International Journal of u- and e- Service, Science and Technology
Vol.8, No.4 (2015), pp.265-278
http://dx.doi.org/10.14257/ijunesst.2015.8.4.24

# An Improved Spatial-based Fuzzy Logic Event Detecting Algorithm for Wireless Sensor Networks

Baowei Wang[1,2], Aoran Zhuang[1], Huiyu Sun[3], Tao Li[1,2] and Xingming Sun[1,2]

*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China*
*Jiangsu Engineering Center of Network Monitoring, Nanjing 210044, Nanjing Wisdom Technology Ltd, Nanjing 211800, China*
*wbw.first@163.com, hangsomelaozhuang@163.com, huiyu.sun@yahoo.co.uk, taoleehome@hotmail.com, sunnudt@163.com*

***Abstract***

*At present, event detection technologies have become an important part in building efficient wireless sensor networks. One of the popular and excellent event detecting algorithms is the improved Neighbor-based Fuzzy Logic algorithm, namely INFLE, which belongs to machine learning technology. However, there may be some false positives when fire has not spread to the detected node. In this paper, we propose an improved spatial-based fuzzy logic event detection algorithm (ISFLE) to dramatically decrease the probability of false positives before fire has spread to the node. In our proposed ISFLE, we put "Distance" variable to the fuzzy logic system to determine node's final state. The simulation results validate that our proposed ISFLE outperforms traditional INFLE in decreasing the number of false positives.*

***Keyword:*** *wireless sensor networks, event detection, fuzzy logic, spatial correlation*

## 1. Introduction

A wireless sensor network is composed by tens to thousands of wireless sensor nodes which can monitor the area and they are used in many applications such as security, surveillance, climatic-change studies, and structural health monitoring. In some poor detecting environment, abnormal events often happen in uncertain time, so we need sensor nodes to set an alarm precisely and punctually if abnormal events happen. Due to the limited computational power, memory, communication range for sensor nodes, designing an ideal event detection algorithm needs to be energy efficient, fault tolerant and robust, resource friendly, and adaptive to multiple event types and environments and it must be accuracy and produces less false alarm [1].

One of the most widely used event detection is to use a threshold value, if the value of sensor node exceeds the pre-defined threshold, an event is considered to happen. Threshold-based event detection is easy to implement but hard to set an appropriate threshold for some events and may produce many false alarms. Model-based event detection techniques are other detecting techniques that model events phenomena in a specific form such as mathematic formulas or maps. It can handle more complex events than threshold-based, as the formal represent a non-linear model for different events. However, building event model needs professional knowledge and is always computationally complex. Pattern matching-based event detection detect event by comparing the sensor node's data with event signatures. It usually can handle complex events and is flexible to various applications, but it requires expert knowledge to set the right parameters [2].

Fuzzy logic is a kind of machine learning which is the most promising in event detecting. Compared to other event detection algorithms, fuzzy logic has some advantages: a) it can tolerate unreliable and imprecise sensor readings; b) it is much closer to our way of thinking.

ISSN: 2005-4246 IJUNESST
Copyright ⓒ 2015 SERSC

For example, we think of fire as an event described by high temperature and smoke rather than an event characterized by temperature above 55℃ and smoke obscuration level above 15%; c) models are far less complicated than mode-based event detection; d) compared to other classification algorithms based on probability theory, fuzzy logic is more intuitive and easier to use [3]. EDA (Event-oriented Data Aggregation) is a distributed fuzzy-based event detection approach that uses fuzzy engines to detect events [4]. This approach was implemented on TelosB sensor nodes in an offshore test bed for ocean surveillance application. Liang and Wang [3] propose to use fuzzy logic in combination with double sliding window detection, to improve the accuracy of event detection. However, they do not study the effect of fuzzy logic alone or the influence of spatial or temporal properties of the data on the detection accuracy. In D-FLER [5] fuzzy logic is used to combine personal and neighbors' observations and determine if an event has occurred. Their results show that fuzzy logic improves the accuracy of event detection. The use of fuzzy values allows D-FLER to distinguish between real fire data and fake fire data. However, the author does not analyze accuracy of D-FLER.

However, there still exist a lot of problems for event detecting using fuzzy logic. In [7], the author proposes an INFLE to decrease the false negatives in NFLE [6]. The experiment is simulated in Matlab which contain 100 sensor nodes whose serial numbers are from 1 to 100. They are randomly deployed in a circle whose radius is 500m. We select node 100 in Figure1 to compare two algorithms for detecting fire on: A) NFLE: Neighbor-based Fuzzy Logic for event detecting(NFLE) which involves all neighbor readings in fuzzy logic system; B) INFLE: improved neighbor-based fuzzy logic which selects appropriate neighbor readings instead of selecting all neighbor readings in fuzzy logic system. Algorithm B is proved to produce less false negatives than algorithm A after fire ignition. However, at 111s before fire is ignited in the node, there is a sudden rise for INFLE as circled in Figure1. This may lead to some false positives in INFLE when fire confidence threshold is set not high enough. To solve the problem, ISFLE is proposed.

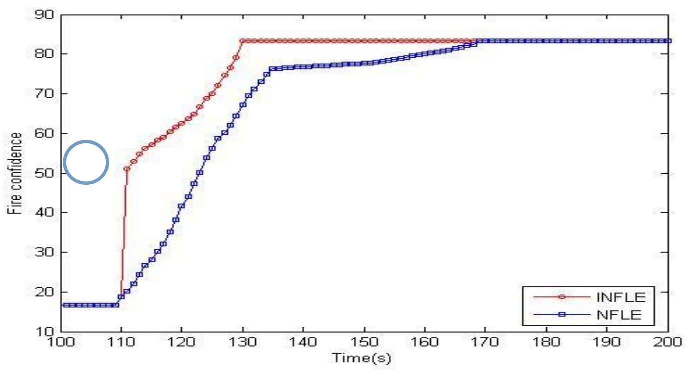



**Figure 1. Comparison between Algorithm A and B**

The paper is organized as follows: Section 2 introduces ISFLE including neighbor nodes selection algorithm and our fuzzy logic model for detecting fire. Section 3 mainly states a comparison result for detection accuracy between ISFLE and INFLE in a simulation. Some other methods for event detection have been concluded in Section 4. Section 5 concludes the paper.

## 2. Our Proposed Improved Spatial-based Fuzzy Logic Event Detection Scheme

When designing an event detection system, one of the main goals is that the system is accurate and the number of false alarms is small. A way to achieve this is to include readings from multiple sensors in the decision process. For instance, we would be more confident that

 Copyright ⓒ 2015 SERSC

there is an actual fire if more than one node which located nearer to each other reports high temperature and smoke readings which locate nearer to each other. The further the distance among the sensors reporting fire, the less likely the report being true. Therefore, we include the concept of location in the event detection logic.

For any node to be detected, it should observe if any of its neighbor node reports fire. If there is no neighbor node reporting fire only using the detected node, the node will judge its fire state by only using its own values in fuzzy logic regardless of neighbor readings [6]. Once the output fire confidence is 100% "high", the node will broadcast its state to all neighbors. Or else, the node and its neighbors will run NFLE algorithm.

If there is any node reporting fire to the detected node, the detected node and its neighbors will run our ISFLE algorithm. The five outputs for ISFLE which are Temperature1, Smoke1, Temperature2, Smoke2 and Distance will be added into fuzzy logic system. Then we can get a fire confidence for the node from fuzzy logic system. If fire confidence reaches the fire threshold, fire is confirmed on the node and the node raises a fire alarm and broadcast it to all its neighbors. Figure2 shows the flow chart for detecting fire on sensor nodes.

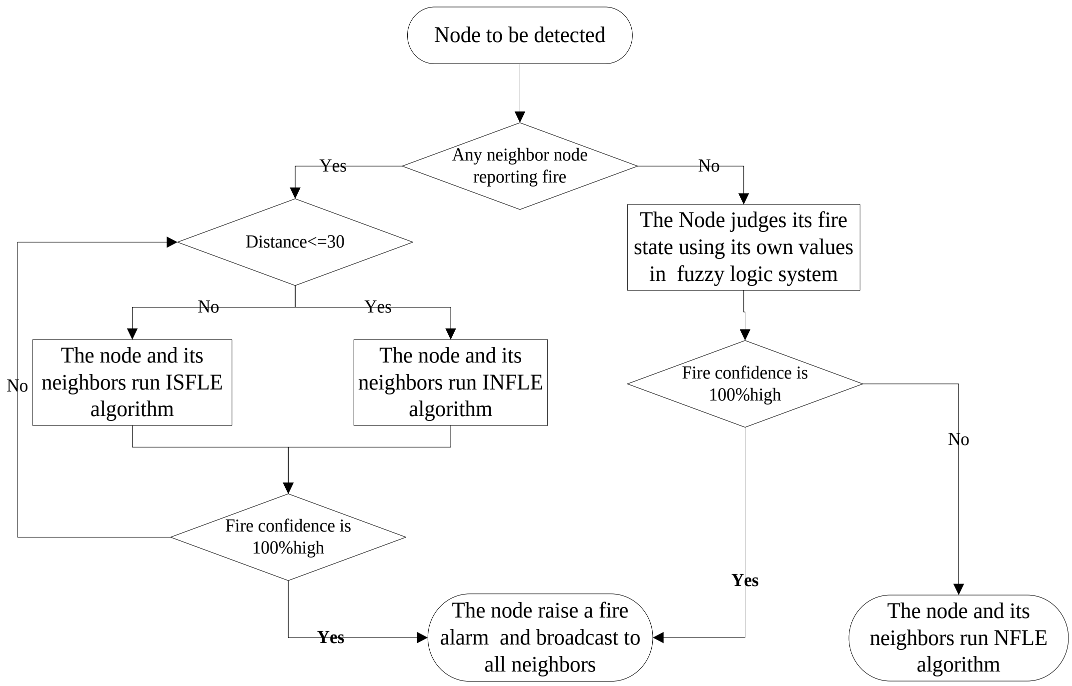



**Figure 2. Scheme for Detecting Fire on Sensor Nodes**

ISFLE algorithm consists of two parts: firstly, we select the readings of appropriate neighbor nodes by neighbor nodes selection algorithm. Secondly, we bring these readings into fuzzy logic model as input. The output of fuzzy logic system is fire confidence for the detected node.

### 2.1. Neighbor Nodes Selection Algorithm

In neighbor nodes selection algorithm, at first, node A will observe whether any of its neighbor nodes report fire. If so, A will keep these nodes coordinates and calculate the distances between itself and each of these nodes. Now suppose, amongst these nodes, E is the one located nearest to A. They form a circle C1 whose diameter is the distance from A to E. then we will choose nodes that are in C1 which are also neighbors to both A and E. Nodes in C1 whose coordinate also satisfy $D_2 <= D_{ME} <= D_{AE}$ is denoted C4 as shown by the shaded region in the figure. We choose the average readings of all the nodes that are contained in C4, in this example there are three, as output Temperature2 and Smoke2. Readings of A are used as Temperature1 and Smoke1, and the average distance between A and each of these nodes in C4 is our output Distance.

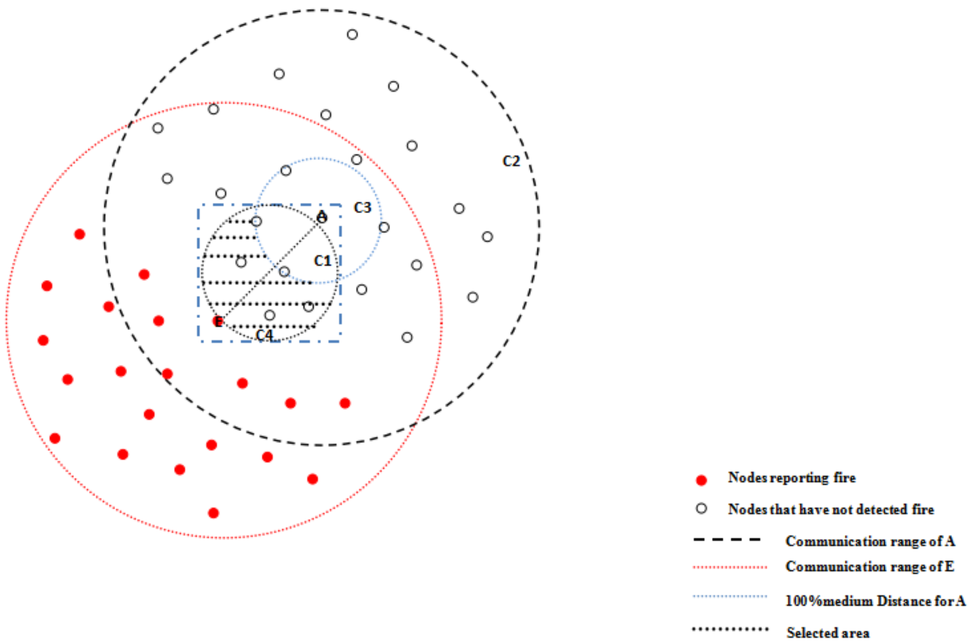



**Figure 3. Detecting Fire on Node a Using Neighbor Nodes Selection Algorithm**

**Definition 1:** Coordinate of the detected node A is $(x_1, y_1)$, E is nearest to A which report fire, its coordinate $(x_i, y_i)$.

**Definition 2:** Neighbors of A are A'=[$A_1 \ldots A_n$], neighbors of A that report fire are $A'_1$=[$A_1 \ldots A_o$](o<=n), neighbors of E are E'=[$E_1 \ldots E_k$], neighbors of both A' and E' are Q=[$Q_1 \ldots Q_q$](q<=n && q<=k), F=[$F_1 \ldots F_l$]($l$<=q).

**Definition 3:** Distance between any two nodes H and L whose coordinates are respectively $(X_H, Y_H)(X_L, Y_L)$ is $D_{HL} = \sqrt{(X_H - X_L)^2 + (Y_H - Y_L)^2}$.

**Definition 4:** Temperature(A) and Smoke(A) is defined as Temperature1 and Smoke1 respectively in our fuzzy logic system, while mean readings of selected neighbors are defined as Temperature2 and Smoke2.
**Definition 5:** Distance of "100%distant" for node A is $D_2$.

| **Neighbor nodes selection algorithm** |
|---|
| **Input:** Node A to be detected and all neighbors A'.<br>**Output:** Readings of A- Temperature1, Smoke1 and its selected neighbors-Temperature2, Smoke2, and Distance- average distance between selected nodes and A<br>1: Every node broadcasts its coordinate on the network and keeps its neighbors' coordinate.<br>2: Any time t, when we want to detect fire on A after ignition.<br>3: **while** any node in A' reports fire **do**<br>4: A calculates and store the distance between A and this node.<br>5: A find the node which locates nearest to A. |

 Copyright ⓒ 2015 SERSC

Suppose the node is E whose coordinate is $(x_i, y_i)$.
Count=0. SumTemperature=0. SumSmoke=0.Distance=0.
**For** each node G($x_g$,$y_g$) that have same coordinate in Q **do**.
**If** $x_i - \frac{|x_A - x_i|}{2} - \frac{D_{AE}}{2} <= x_g <= x_i + \frac{|x_A - x_i|}{2} + \frac{D_{AE}}{2}$ && $y_i - \frac{|y_A - y_i|}{2} - \frac{D_{AE}}{2} <= y_g <= y_i + \frac{|y_A - y_i|}{2} + \frac{D_{AE}}{2}$ **then**;
put G into F.
**end if**
**For** each node M in F whose coordinate is $(x_m, y_m)$ **do**
**If** $D_{MA}^2 + D_{ME}^2 <= D_{AE}^2$ **then**;
**If** $D_2 <= D_{ME} <= D_{AE}$ **then**;
Count++. SumTemperature= SumTemperature + Temperature(M).
SumSmoke= SumSmoke + Smoke(M). Distance=Distance+Distance(MA).
**end if**
**end for**
**end for**
**end while**
Temperature1= Temperature(A). Smoke1= Smoke(A).
Temperature2= SumTemperature/Count. Smoke2= SumSmoke/Count.
Distance=Distance/Count.
**Return** Temperature1, Smoke1, Temperature2, Smoke2, Distance.

## 2.2. Fuzzy Logic Model

A fuzzy logic model consists of four steps. Firstly, the fuzzifier converts the crisp input variables including Temperature1, Smoke1, Temperature2, and Smoke2 into fuzzy linguistic variables. Secondly we can get a series of output in percentage for some rules by applying T fuzzy operator. Thirdly, the output in percentage can be converted into a combined fuzzy output by fuzzy implication. Finally, a crisp output will be got by defuzzifier. Figure4 shows the process of our fuzzy logic system.

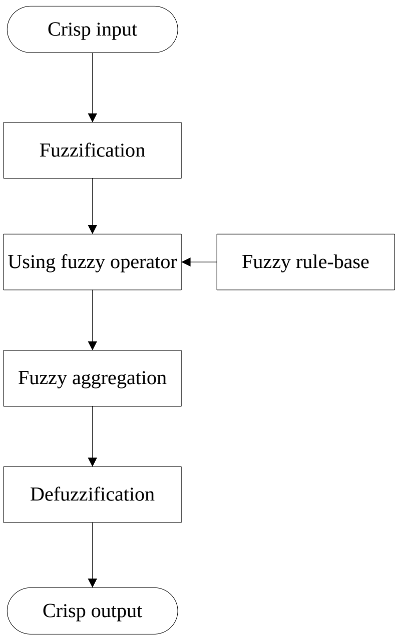



**Figure 4. The Structure of a Fuzzy Logic System**

**2.2.1 Fuzzification:** This step is to convert a crisp value into degrees of membership by applying the corresponding membership functions. Membership functions are defined by either relying on domain knowledge or through the application of different learning techniques. They determine the certainty with which crisp values are associated with specific linguistic values. These specific linguistic values are called antecedents of rule-base. Some shapes of membership function include triangular, trapezoidal and Gaussian-shaped. Triangular shapes and trapezoid are the most widely used in WSNs.

**2.2.2. Using fuzzy operator:** After the fuzzification step, we get a number of specific linguistic values. There are a lot of rules for a fuzzy logic application and the domains are set by experts. If one rule has more than one antecedent, for example, a rule is composed of t antecedents and 1 output:

[illegible]

When input $x' = \{x'_1, x'_2, \ldots, x'_t\}$, the degree of firing of the rule can be computed as:

$$f(x') = f_{S_1}(x'_1) * f_{S_2}(x'_2) * \cdots * f_{S_t}(x'_t) = T_{i=1}^{t} f_{S_i}(x'_i)$$

Here f represents the membership function and both * and T indicate the chosen triangular norm. This step will produce a series of output that is in the form of percentage for some rules [12]. We augment the rules in the rule-base with a linguistic variable named "Distance" that serves as a spatial guard. This variable expresses the application requirements about the distance between the reporting sensors.

**Table 1. Format of our Fuzzy Rule**

| Rule | Temperature1 | Smoke1 | Temperature2 | Smoke2 | Distance | Confidence |
|---|---|---|---|---|---|---|

Table1 describes the form of our fuzzy rule, which have five inputs called Temperature1, Smoke1, Temperature2, Smoke2, Distance and one output called Confidence. Suppose every node is equipped with temperature, smoke and GPS sensor. Temperature1 and Smoke1 are the readings of the node that is to be detected, while Temperature2 and Smoke2 represent the mean values of selected neighbor nodes' readings and Distance means the average distance between the selected nodes and the detected nodes. The higher output Confidence is, the higher possibility of fire happened in the detected node.

**2.2.3. Fuzzy Aggregation:** The output by the second step can be converted into a combined fuzzy output by fuzzy implication. "Fuzzy and" is used in our fuzzy compositional operation [8].

**2.2.4. Defuzzification:** The final step is to convert the comprehensive output fuzzy value to a crisp output value. Defuzzification is the transformation of this set of percentages into a single crisp value. Centroid methods are the most used in WSNs which are used in our fuzzy logic system. For instance, in our algorithm, if we get 50% "medium" and 50% "high" for fire confidence, the output fire confidence will be 60 after this step. In our fuzzy logic system, the crisp value is fire confidence which is used to describe the probability of fire happened to the node. If fire confidence for one node is 83.33 which is 100% "high", fire is confirmed on the node.

## 3. Simulation and Evaluation

### 3.1. Experiment Environment

We design a simulation in Matlab as Figure5 shows to compare ISFLE with INFLE for fire detecting. There are 100 nodes which have ids from 0 to 100 deployed randomly in a circle

 Copyright ⓒ 2015 SERSC

whose radius is 500m. The sensor radio range is set to 100m. Suppose at time 0 fire is ignited from center of the circle, and it spread outwards in all directions at the same speed of 2m/s. Node's sample rate is set to be 1Hz.

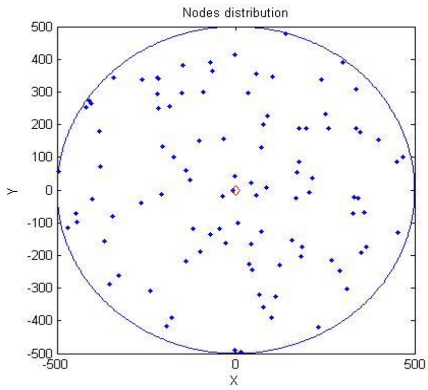
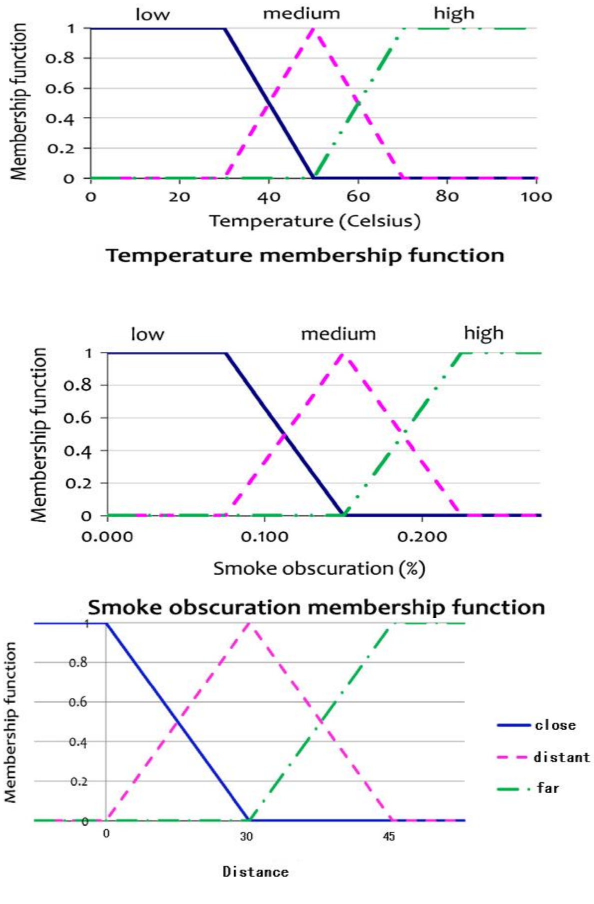



**Figure 5. Experiment Environment**

## 3.2. Our Fuzzy Logic System

The fuzzy logic system for INFLE and ISFLE is displayed through the FuzzyJ Toolkit for Java [9]. Temperature, Smoke membership functions, Distance membership functions [10] and output fire confidence membership functions are showed in Figure6.



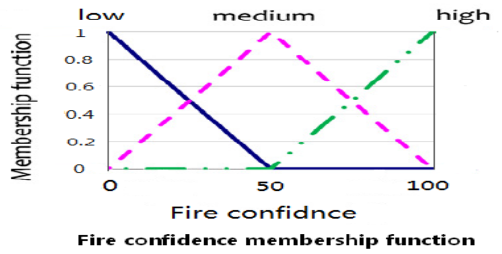



**Figure 6. Membership Functions for Input and Output in our Fuzzy Logic System**

Furthermore, we define parts of the rules for our fuzzy logic system which are using parts of NFLE and by specialized knowledge which is showed in Table1. One rule consists of five inputs and one output.

**Table 2. Rules for our Fuzzy Logic System**

| Rule | Temperature1 | Smoke1 | Temperature2 | Smoke2 | Distance | Confidence |
|---|---|---|---|---|---|---|
| 1 | low | low | <=high | <=medium | <=far | low |
| 2 | low | low | medium | high | >=distant | low |
| 3 | low | low | >=medium | high | <=distant | medium |
| 4 | low | low | high | high | far | low |
| 5 | high | high | medium | low | <=distant | medium |
| 6 | high | high | medium | low | far | high |
| 7 | high | high | low | <=high | <=Distant | medium |
| 8 | high | high | low | <=high | far | high |
| 9 | high | high | low | low | <=distant | medium |
| 10 | high | high | medium | low | far | high |
| 11 | high | high | low | <=high | <=far | medium |
| 12 | high | high | low | <=high | far | high |
| 13 | high | high | medium | >=medium | <=far | high |
| 14 | high | high | high | <=high | <=far | high |
| 15 | medium | low | low | <=high | >=distant | low |
| 16 | medium | low | medium | low | >=distant | low |
| 17 | medium | low | >=medium | >=medium | <=distant | medium |
| 18 | medium | low | >=medium | >=medium | far | low |
| 19 | medium | low | high | low | <=distant | medium |
| 20 | medium | low | high | low | far | low |
| 21 | medium | low | high | low | <=distant | medium |
| 22 | medium | medium | >=low | >low | <=far | medium |
| 23 | medium | medium | low | low | far | medium |
| 24 | medium | medium | low | low | <=distant | low |
| 25 | medium | medium | >=medium | low | >=close | medium |
| 26 | medium | high | high | <=medium | close | high |
| 27 | medium | high | high | <=medium | >=distant | medium |
| 28 | medium | high | <=medium | <=medium | >=close | high |
| 29 | medium | high | <=medium | high | <=far | medium |
| 30 | medium | high | high | high | close | high |
| 31 | medium | high | high | high | >=distant | medium |
| 32 | low | medium | <=medium | <=medium | <=far | low |
| 33 | low | medium | low | high | close | medium |
| 34 | low | medium | low | high | >=close | low |
| 35 | low | medium | high | low | close | medium |
| 36 | low | medium | high | low | >=distant | low |
| 37 | low | medium | high | >medium | <=distant | medium |
| 38 | low | medium | high | >medium | far | low |
| 39 | low | medium | medium | high | <=distant | medium |
| 40 | low | medium | medium | high | far | low |
| 41 | low | high | low | <=medium | distant | low |
| 42 | low | high | low | <=medium | >=distant | medium |
| 43 | low | high | low | high | <=far | medium |
| 44 | low | high | >=medium | >low | >=distant | medium |
| 45 | low | high | medium | low | distant | low |
| 46 | low | high | medium | low | >=distant | low |
| 47 | low | high | high | low | >=distant | medium |
| 48 | high | medium | >=low | >=low | >=distant | high |
| 49 | high | medium | <=medium | <=medium | distant | medium |
| 50 | high | medium | high | <=medium | close | high |
| 51 | high | medium | <=medium | high | close | high |


Copyright ⓒ 2015 SERSC

When fire is not ignited, we set temperature to 25℃ which is 100%low and smoke is set to 0.07% which is also 100%low. Considering that smoke spreads faster than fire, when fire is less than 50m from one sensor node, we set a random smoke value between 0.07% and 0.15% which is “a bit low” and “a bit medium”. If fire is 20m to 10m from one node, temperature is set randomly between 25℃ and 55℃ which is “a bit low” and “a bit medium”, besides we set smoke to a range between 0.15% and 0.2% which is “a bit medium” and “a bit high”. When fire is less than 10m from the node, we set a random temperature value between 55℃ and 75℃ which is “a bit medium” and “a bit high”, the smoke is set between 0.2% and 0.23%. When fire has spread to the node, considering the measurement range for sensors, we set the temperature 75℃ and smoke 0.23% as “100%high”.

### 3.3. Simulation Results

At time 0 when fire is ignited, fire starts from the center of the circle. Figure7 presents NFLE, INFLE and ISFLE for detecting fires on node 100 whose coordinate is (122.5, -228.0). It has six neighbor nodes 85, 90, 91, 93, 95, 97 whose coordinates are (37.3, -225.5), (44.3, -245.2), (157.2, -154.6), (110.93,-326.3), (185.9,-174.9), (183.0,-202.61) separately.

At time 111s, we can see the red line from Figure 7 that node 91 reports fire and INFLE selects the readings of 91 as Temperature2 and Smoke2 which will lead to a sudden rise for fire confidence, this is not accurate for the reason that fire is far from node 100. However, in ISFLE, the distance between node 100 and 91 is 81m which is 100% far and the fire confidence for ISFLE in 111s is 20.6. Besides, when fire has spread to 100, ISFLE can also detect fire in a timely fashion.

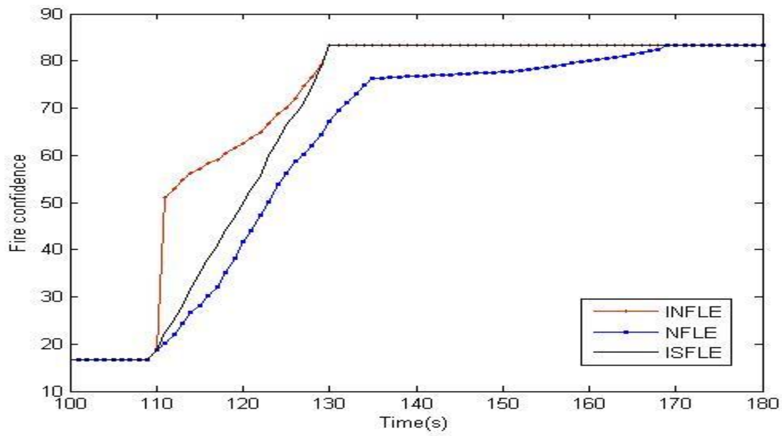



**Figure7. Comparison between ISFLE and INFLE on Detecting Fire on Node 100**

Figure 8 demonstrates the comparison between our INFLE and ISFLE on false positives when fire confidence threshold value is set to 55. We can see from the figure that ISFLE can produce few fault positives than that of INFLE. Maxium false positives of ISFLE is 6% which is much smaller than that of INFLE which reaches 22%.

Copyright ⓒ 2015 SERSC

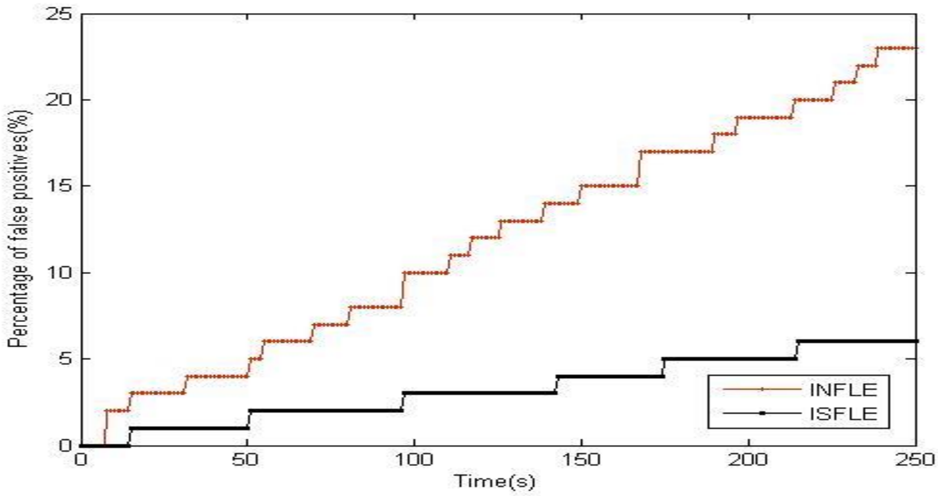



**Figure 8. Comparison between ISFLE and INFLE on Fault Positives for Detecting Fire on the Whole Networks**

Figure 9 shows the comparison between our INFLE and ISFLE on false positives when fire confidence threshold value is set to 65. We can also see from the figure that IDFLE can produce few false positives than that of INFLE. And ISFLE produces not more than 3 false positives all the time when fire is ignited. Maxium false positives of ISFLE is 5% which is much smaller than that of INFLE which reaches 19%.

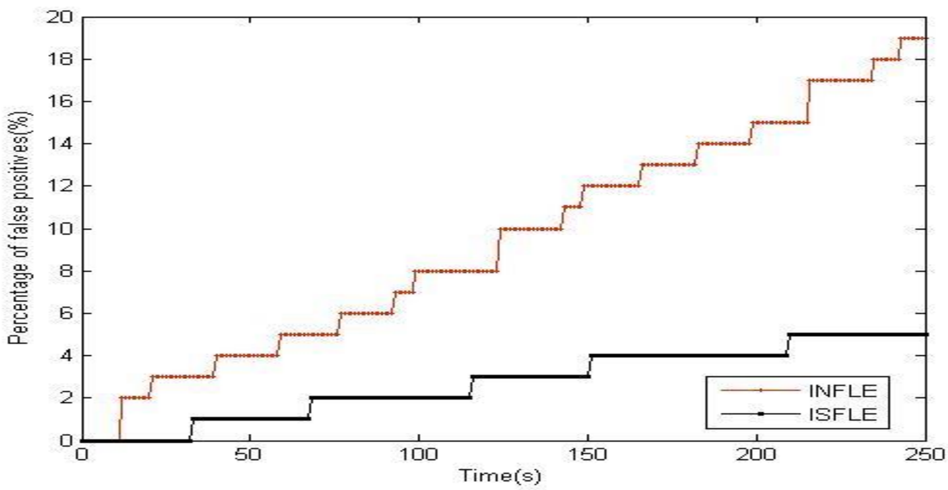



**Figure 9. Comparison between ISFLE and INFLE on Fault Positives for Detecting Fire on the Whole Networks**

### 3.4. Energy Costs and Time Complexity Analyses for INFLE and ISFLE

In neighbor nodes selection algorithm for ISFLE, the detected node needs to transmit and store more information that is the distance between the selected nodes and the detected node every sample period compared than INFLE. For one node that has n neighbors, time complexity of INFLE and ISFLE is $O_{(o*q*l)}$(o<=n, q<=n && q<=k, $l$<=q).

Nodes will need to store three times fuzzy rules in ISFLE compared with INFLE. When fuzzy linguistic values grow larger, space needed to store fuzzy rules will be an exponential growth. Nodes need to store 243 fuzzy rules in fuzzy logic system, which is three times for INFLE. However, one rule in ISFLE consists of only five inputs and one output, sensor nodes can afford the energy to store these rules. ISFLE needs 607.5 bytes to store fuzzy rules, while

 Copyright ⓒ 2015 SERSC

INFLE needs 172.125 bytes to store. However, a sensor node can afford the RAM to store the fuzzy rules, for instance, a telosb node have a 10KB RAM .

**Table 3. Space Needed for INFLE**

| Rule | Temperature1 | Smoke1 | Temperature2 | Smoke2 | Confidence |
|---|---|---|---|---|---|
| 7 | 2 | 2 | 2 | 2 | 2 |

**Table4. Space Needed for ISFLE**

| Rule | Temperature1 | Smoke1 | Temperature2 | Smoke2 | Distance | Confidence |
|---|---|---|---|---|---|---|
| 8 | 2 | 2 | 2 | 2 | 2 | 2 |

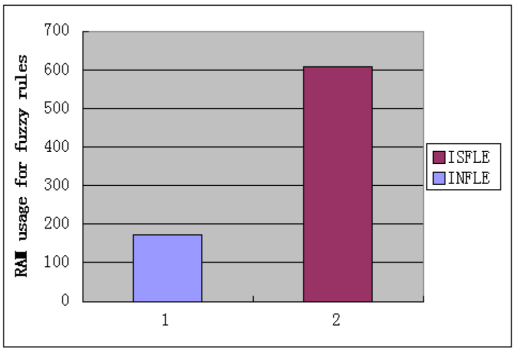



**Figure 10. Two Algorithms on RAM Usage for Fuzzy Rules in Sensor Nodes**

## 4. Related Work

C. T. Vu propose a threshold-based composite event detection method, in which an event is decomposed into a number of sub-events [11]. An event is detected if all sub-events occur simultaneously. A consensus-based threshold-based event detection scheme for volcano monitoring is presented in [12]. The authors propose a complete framework for monitoring volcano activities and then implement the approach on TMote Sky sensor node.

M. Bager [13] propose a voting graph neuron (VGN) algorithm to detect events distributed in large-scaled sensor networks. VGN algorithm is based on the distributed cooperative problem solving concept that solves a problem by breaking it into smaller parts. In this approach, event patterns are stored in a distributed graph over the network and then events are detected by matching sensory data of each sensor node with a subset of the graph. A Noise-Tolerant Event and Event Boundary Detection scheme [14] is an event detection scheme that can detects both events and event boundaries distributed in WSNs. The approach uses a moving average technique for noise effect reduction and a statistical method for event and event boundary detection.

Authors in [15] propose a local event detection scheme that uses principal component analysis to get the signature of events. They then use a threshold to separate event data from non-event data to check whether events happened. F. Martincic *et al.* proposed another signature matching technique [16] that divides the whole network into cells and detect events by comparing the cell's signature with event's signature.

## 5. Conclusion

A disadvantage of INFLE is that when fire ignites, it may produce some false positives when the selected nodes are far from the detected node. In this paper, we show that our ISFLE can solve this problem by putting the "Distance" variable into fuzzy logic system. However, ISFLE may cost much more energy on storing the fuzzy rules on sensor nodes compared with INFLE.

Copyright ⓒ 2015 SERSC

## Acknowledgements

This work is supported by the NSFC (61232016, 61173141, 61173142, 61173136, 61103215, 61373132, 61373133, 61300238), GYHY201206033, 201301030, 2013DFG12860, BC2013012, BK20131004 and PAPD fund.

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A survey on sensor networks”, Communications Magazine, IEEE, vol. 40, no. 8, **(2002)** August, pp. 102-114.
[2] N. Wang and X. Meng, "Real-time forest fire detection with wireless sensor networks", Wireless Communications, Networking and Mobile Computing, **(2005)**.
[3] Q. Liang and L. Wang, "Event detection in wireless sensor networks using fuzzy logic system”, Computational Intelligence for Homeland Security and Personal Safety (CIHSPS), Institute of Electrical and Electronics Engineers (IEEE), USA, **(2005)**, pp. 52-55.
[4] Y. Guo, F. Hong, Z. Guo, Z. Jin and Y. Feng, “EDA: Event-oriented Data Aggregation in Sensor Networks”, IPCCC2009, Phoenix, Arizona, USA, **(2009)**.
[5] M. Marin-Perianu and P. Havinga, “D-FLER: a distributed fuzzy logic engine for rule-based wireless sensor networks”, Proceedings of the 4th International Conference on Ubiquitous Computing Systems (UCS), Springer-Verlag, Berlin, Heidelberg, **(2007),** pp. 86–101.
[6] K. Kapitanova, Sang H. Son and K.-D. Kang, “Using fuzzy logic for robust event detection in wireless sensor networks”, Ad Hoc Networks, USA, **(2012)**.
[7] X. Sun, A. Zhuang, B. Wang, T. Li and Q. Liu, “INFLE:An improved neighbor-based fuzzy logic event detecting algorithm for wireless sensor networks”, International Journal of Future Generation Communication and Networking, vol. 7, no. 3, (**2014**), June.
[8] G. J. Klir and B. Yuan, "Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall Inc", Upper Saddle River, NJ, USA, **(1995)**.
[9] G. of Canada, “National Research Council Canada, Institute for Information Technology”, FuzzyJ ToolKit from the Java(tm) Platform & FuzzyJess–IIT-ITI- NRC-CNRC, **(2011)**.
[10] H R. Berenji and P. Khedkar, “Learning and tuning fuzzy logic controllers through reinforcements [J]”, Neural Networks, IEEE Transactions on, vol. 3, no. 5, **(1992)**, pp. 724-740.
[11] C. T. Vu, Raheem A. Beyah and Yingshu Li, "Composite Event Detection in Wireless Sensor Networks". Performance, Computing, and Communications Conference, **(2007)**, IPCCC 2007. IEEE International, (**2007**).
[12] G. Werner-Allen, "Fidelity and yield in a volcano monitoring sensor network", The 7th symposium on Operating systems design and implementation (OSDI '06), **(2006).**
[13] M. Baqer and A. I. Khan, “Event Detection in Wireless Sensor Networks Using a Decentralised Pattern Matching Algorithm”, White Paper, **(2008)**.
[14] G. Jin adn S. Nittel, "NED: An Efficient Noise-Tolerant Event and Event Boundary Detection Algorithm in Wireless Sensor Networks”, 7th International Conference on Mobile Data Management, **(2006)**.
[15] J. Gupchup and Alex Szalay, “Model-Based Event Detection in Wireless Sensor Networks”, Workshop on Data Sharing and Interoperability on the World-Wide Sensor Web (DSI), **(2007)**.
[16] F. Martincic and L. Schwiebert, “Distributed Event Detection in Sensor Networks”, Systems and Networks Communications, ICSNC '06., Tahiti **(2006)**.

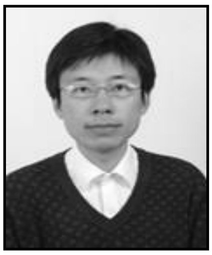
## Authors

**Baowei Wang,** He received his B.S. and Ph.D. degrees in Computer Science from Hunan University in 2005 and 2011, respectively. He is currently working as a lecturer in School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include steganography, wireless networks and securing ad hoc networks.

 Copyright ⓒ 2015 SERSC

**Aoran Zhuang,** He received his B.S. degree in network engineering from Shandong agriculture University, China in 2011. Currently he is studying for his M.S degree in computer application technology at Nanjing University of Information Science and Technology, China. His research interests include wireless networks and event detection.

**Huiyu Sun,** He received his B.S. degree in 2014 and is currently working as a networking and security consultant at Nanjing Wisdom Technology-a product development company specializing in wireless networks. His research interests include cyber security and wireless networks.



**Tao Li,** He received his B.S. and M.D. degrees in Computer Science and technology from Nanjing University of Technology in 2002 and 2004, respectively. He received his Ph.D. degrees in Institute of information science and engineering from Southeast University in 2010. His research interests include embedded system and wireless sensor network routing technology.

**Xingming Sun,** He is a professor in the School of Computer and Software, Nanjing University of Information Science and Technology, China from 2011. He received the B.S.degree in Mathematical Science from Hunan Normal University and M.S. degree in Mathematical Science from Dalian University of Technology in 1984 and 1988, respectively. Then, he received the Ph.D. degree in Computer Engineering from Fudan University in 2001. His research interests include information security, network security, cryptography and ubiquitous computing security.

Copyright ⓒ 2015 SERSC

Copyright ⓒ 2015 SERSC