that are related to normal event, rather than keeping the records of all occurring events. We have also shown that our approach is most effective for small subpattern size, since it uses only a small portion of the memory space in a typical physical sensor node in WSN network.

In addition to this efficient memory usage, DHGN also eliminates the need for complex computations for event classification technique. With the adoption of single-cycle learning and adjacency comparison approaches, DHGN implements a non-iterative and lightweight computational mechanism for event recognition and classification. The results of our performance analysis have also shown that DHGN recognition time increases linearly with an increase in the number of processing elements (PEs) within the network. This simply reveals that DHGN's computational complexity is also scalable with an increase in the size of the subpatterns.

DHGN is a distributed pattern recognition algorithm. By having this distributed characteristic, DHGN would be readily-deployable over a distributed network. With such feature, DHGN has the ability to perform as a front-end detection scheme for event detection within WSN. Through divide-and-distribute approach, complex events could be perceived as a composition of events occurring at specific time and location. Our approach eventually would be able to be used in event tracking. However, the discussion on event tracking is not within the scope of this chapter. Nevertheless, our proposed scheme has been demonstrated to perform efficiently within an event detection scheme such as forest fire detection using WSN.

Despite all its benefits, DHGN has its own limitations. Firstly, DHGN simple data representation would requires significant advanced pre-processing at the front-end of the system. This might not be viable for strictly-resource constrained sensor nodes, where processing capability is very limited. In addition, DHGN single-hop communication for event detection scheme is not viable for large area monitoring, due to high possibility of communication error due to data packet loss during transmission. Our existing DHGN implementation has also been focusing on supervised classification. However, there is a need for unsupervised classification technique to be deployed for rapid event detection scheme.

Overcoming the DHGN distributed event detection scheme limitations would be the path of our future research direction. We intend to look into event tracking scheme using DHGN distributed detection mechanism, as well as providing unsupervised classification capability for rapid and robust event detection scheme. Furthermore, we are looking forward into implementation of this scheme in large-area monitoring using multi-hop communication strategy.

7. Conclusion

The development of event detection scheme within WSN has been made viable with the advancement in communication, computational, and sensor technologies. However, existing detection/recognition algorithms fail to achieve optimum performance in a distributed environment, due to its tightly-coupled and computationally intensive nature. In this chapter, we have presented our readily-distributable event detection scheme for WSN network which is known as Distributed Hierarchical Graph Neuron (DHGN). Throughout our studies, we discover that DHGN is able to perform recognition and classification processes with limited training data and within a one-shot learning. These DHGN features have given added-value for implementing this scheme within a lightweight distributed network such as WSN. In addition, our proposed adaptive threshold binary signature scheme has the ability to provide generalisation and simplification of datasets to be used in DHGN distributed pattern recognition scheme.

Current implementation of DHGN in event detection using WSN has been focusing on the front-end processing, in which detection could be carried out earlier using the available wireless sensor nodes. Our approach differs from other existing event detection schemes in which major processing steps are conducted at the base station. By having a front-end detection, our proposed scheme is able to alleviate the computational costs experienced by the centralised-processing undertaken by the base station.

In this chapter, we have also discussed the advantages and limitations of our proposed scheme. The future direction of this research lies in the development of a complete event detection scheme that incorporates front-end detection and back-end complex event analysis. We foresee our DHGN distributed pattern recognition scheme as a complete event detection and analysis tool that is deployable over different types of event detection schemes on WSN networks.

8. References

Ai, C., Hou, H., Li, Y. and Beyah, R. (2009) *Ad Hoc Networks,* **7,** 599-613.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) In *IEEE Communications*, Vol. 40 IEEE, pp. 102-114.

Banerjee, T., Xie, B. and Agrawal, D. P. (2008) *Journal of Parallel and Distributed Computing,* **68,** 1222-1234.

Benyuan, L., Peter, B., Olivier, D., Philippe, N. and Don, T. (2005) In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*ACM, Urbana-Champaign, IL, USA.

Breejen, E. d., Breuers, M., Cremer, F., Kemp, R., Roos, M., Schutte, K. and Vries, J. S. d. (1998) In *The Third International Conference on Forest Fire Research*, Vol. 2 Luso, Portugal, pp. 2003-2012.

Cortez, P. and Morais, A. (2007) In *Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence*(Eds, Neves, J., Santos, M. F. and Machado, J.) APPIA, Guimarães, Portugal, pp. 512-523.

Culler, D., Estrin, D. and Srivastava, M. (2004) In *Computer*, Vol. 37, pp. 41-49.

De Groot, W. J., Wardati and Wang, Y. (2005) *International Journal of Wildland Fire,* **14,** 161–168.

Elaine, C., Kristof Van, L. and Martin, S. (2003) In *Proceedings of the eighth international conference on Artificial life*MIT Press.

Fleming, J. and Robertson, R. G. (2003) In *Fire Management Tech Tips* SDTDC USDA Forest Service, San Dimas CA, USA.

Guralnik, V. and Srivastava, J. (1999) In *The Fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99)*ACM, San Diego, CA, USA, pp. 33-42.

Hafez, R., Haroun, I. and Lambaridis, I. (2005) In *Systems & Subsystems*, Vol. 2008 Penton Media, Inc.

Hefeeda, M. and Bagheri, M. (2007) In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, 2007 (MASS 2007)* IEEE Press, Pisa, Italy, pp. 1-6.

Hopfield, J. J. and Tank, D. W. (1985) *Biological Cybernetics,* **52,** 141-152.

Khan, A. I. (2002) In *The Proceedings of The Thirteenth Australasian Conference on Information Systems*Melbourne, Australia.

Khan, A. I. and Muhamad Amin, A. H. (2007) In *AI 2007: Advances in Artificial Intelligence, 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, December 2-6, 2007, Proceedings*, Vol. 4830 (Eds, Orgun, M. A. and Thornton, J.) Springer, pp. 705-709.

Kohonen, T. (2001) *Self-Organizing Maps,* Springer.

Kulakov, A. and Davcev, D. (2005a) In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)*IEEE Computer Society, Cartagena, Spain.

Kulakov, A. and Davcev, D. (2005b) In *Proceedings of International Conference on Information Technology: Coding and Computing, ITCC 2005*Las Vegas, NV, USA, pp. 534-539.

Levis, P. and Culler, D. (2002) In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)*ACM, San Jose, CA, USA.

Lewis, F. (2004) In *Smart Environments: Technology, Protocols and Applications*(Eds, Cook, D. and Das, S.) Wiley.

Li, S., Lin, Y., Son, S. H., Stankovic, J. A. and Wei, Y. (2004) *Telecommunication Systems,* **26,** 351-368.

Li, Y. and Parker, L. E. (2008) In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008 (IROS 2008)*, vol., no., pp.3292-3298, 22-26 Sept. 2008 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4651031&isnumber=4650570, pp. 3292-3298

Muhamad Amin, A. H. and Khan, A. I. (2008a) In *Sixth Australasian Symposium on Grid Computing and e-Research (AusGrid 2008)*, Vol. 82 (Eds, Kelly, W. and Roe, P.) ACS, Wollongong, NSW, Australia, pp. 27-34.

Muhamad Amin, A. H. and Khan, A. I. (2008b) In *Proceedings of the 2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies 2008 (PDCAT'08)*IEEE Computer Society, Dunedin, New Zealand, pp. 305-308.

Muhamad Amin, A. H., Raja Mahmood, R. A. and Khan, A. I. (2008) In *Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on*, pp. 153-158.

Munir, S. A., Ren, B., Jiao, W., Wang, B., Xie, D. and Ma, J. (2007) In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*IEEE Computer Society.

Nascimento, M. A. and Chitkara, V. (2002) In *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'02)*ACM, Madrid, Spain, pp. 687-692.

Nasution, B. B. and Khan, A. I. (2008) *IEEE Transactions on Neural Networks,* **19,** 212-229.

Park, S., Locher, I. and Srivastava, M. (2002) In *6th International Symposium on Wearable Computers (ISWC2002)*Seattle, WA, USA.

Raja Mahmood, R. A., Muhamad Amin, A. H. and Khan, A. I. (2008) In *Intelligent Networks and Intelligent Systems, 2008. ICINIS '08. First International Workshop on*Wuhan, China, pp. 675-680.

Saha, S. and Bajcsy, P. (2003) In *The IASTED International Conference on Communications, Internet and Information Technology 2003 (CIIT 2003)*(Ed, Hamza, M. H.) ACTA Press, Scottsdale, AZ, USA.

Sahin, Y. G. (2007) *Sensors,* **7,** 3084-3099.

Shih, K.-P., Wang, S.-S., Chen, H.-C. and Yang, P.-H. (2008) *Computer Communications,* **31,** 3124-3136.

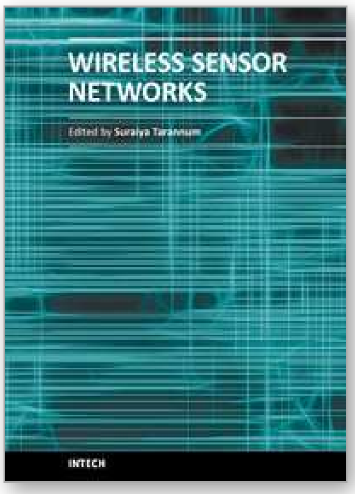
Shih, K.-P., Wang, S.-S., Yang, P.-H. and Chang, C.-C. (2006) In *IEEE Symposium on Computers and Communications (ISCC)*, pp. 935-940.

Stankovic, J. A. (2008) In *Computer*, Vol. 41 IEEE Computer Society, New York, NY, pp. 92-95.

Vesanto, J., Himberg, J., Alhoniemi, E. and Parhankangas, J. (2000) In *Proceedings of the Matlab DSP Conference*, pp. 35-40.

Wang, W., Srinivasan, V. and Chua, K.-C. (2008) *IEEE/ACM Transactions on Networking,* **16,** 1108-1120.

Zhang, J., Li, W., Han, N. and Kan, J. (2008) *Frontiers of Forestry in China,* **3,** 369-374.



Wireless Sensor Networks
Edited by

ISBN 978-953-307-325-5
Hard cover, 342 pages
Publisher InTech
Published online 29, June, 2011
Published in print edition June, 2011

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821