International Journal of Future Generation Communication and Networking
Vol.7, No.3 (2014), pp.13-24

INFLE: An Improved Neighbor-Based Fuzzy Logic Event Detecting Algorithm for Wireless Sensor Networks

Xingming Sun[1,2], Aoran Zhuang[1], Baowei Wang[1,2], Tao Li[1,2] and Qi Liu[1,2]

School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
Jiangsu Engineering Center of Network Monitoring, Nanjing 210044, China
sunnudt@163.com, hangsomelaozhuang@163.com, wbw.first@163.com, taoleehome@hotmail.com, qrankl@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 Neighbor-based Fuzzy Logic algorithm, namely NFLE, which belongs to machine learning technology. However, traditional fuzzy logic algorithm cannot work well in high-precision fire detecting networks. In this paper, we propose an improved NFLE (INFLE) which can dramatically increase the precision of fire detection. In NFLE, the final fire confidence of one node is partly determined by the average readings of its neighbors in fuzzy logic system, which may lead to inaccurate fire detection when event occurred in an area that covers only part of its neighbors. In our proposed INFLE, we select some of neighbors, by specific rules, to determine node's final state. The simulation results validate that our proposed INFLE outperforms traditional NFLE in event detecting precision.

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

1.Introduction

A wireless sensor network may contain tens to thousands of wireless sensor nodes to 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 environment and it must be accuracy and produces less false alarm [1].

The initial idea of 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. For example, in fire event detection, if "temperature>=55℃", fire is present. Threshold-based event detection has the following features: a) easy to implement; b) Hard to set an appropriate threshold for some event; c) when events become complicated, false alarms will rise quickly.

Model-based event detection techniques are another detecting techniques that model event phenomena in a specific form such as mathematic formulas or maps. Model-based event detection has the following features: a) can handle more complex events than threshold-based, as they represent a non-linear model for different events; b) need professional knowledge to tune the parameter of the models; c) due to complexity of events models, they are always computationally intensive. Pattern matching-based event

ISSN: 2233-7857 IJFGCN

detection methods define a data pattern for events and then an event is detected if data pattern matches with event pattern.

Event signatures are matrices that can be used to storing special-related values, event is detected by comparing the sensor node's data with event signatures. Pattern matching-based event detection has the following features: a) usually can handle complex events; b) are flexible and adaptable to various applications; c) need an expert knowledge to tune the right parameters of the approaches [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 than crisp logic. 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 nuisance tests. However, the author used does not analyze accuracy of D-FLER.

However, there still exists a lot of problems for event detecting using fuzzy logic. In [6], the author proposes a NFLE to increase the fire detection accuracy. The experiment use real fire data in NIST, and the experiment contain two algorithms for detecting fire on one node: A) only using own sensor readings for event detection; B) Neighbor-based Fuzzy Logic for event detecting(NFLE). Pink line represents algorithm A and yellow line represents algorithm B in Figure1. Algorithm B is proved to produce less false alarms than algorithm A before fire ignition. While after fire actually spreads to one node at 23min as circled in Figure1, algorithm B may cause fire confidence low which will not raise a fire alarm in the node. The main reason is that when fire spreads to one sensor node and has not spread to its neighbor node, readings of neighbor node kept low which have negative effects on the fire confidence. We propose INFLE mainly purposed to solve this problem for NFLE after fire is ignited.

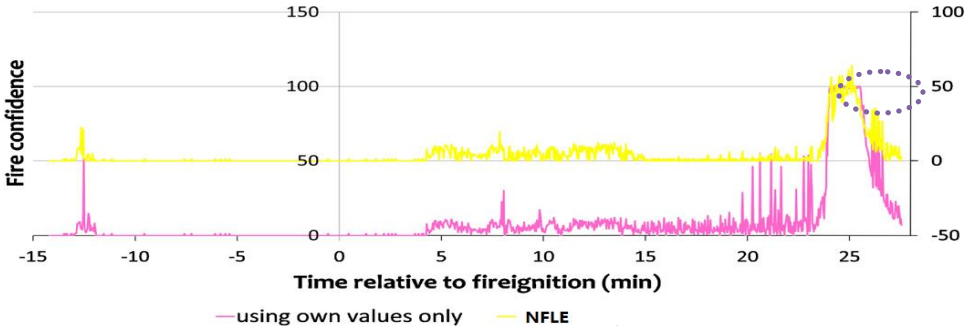


Figure 1. Comparison Between Algorithm A and B

The paper is organized as follows: Section 2 introduces INFLE including neighbor nodes selection algorithm and our fuzzy logic model for detecting fire. Section 3 mainly

states a comparison result for detection accuracy between INFLE and NFLE in a simulation. Some other methods for event detection have been concluded in Section 4. Section 5 concludes the paper.

2. Our Proposed Improved Neighbor-Based Fuzzy Logic Event Detection Scheme

For any node to be detected, it should see whether any of its neighbor node report fire. If there is no neighbor node reporting fire for 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 broadcasts its state to all neighbors. Or else, the node and its neighbors will run NFLE algorithm.

If there is any node reporting fire for the detected node, the detected node and its neighbors will run our INFLE algorithm. Four outputs for INFLE including Temperature1, Smoke1, Temperature2, Smoke2 will be added into fuzzy logic system. Then we can get a fire confidence for the node from fuzzy logic system. If fire confidence for the node is 100%High, fire is confirmed on the node and the node raise a fire alarm and broadcast to all neighbors.

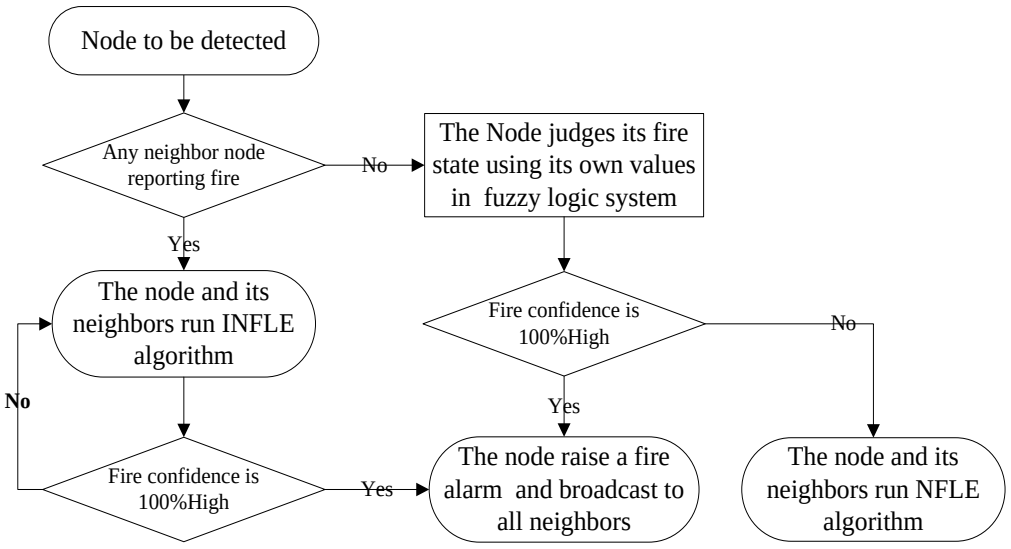


Figure 2. Scheme for Detecting Fire on Sensor Nodes

2.1. INFLE Algorithm

INFLE 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.

2.1.1. Neighbor Nodes Selection Algorithm

In neighbor nodes selection algorithm, at first, node A will see whether there is any neighbor node that reports fire. If yes, A will keep these nodes coordinate and calculate the distance. Then suppose node E is chosen as the node which reports fire that locate 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 from the nodes which are neighbors of both A and E. In Figure 3, the nodes in black dotted line and red dotted line are what we want. In Figure3 five nodes are selected in C1, then we choose the average readings of these five nodes

Temperature2, Smoke2 and readings of A as Temperature1, Smoke1 as our output. While in NFLE, all readings in C2 will be added into Temperature2 and Smoke2, which will have a negative effect on the output fire confidence when fire has spread to A but not spread to most of its neighbors.

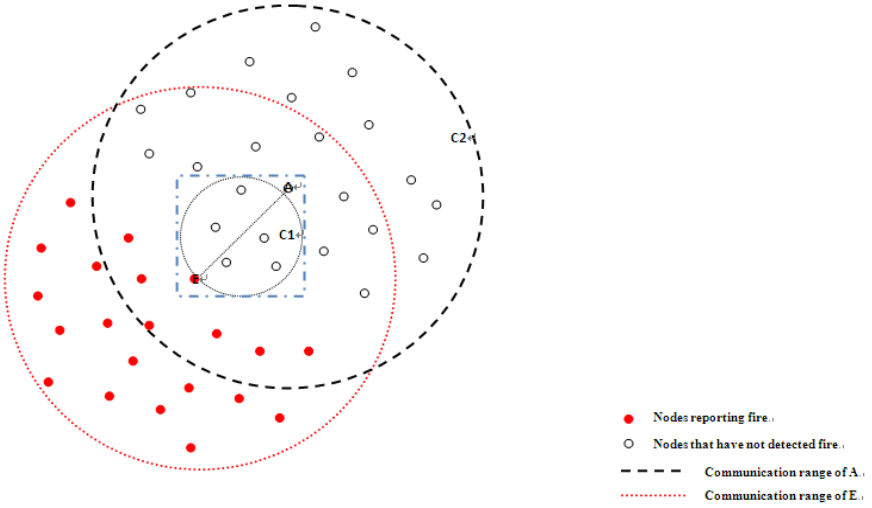


Figure 3. Detecting Fire on Node A Using Neighbor Nodes Selection Algorithm

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 coordinate are separately $(x_H, y_H), (x_L, y_L)$ are $D_{HL}=\sqrt{(x_H - x_L)^2 + (y_H - y_L)^2}$.

Definition 4: Temperature(A) and Smoke(A) represent Temperature1 and Smoke1 in our fuzzy logic system, while mean readings of selected neighbors represent Temperature2 and Smoke2.

Neighbor Nodes Selection Algorithm

Input: Node A that is to be detected and all neighbors A'.

Output: Readings of A- Temperature1, Smoke1 and its selected neighbors- Temperature2, Smoke2.

1: Every node broadcasts its coordinate on the network and keeps its neighbors' coordinate.
2: Any time t we want to detect fire on A after ignition.
3: **while** any node in A' reports fire **do**
4: A calculate and store the distance between A and this node.
5: A find the shortest distance that are stored in A.
6: Suppose the node is E whose coordinate is (x_i, y_i).
7: Count=0. SumTemperature=0. SumSmoke=0.
8: **For** each node G(x_g,y_g) that have same coordinate in Q **do**.

9: **If** $x_i + \frac{|x_1 - x_i|}{2} - \frac{D_{AE}}{2} <= x_g <= x_i + \frac{|x_1 - x_i|}{2} + \frac{D_{AE}}{2}$ && $y_i + \frac{|y_1 - y_i|}{2} - \frac{D_{AE}}{2} <= y_g <= y_i + \frac{|y_1 - y_i|}{2} + \frac{D_{AE}}{2}$ **then**;
10: put G into F.
11: **end if**
12: **For** each node M in F whose coordinate is $(x_{m,} y_m)$ **do**
13: **If** $D_{MA}^2 + D_{ME}^2 <= D_{AE}^2$ **then**;
14: Count++. SumTemperature= SumTemperature + Temperature(M).
15: SumSmoke= SumSmoke + Smoke(M).
16: **end if**
17: **end for**
18: **end for**
19: **end while**
20: Temperature1= Temperature(A). Smoke1= Smoke(A).
21: Temperature2= SumTemperature/Count. Smoke2= SumSmoke/Count.
22: **Return** Temperature1, Smoke1, Temperature2, Smoke2.

2.1.2. Fuzzy Logic Model

There are four outputs Temperature1, Smoke1, Temperature2, Smoke2 after neighbor nodes selection algorithm. Our fuzzy logic model consists of four steps. Firstly, the fuzzifier converts the crisp input variables including Temperature1, Smoke1, Temperature2, Smoke2 into fuzzy linguistic variables. Secondly we can get a series of output that is in the form of percentage for some rules by applying T fuzzy operator. Thirdly, the output in the form of percentage can be converted into a combined fuzzy output by fuzzy implication. Finally, an crisp output will be got by defuzzifier. Figure4 shows the process of our fuzzy logic system. Figure4 shows the structure of a fuzzy logic system.

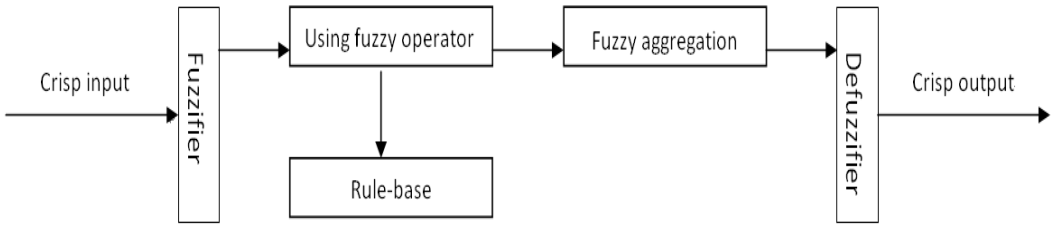


Figure 4. The Structure of a Fuzzy Logic System

2.1.2.1. 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.

2.1.2.2. 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 are set by domain experts. If one rule has more than one antecedent, for example, a rule is composed of t antecedents and 1 output:

$R^i : \text{IF } x_1 \text{ is } S_1^i \text{ and } x_2 \text{ is } S_2^i \text{ and} \ldots \text{and } x_t \text{ is } S_t^i \text{ THEN y is } A^i$

When input $x' = \{x_1', x_2', ..., x_t'\}$, the degree of firing of the rule can be computed as:

$$f_{s_1^i}(x_1') * f_{s_2^i}(x_2') * ... * f_{s_t^i}(x_3') = T_{l=1}^t f_{s_l^i}(x_l')$$

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]

Table 1. Format of our Fuzzy Rule

Rule	Temperature1	Smoke1	Temperature2	Smoke2	Confidence

Table 1 describes the form of our fuzzy rule, which has four inputs called Temperature1, Smoke1, Temperature2, Smoke2 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. The higher output Confidence is, the higher possibility of fire happened in the detected node.

2.1.2.3. 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[7].

2.1.2.4. 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 example, 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 Figure 5 shows to compare INFLE with NFLE for fire detecting. There are 100 nodes which have ids from 0 to 100 deployed randomly in a circle 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 the fire spread in circle rings whose speed is 2m/s. Node's sample rate is set to 1Hz.

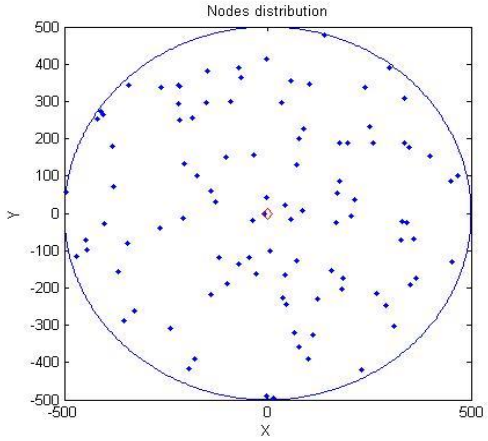


Figure 5. Experiment Environment

3.2. Our Fuzzy Logic System

The fuzzy logic system for NFLE and INFLE are displayed through the FuzzyJ Toolkit for Java [8]. Temperature, Smoke membership functions and output fire confidence membership functions are showed in Figure6.

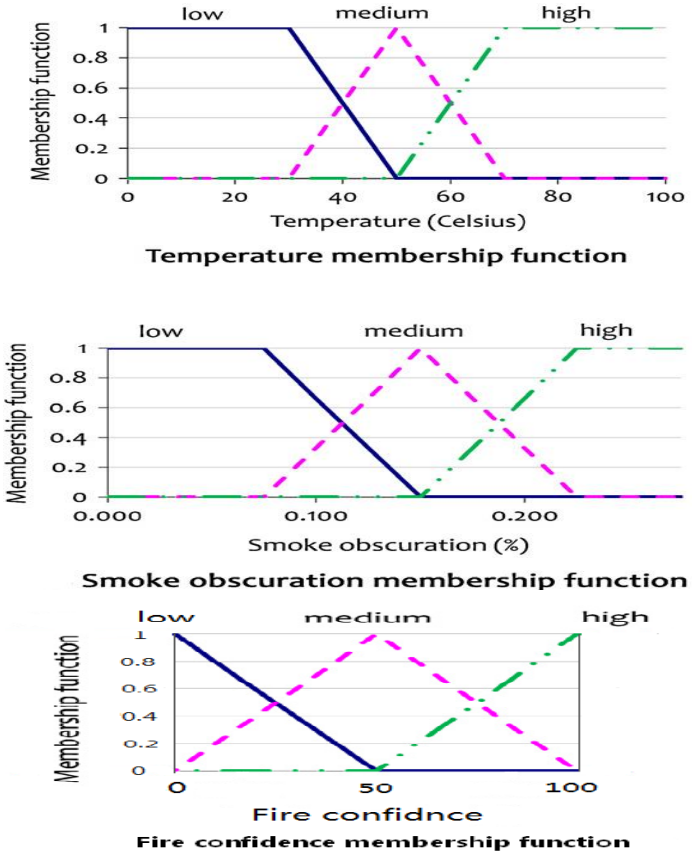


Figure 6. Membership Functions for Input and Output in our Fuzzy Logic System

Besides, we define rules for our fuzzy logic system based on parts of NFLE which is showed in Table1. One rule consists of four inputs and one output.

Table 2. Rules for our Fuzzy Logic System

Rule	Temperature1	Smoke1	Temperature2	Smoke2	Confidence
1	L	L	<=H	<=M	L
2	L	L	M	H	L
3	L	L	H	H	M
4	H	H	M	L	M
5	H	H	L	<=H	M
6	H	H	M	M	M
7	H	H	M	>M	H
8	H	H	H	<=H	H
9	M	L	L	<=H	L
10	M	L	M	L	L
11	M	L	>=M	>=M	M
12	M	L	H	L	M
13	M	M	>=L	>L	M
14	M	M	L	L	L
15	M	M	>=M	L	M
16	M	H	<=H	<=M	M
17	M	H	<=M	H	M
18	M	H	H	H	H
19	L	M	<=M	<=M	L
20	L	M	L	H	L
21	L	M	H	L	L
22	L	M	H	>M	M
23	L	M	M	H	M
24	L	H	L	<=H	L
25	L	H	>=M	>L	M
26	L	H	M	L	L
27	L	H	H	L	M
28	H	M	<=M	<=H	M
29	H	M	H	L	M
30	H	M	H	>=M	H
31	H	L	>=L	>=L	M

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. When output confidence reaches 83.3 which is 100%high, fire is considered to happen in the node.

3.3. Simulation Results

At time 0 when fire is ignited, fire starts from the center of the circle. The Figure7 presents INFLE and NFLE 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. Fire should spreads to node 100 at 129.8s. Since sample rate of sensor node is set to 1hz, fire confidence of INFLE reach 83.3 which is 100%high at 130s. However, fire confidence of NFLE does not reach 83.3 until 172s when fire spread to all its neighbors.

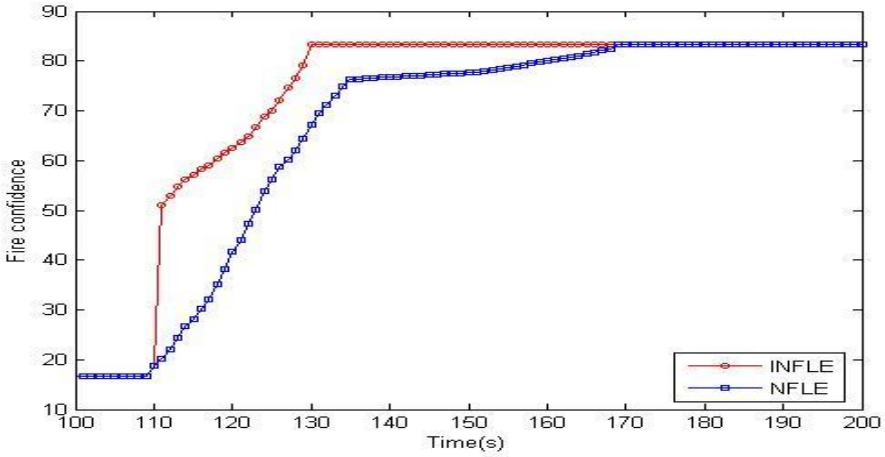


Figure 7. Comparison Between INFLE and NFLE on Detecting Fire on Node 100

Figure 8 demonstrates the comparison between our INFLE and NFLE on detecting fire for all nodes in the detected area. Nodes will raise fire alarm when fire confidence reaches 83.3 which is 100%high. We can see that our INFLE can detect fire much quicker than NFLE when fire actually spreads to these sensor nodes. Every time fire spread to one node, INFLE can immediately detect it, however NFLE cannot raise a fire alarm until fire spreads to all of its neighbors.

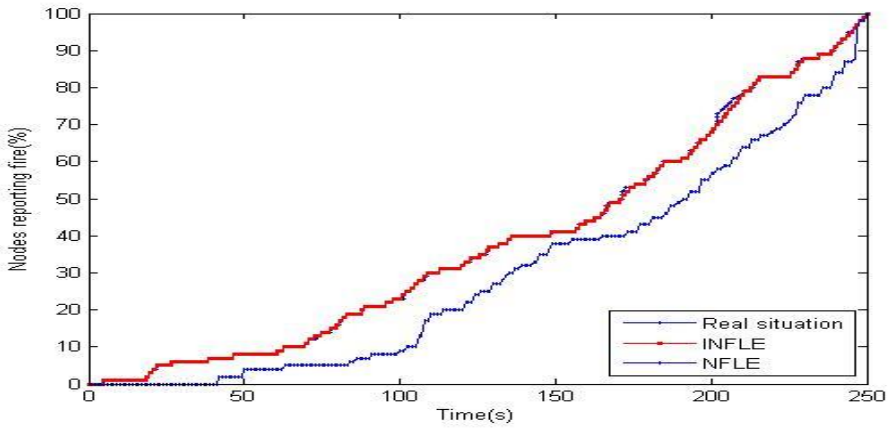


Figure 8. Comparison between INFLE and NFLE for Reporting Fire on Sensor Nodes

3.4. Energy Costs and Time Complexity Analyses for INFLE and NFLE

When fire is ignited, one node in NFLE will need to communicate with all neighbor nodes at a sample time. In above experiment environment, after node 100 receives the fire alarm from its neighbor node 91 at 111s, our INFLE find that there is no node between the circle built by node 91 and 100, readings of node 91 is selected as Temperature2 and Smoke2. Node 100 only need to communicate with one node 93 at 111s. Then at 115s, node 85 reports fire, there is no node between the circle built by node 100 and node 85, so node 100 only need to communicate with node 85 at 115s in INFLE. When fire is confirmed on node 100, INFLE will select readings of node 95 that is nearest to 100 which reports fire. While in NFLE, node 100 needs to communicate with its six neighbor

nodes every sample time. Once fire is ignited, INFLE will save much energy on communicating with neighbor nodes compared with NFLE.

In the detected node, INFLE need to find the appropriate nodes that have the most space correlation with the detected node every sample time. For one node that has n neighbors, time complexity of INFLE is $O_{(o*q*l)}$(o<=n, q<=n && q<=k, *l*<=q). NFLE only needs to know all neighbors' reading the detected node every sample time, so its time complexity is $O_{(n)}$. INFLE may cost more energy on sensor nodes to decide which neighbors to choose, however, energy consumed on choosing neighbor nodes is far less than energy consumed on communication between sensor nodes.

So it is more important to reduce the communication energy than to reduce the energy for operation on sensor node. INFLE is proved to be able to work better and saves more energy than NFLE.

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[9]. An event is detected if all sub-events occur simultaneously. A consensus-based threshold-based event detection scheme for volcano monitoring is presented in [10]. The authors propose a complete framework for monitoring volcano activities and then implement the approach on TMote Sky sensor node.

M. Bager [11] 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 [12] 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 [13] 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[14] 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 NFLE is that when fire ignites, it makes an imprecise decision on detecting fire until fire spreads to most of their neighbor nodes. In this paper, we show that our INFLE can solve this problem by selecting appropriate neighbor nodes. Fire alarm will be raised once fire confidence reaches 83.33, INFLE not only can timely detect fire every time fire spreads to one node, but also save much energy when fire is ignited.

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] 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.

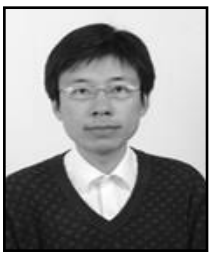
[2] N. Wang and X. Meng, "Real-time forest fire detection with wireless sensor networks", Wireless Communications, Networking and Mobile Computing, **(2005)**.
[3] Q. Liang and L. Wang, "Event detection in wireless sensor networks using fuzzy logic system", Computational Intelligence for Homeland Security 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. M. 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, S. H. Son and K.-D. Kang, "Using fuzzy logic for robust event detection in wireless sensor networks", Ad Hoc Networks, USA, **(2012)**.
[7] G. J. Klir and B. Yuan, "Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall Inc", Upper Saddle River, NJ, USA, **(1995)**.
[8] G. of Canada, "National Research Council Canada", Institute for Information Technology, FuzzyJ ToolKit from the Java(tm) Platform & FuzzyJess–IIT-ITI- NRC-CNRC, **(2011)**.
[9] C. T. Vu, R. A. Beyah and Yingshu Li, "Composite Event Detection in Wireless Sensor Networks", Performance, Computing, and Communications Conference, IPCCC, IEEE Internationa, **(2007)**.
[10] G. W. Allen, "Fidelity and yield in a volcano monitoring sensor network", The 7th symposium on Operating systems design and implementation (OSDI '06), **(2006)**.
[11] M. Baqer and A. I. Khan, "Event Detection in Wireless Sensor Networks Using a Decentralised Pattern Matching Algorithm", White Paper, **(2008)**.
[12] G. Jin and 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)**.
[13] J. Gupchup and A. Szalay, "Model-Based Event Detection in Wireless Sensor Networks". Workshop on Data Sharing and Interoperability on the World-Wide Sensor Web (DSI), **(2007)**.
[14] F. Martincic and L. Schwiebert, "Distributed Event Detection in Sensor Networks". Systems and Networks Communications, ICSNC '06., Tahiti, **(2006)**.

Authors

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.

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.

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.



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.

Qi Liu, he received his BSc degree in Computer Science and Technology from Zhuzhou Institute of Technology, China in 2003, and his MSc and PhD in Data Telecommunications and Networks from the University of Salford, UK in 2006 and 2010. His research interests include context awareness, data communication in MANET and WSN, and smart grid. His recent research work focuses on intelligent agriculture and meteorological observation systems based on WSN.