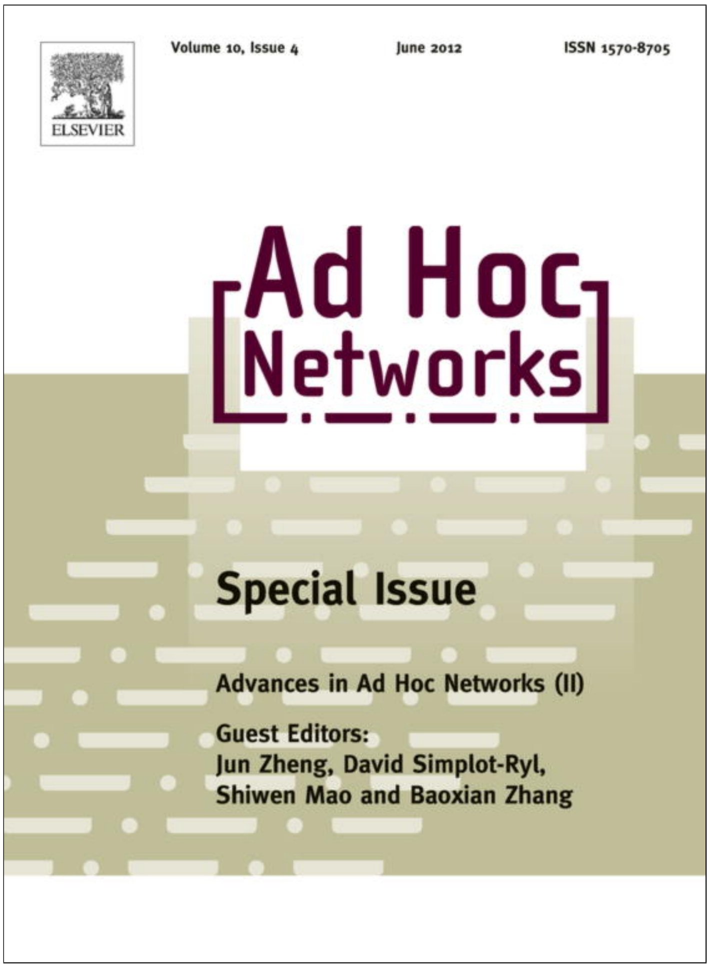
Volume 10, Issue 4
June 2012
ISSN 1570-8705
ELSEVIER
Ad Hoc Networks
Special Issue
Advances in Ad Hoc Networks (II)
Guest Editors:
Jun Zheng, David Simplot-Ryl,
Shiwen Mao and Baoxian Zhang

Ad Hoc Networks 10 (2012) 709–722

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Using fuzzy logic for robust event detection in wireless sensor networks

Krasimira Kapitanova [a,*], Sang H. Son [a], Kyoung-Don Kang [b]

[a] *University of Virginia, Charlottesville, VA, USA*
[b] *Binghamton University, Binghamton, NY, USA*

A R T I C L E I N F O

Article history:
Received 9 March 2011
Received in revised form 16 May 2011
Accepted 20 June 2011
Available online 26 June 2011

Keywords:
Wireless sensor networks
Fuzzy logic
Event description
Event detection accuracy

A B S T R A C T

Event detection is a central component in numerous wireless sensor network (WSN) applications. Nevertheless, the area of event description has not received enough attention. The majority of current event description and detection approaches rely on using precise values to specify event thresholds. However, we believe that crisp values cannot adequately handle the often imprecise sensor readings. In this paper we demonstrate that using fuzzy values instead of crisp ones significantly improves the accuracy of event detection. We also show that our fuzzy logic approach provides higher event detection accuracy than two well-established classification algorithms.

A disadvantage of using fuzzy logic is the exponentially growing size of the fuzzy logic rule-base. As sensor nodes have limited memory, storing large rule-bases could be a challenge. To address this issue, we have developed a number of techniques that help reduce the size of the rule-base by more than 70%, while preserving the event detection accuracy.

1. Introduction

Event detection is one of the main components in numerous wireless sensor networks (WSNs). WSNs for military application are deployed to detect the invasion of enemy forces, health monitoring sensor networks are deployed to detect abnormal patient behavior, fire detection sensor networks are deployed to set an alarm if a fire starts somewhere in the monitored area. Regardless of the specific application, the network should be able to detect if particular events of interest, such as fire, have occurred or are about to. However, similar to many other human-recognizable events, the phenomenon *fire* has no real meaning to a sensor node. Therefore, we need suitable techniques that would allow us to describe events in ways that sensor nodes would be able to "understand". The area of event description and detection in WSNs, however, has not been explored much.

Most previous work on event description in WSNs uses precise, also called *crisp*, values to specify the parameters that characterize an event. For example, we might want to know if the temperature drops below 5 °C or the humidity goes above 46%. However, sensor readings are not always precise. In addition, different sensors, even if located close to each other, often vary in the values they register. Consider an example scenario where we want the air conditioning in a room to be turned on if the temperature goes above 5 °C. Two sensors, *A* and *B*, measure the temperature in the room. The average of their values is used to determine if an action should be taken. At some point, sensor *A* reports 5.1 °C and sensor *B* reports 4.8 °C. The average, 4.95 °C, is below our predefined threshold and the cooling remains off. However, if sensor *B*'s measurement is inaccurate and, therefore, lower than the actual temperature, we have made the wrong decision. The situation becomes even more complex when more than two sensor measurements are involved. This makes determining the precise event thresholds an extremely hard task which has led us to believe that using crisp values to describe WSN events is not the most suitable approach. Fuzzy logic, on the other hand, might be able to address these challenges better than crisp logic.

Fuzzy logic has a number of properties that make it suitable for describing WSN events:

* Corresponding author.
E-mail addresses: krasi@virginia.edu (K. Kapitanova), son@virginia.edu (S.H. Son), kang@cs.binghamton.edu (K.-D. Kang).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.06.008

- It can tolerate unreliable and imprecise sensor readings.
- 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 °C and smoke obscuration level above 15%.
- Compared to other classification algorithms based on probability theory, fuzzy logic is much more intuitive and easier to use.

A disadvantage of using fuzzy logic is that storing the rule-base might require a significant amount of memory. The number of rules grows exponentially to the number of variables. With n variables each of which can take m values, the number of rules in the rule-base is m^n. Adding spatial and temporal semantics to the decision process further increases the number of rules. Since sensor nodes have limited memory, storing a complete rule-base on every node might not be reasonable. In addition, constantly traversing a large rule-base might considerably slow down the event detection. To address this problem, we have designed a number of techniques that reduce the size of the rule-base. A key property of these techniques is that they do not decrease the event detection accuracy of the system.

This paper has three main contributions. First, we show that using fuzzy logic results in more accurate event detection than when either crisp values or well established classification algorithms, such as Naive Bayes classifiers or decision trees, are used. Second, we incorporate event semantics into the fuzzy logic rule-base to further improve the accuracy of event detection. Third, we have designed techniques that can be used to prevent the exponential growth of the rule-base without compromising the accuracy of event detection.

The rest of the paper is organized as follows: We discuss the related work in Section 2. Section 3 introduces a brief overview of fuzzy logic and fuzzy systems. Section 4 discusses the spatial and temporal semantics of wireless sensor network events. Section 5 describes the reduction techniques we use to decrease the size of the rule-base. We evaluate and analyze how using fuzzy logic affects the accuracy and timeliness of event detection in Sections 6 and 7 concludes the paper.

2. Related work

2.1. Event detection

Relatively little research has focused on providing methods for event description and detection in WSNs that can support data dependency and collaborative decision making. The prevailing approach is to use SQL-like primitives [1–4]. The papers that employ this method vary in semantics. In [1,2], the authors use general SQL primitives to define events in sensor networks. The limitation of this approach is that the events can only be defined by predicates on sensor readings with very simple temporal and spatial constraints connected by AND and OR operators. Madden et al. have extended the SQL primitives by incorporating streaming support where a desired sample rate can be included [3]. Li et al. define events using a sub-event list and confidence functions in SQL [4]. However, SQL is not very appropriate for describing WSN events. Some of its drawbacks include that it: (i) cannot capture data dependencies and interactions among different events or sensor types; (ii) does not explicitly support probability models; (iii) is awkward in describing complex temporal constraints and data dependencies; (iv) lacks the ability to support collaborative decision making and triggers [5]; (v) does not facilitate any analysis of the event system.

Another approach to formally describe events in WSNs has been the use of extended Petri nets. This was initially proposed by Jiao et al. [6]. The authors design a Sensor Network Event Description Language (SNEDL) which can be used to specify event logic for WSN applications. Petri nets were also used in MEDAL [7], an extension of SNEDL that supports the description of additional WSN specific features such as communication, actuation, and feedback control. Both SNEDL and MEDAL, however, use crisp values in the definitions of their Petri nets.

2.2. Stochastic methods

There is a long history of using stochastic formalisms in different WSN applications. Bayesian classifiers and hidden Markov models have been extensively used in activity recognition [8,9] and decision fusion [10,11]. Dempster–Shafer evidence theory has been applied to intrusion detection [12], sensor fusion [13,14], and assisted living applications [15]. Probabilistic context free grammars have been used to solve problems such as inferring behaviors [16] as well as movement and activity monitoring [17,18].

2.3. Fuzzy logic

Fuzzy sets and logic were introduced by L. Zadeh in 1965. Numerous fields have taken advantage of their properties since then. In WSNs, fuzzy logic has been used to improve decision-making, reduce resource consumption, and increase performance. Some of the areas it has been applied to are cluster-head election [19,20], security [21,22], data aggregation [23], routing [24,25], MAC protocols [26], and QoS [27,28]. However, not much work has been done on using fuzzy logic for event description and detection. Liang and Wang [29] 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 classification accuracy.

In D-FLER [30] 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 approach used in D-FLER does not incorporate any temporal semantics. In addition, since all of the experiments last only 60 s after the fire ignition, the authors do not analyze the number of false alarms raised by D-FLER.

3. Overview of fuzzy logic

Fig. 1 shows the structure of a general fuzzy logic system (FLS). The fuzzifier converts the crisp input variables $x \in X$, where X is the set of possible input variables, to fuzzy *linguistic variables* by applying the corresponding membership functions. Zadeh defines linguistic variables as "variables whose values are not numbers but words or sentences in a natural or artificial language" [31]. An input variable can be associated with one or more fuzzy sets depending on the calculated membership degrees. For example, a temperature value can be classified as both Low and Medium.

The fuzzified values are processed by *if–then* statements according to a set of predefined rules derived from domain knowledge provided by experts. In this stage the inference scheme maps input fuzzy sets to output fuzzy sets. Finally, the defuzzifier computes a crisp result from the fuzzy sets output by the rules. The crisp output value represents the control actions that should be taken. The above three steps are called fuzzification, decision making, and defuzzification, respectively. We describe each of them in more detail in the following subsections.

3.1. Fuzzification

The fuzzifier converts a crisp value into degrees of membership by applying the corresponding membership functions. A membership function determines the certainty with which a crisp value is associated with a specific linguistic value. Fig. 2 shows an example of a temperature membership function. According to this membership function, a temperature value of −2 °C is classified as 20% Freezing and 80% Cold. The membership functions can have different shapes. Some of the most frequently used shapes include triangular, trapezoidal, and Gaussian-shaped. Membership functions are defined by either relying on domain knowledge or through the application of different learning techniques, such as neural networks [32,33] and genetic algorithms [34].

3.2. Decision making

A rule-base consists of a set of linguistic statements, called rules. These rules are of the form **IF premise, THEN consequent** where the *premise* is composed of fuzzy input variables connected by logical functions (e.g. AND, OR, NOT) and the *consequent* is a fuzzy output variable. The rule-base is usually generated as an exhaustive set of all possible value-combinations for the input linguistic variables that constitute the *premise*. Similarly to how membership functions are defined, the rule-base is derived either based on domain knowledge, or through using machine learning techniques. Consider a t-input 1-output FLS with rules of the form:

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

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

$$\mu_{S_1^i}(x'_1) * \mu_{S_2^i}(x'_2) * \cdots * \mu_{S_t^i}(x'_t) = T_{l=1}^{t} \mu_{S_l^i}(x'_l)$$

Here μ represents the membership function and both $*$ and T indicate the chosen triangular norm. A triangular norm is a binary operation such as AND or OR applied to the fuzzy sets provided by the membership functions [35].

3.3. Defuzzification

Executing the rules in the rule-base generates multiple shapes representing the modified membership functions. For example, a set of rules designed to decide the probability that there is a fire may produce the following result: Low (56%), Medium (31%), and High (13%). Defuzzification is the transformation of this set of percentages into a single crisp value. Based on how they perform this transformation, defuzzifiers are divided into a number of categories. The most commonly used defuzzifiers are *center of gravity*, *center of singleton*, and *maximum methods* [35]:

- The *center of gravity* approach finds the centroid of the shape obtained by superimposing the shapes resulting from applying the rules. The output of the defuzzifier is the x-coordinate of this centroid.
- The defuzzification process can be significantly simplified if the *center of singleton* method is used. With this method, the membership functions for each rule are defuzzified separately. Each membership function is reduced to a singleton which represents the function's center of gravity. The simplification consists in that the singletons can be determined during the design of the system. The center of singleton method is an approximation of the center of gravity method. Although experiments have shown that there are slight differences between these two approaches, in most cases the differences can be neglected [36].
- The class of *maximum methods* determines the output by selecting the membership function with the maximum value. If the maximum is a range, either the lower, upper, or the middle value is taken for the output value depending on the method. Using these methods, the

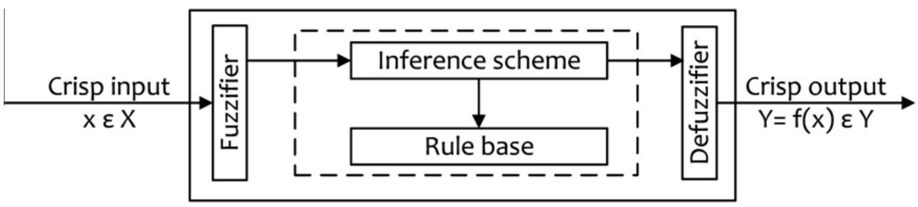

Fig. 1. The structure of a fuzzy logic system.

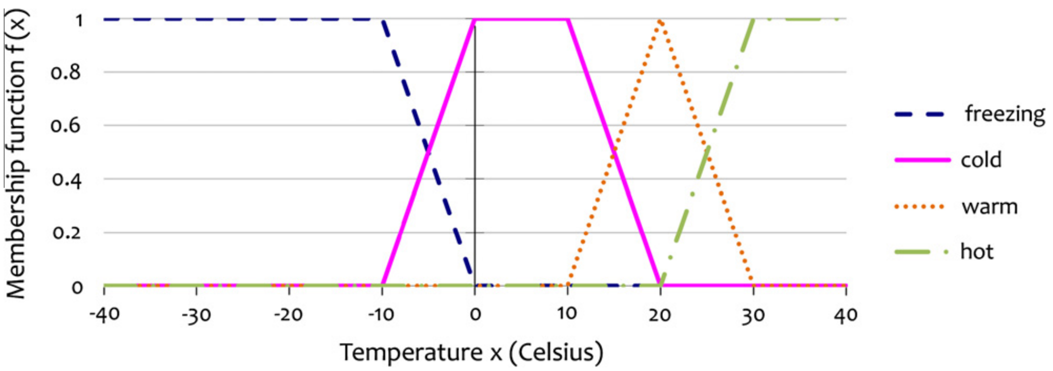

Fig. 2. Temperature membership function.

rule with the maximum activity always determines the output value. Applying this approach to the aforementioned fire detection example will produce a decision that there is a Low probability of fire and the other fuzzy values will be automatically ignored. Since the class of maximum methods shows discontinuous output on continuous input, these methods are not considered to be very suitable for use in controllers.

4. Event semantics

Sensors are generally believed to be unreliable and imprecise. Therefore, to increase our confidence in the presence of an event somewhere in the monitored area, we often need readings from multiple sensors and/or readings over some period of time. This could be achieved by instrumenting the event description logic with temporal and spatial semantics. We believe that this can significantly decrease the number of false positives. It will also allow us to describe and detect more complex events. To the best of our knowledge, no previous work on applying fuzzy logic to event detection has considered the effects of temporal and spatial semantics on the accuracy of event detection.

Consider, for example, a fire detecting scenario. A sensor network is deployed to monitor a building and trigger an alarm if a fire starts. There are a number of temperature and smoke sensors in each room, as well as in the hallways. The floors in the building are monitored separately, and there is a master node on each floor. The rest of the sensor nodes send their readings to the master node on their floor. Based on these readings, the master node determines if there is a fire or not. The fire detection is based not only on the temperature and smoke obscuration readings for a particular moment in time, but also on the rate of change of both the temperature and smoke levels. Therefore, our fire detection logic takes four linguistic variables as input – temperature (T), temperature change (ΔT), smoke obscuration (S), and smoke obscuration change (ΔS). The linguistic values for all four variables can be classified as Low (L), Medium (M), and High (H). The accuracy of event detection might be higher if linguistic variables with higher granularity are used, i.e. instead of only holding Low, Medium, or High values, they can also hold values such as Very Low, Low–Medium, Medium–High, and Very High. However, the designer of a WSN-based event detection system should use the smallest number of membership sets that can provide high event detection accuracy, while minimizing the size of the rule-base and the corresponding memory consumption.

In order to increase the accuracy of the fire detection scheme, we require that at least two temperature readings and one smoke reading are used to make a decision. Table 1 shows an example rule-base for this fire detection scenario. This rule-base, however, introduces a number of concerns which we address in the rest of this section.

4.1. Spatial semantics

One of the main goals when designing an event detection system 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 there is an actual fire if more than one node reports high temperature and smoke readings. If, for example, three sensors from the same room send reports indicating fire, the probability that there is an actual fire in that room is very high. In general, there is a negative correlation between the distance among the sensors reporting fire and the probability of this report being true. Therefore, we include the concept of location in the event detection logic. We achieve this by augmenting the rules in the rule-base with a linguistic variable that serves as a spatial guard. This variable expresses the application requirements about the distance between the reporting sensors. In our fire detection scenario, we can name this variable *distance* and classify it as Close

Table 1
An example fire detection rule-base.

Rule #	T_1	ΔT_1	T_2	ΔT_2	S	ΔS	Confidence
1	L	L	L	L	L	L	L
2	L	L	L	L	L	M	L
3	L	L	L	L	L	H	L
4	L	L	L	L	M	L	L
5	L	L	L	L	M	M	L
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
728	H	H	H	H	H	M	H
729	H	H	H	H	H	H	H

(C), Distant (D), and Far (F), for example. Incorporating this *distance* variable into the rule-base, however, changes the format of the rules and adds an extra column to the rule-base. Now the format of the rules in Table 1 changes to:

$$\text{IF } T_1 \text{ is H } \textit{and } \Delta T_1 \text{ is H } \textit{and } T_2 \text{ is H } \textit{and } \Delta T_2 \text{ is H } \textit{and } \text{S is H } \textit{and } \Delta\text{S is H } \textit{and distance} \text{ is F, THEN Fire is M.}$$

4.2. Temporal semantics

To further decrease the number of false alarms, we also need to take into account the temporal properties of the monitored events. The event detection confidence is higher if the sensor readings indicating that a particular event has occurred have been generated within a short period of time of each other. We call the length of this time period *temporal distance* of the readings. The event detection confidence decreases as the temporal distance between the sensor readings increases.

Temporal semantics are especially important for WSNs because of the inherent nature of sensor communication. It is very likely for messages in a WSN to be delayed because of network congestions or routing problems. Consequently, a reliable event detection rule-base should take into consideration the generation times of the sensor readings. To accommodate this, we include another linguistic variable that serves as a temporal guard. This variable, *time*, represents the difference in the generation times of the sensor readings. For example, in our fire detection scenario, *time* could have three semantic values: Short (S), Medium (M), and Long (L). In this way, the information about the time interval within which the sensor readings have been generated is included in the decision process.

5. Decreasing the size of the rule-base

Augmenting the rule-base with temporal and spatial variables increases the number of rules. As mentioned earlier, the size of the rule-base grows exponentially to the number of linguistic variables. In our fire monitoring example, where the only sensor readings we consider are temperature and smoke, the full rule-base has 6561 rules. In more complicated scenarios that require more than two types of sensors, the number of rules in the fuzzy rule-base could be much higher. Storing such rule-bases might be a challenge for memory constrained sensor nodes. In addition, traversing the full rule-base every time there are new sensor readings will slow down the event detection. To address these concerns, we have designed three techniques to help reduce the number of rules. We demonstrate how these techniques are applied on a relatively small rule-base with a few linguistic variables that can take three values – Low, Medium, and High. However, these techniques can be applied in the same fashion to larger rule-bases that contain more linguistic variables characterized by more complex membership functions.

Although the rule-base reduction techniques alleviate both the storage problem and the rule traversal process, they might have a negative effect on the event detection accuracy. Therefore, maintaining high event detection accuracy was a primary goal when designing the reduction techniques described in this section. We achieve this by carefully modifying the rule-base through merging important rules and removing the rules that do not affect the detection accuracy of the events of interest.

Table 2
Rule-base for a temperature sensor.

Rule #	T	ΔT	Confidence
1	L	L	L
2	L	M	L
3	L	H	M
4	M	L	L
5	M	M	M
6	M	H	H
7	H	L	M
8	H	M	H
9	H	H	H

5.1. Separating the rule-base

The first technique we use to reduce the size of the rule-base is to separate the rules on a “need to know” basis. Each node stores only the rules corresponding to the types of sensors it has. If, for example, some of the nodes in our fire detection scenario are only equipped with temperature sensors, they do not need to store the whole rule-base. Instead, they store a smaller modified rule-base similar to the one shown in Table 2. This rule-base contains only rules with premise linguistic variables based on the values from the temperature sensors. In this way, the event detection logic on each node considers only the rules that are relevant to the node’s sensor readings. This separation simplifies the decision process and makes the rule-base traversal faster. The rule-base for the smoke sensors can be constructed in a similar way.

5.2. Combining rules with similar outcomes

Rules 1 and 2 in Table 2 have the same outcome and only differ in the values of ΔT. This observation is also valid for rules 8 and 9. Combining these rule couples could help us further decrease the size of the rule-base. For the rule-base in Table 2 applying such an optimization leaves us with seven rules. The rules, however, have a slightly different syntax. Instead of:

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

some of the rules have the following different form:

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

In the modified rules $\leqslant$ stands for “in this fuzzy set or in fuzzy sets smaller than it” and $\geqslant$ stands for “in this fuzzy set or in fuzzy sets greater than it”. Table 3 shows the result of applying this reduction technique on the rule-base in Table 2.

Table 3
Reduced rule-base for a temperature sensor.

Rule #	T	ΔT	Confidence
1	L	$\leqslant$M	L
2	L	H	M
3	M	L	L
4	M	M	M
5	M	H	H
6	H	L	M
7	H	$\geqslant$M	H

5.3. Incomplete rule-base

A rule-base is considered *complete* if there are rules for every possible combination of the input variables. However, only some of these combinations have outcomes that are important to the event detection system. For example, rules containing variables which do not satisfy the temporal and spatial constraints cannot trigger an alarm. Therefore, the rules with *distance* variable Distant or Far can be removed from the rule-base. This step leaves us with just a third of the original number of rules in the rule-base. Similarly, applying the same approach to the *time* variable and removing the rules with values Medium and Long decreases the rule-base by yet another two thirds.

In addition, if we exclude the rules with consequents that are of no interest to the event detection system, such as rules indicating that the possibility that a fire has occurred is Low, we reduce the size of the rule-base even more. As a result, by lowering the level of completeness of the rule-base, we significantly decrease the number of rules that need to be stored on the sensor nodes. This "trimming" process, however, should be performed very carefully in order to prevent the removal of important consequents. To make sure that the system knows how to proceed if none of the rules in the rule-base has been satisfied, we introduce a *default* rule that is triggered if no other rule has been satisfied.

6. Evaluation

We use the FuzzyJ Toolkit for Java [37] to implement the necessary fuzzy logic functionality. To avoid the danger, cost, and non-repeatability of creating fires, we perform trace-based simulations using real fire data publicly available on the National Institute of Standards and Technology (NIST) website [38]. The study they conduct provides sensor measurements from a number of different real fires as well as nuisance scenarios. We have used three of the available real fire scenarios: fire caused by a burning mattress, fire caused by a burning chair, and cooking oil fire. The purpose of the nuisance tests is to study common household nuisance alarm scenarios. We have used two of these tests in our experiments: frying margarine and broiling hamburgers.

6.1. Experiments using real fire data

The membership functions for the smoke and temperature input linguistic variables used in the experiments are shown in Fig. 3. In addition to the temperature and smoke obscuration variables, we also take into consideration the temperature and smoke obscuration difference between two consecutive readings. These two additional variables give us a notion of how fast the temperature and smoke obscuration are changing. Fig. 4 shows the membership function for the output fire confidence. This linguistic variable represents the system's confidence in the presence of fire. For example, if the fire confidence value is higher than 80, we are more than 80% certain that there is a fire. If the fire confidence is smaller than 50, it is more likely that there is no fire.

In the system model we use for our simulations every node decides locally if a fire event has occurred. If it decides that a fire is present, a node forwards its decision to the master node for the house. An alternative system model, where the nodes send a subset of their readings to the master node, and the master node makes a decision, is also possible. In this model, the base node has the aggregated information from all sensors, and might be able to make more accurate decisions. However, because of the increased amount of communication, the lifetime of the network might decrease. Therefore, which model is appropriate depends on the nature of the application and the lifetime requirements of the network.

To provide a baseline for our results, we performed crisp-value experiments with the burning mattress, burning chair, and cooking oil data. The temperature and smoke obscuration thresholds used in the crisp logic experiments are threshold values used in commercial smoke and heat detectors, 55 °C and 0.15 m^{-1}, respectively [39,40]. The membership functions in Fig. 3 were also built according to these threshold values. We used the commercial crisp thresholds as the border between Low and High, which in our scenario is classified as 0% Low, 100% Medium, and 0% High for all four linguistic variables. We relied on domain knowledge to determine the remaining details of the membership functions.

The results from the crisp-value experiment are shown in Figs. 5a, 6a, and 7a. In these and all following figures, the origin of the graph represents the time of fire ignition. As we can see from the three figures, using crisp values resulted in a very large number of false fire detections. In the burning mattress scenario in particular, there were 40 false fire detections in the period prior to the fire ignition, which constitutes about 1.3% of the measurements. This considerable number or false positives significantly affects the efficiency and fidelity of an event detection system. Admittedly, part of these false positives can be attributed to the aggressive crisp value thresholds. However, if the thresholds are set higher, this could lead to failures in detecting actual fires. In a real fire detection system it is more important to decrease the number of false negatives than that of false positives. Therefore, we have kept the threshold values in compliance with the commercial standards.

What we wanted to investigate with our next set of experiments was whether fuzzy logic can do better in terms of false positives, while still reporting promptly the presence of a fire when one actually occurs. In the first set of fuzzy logic experiments, a node decides if there is a fire based

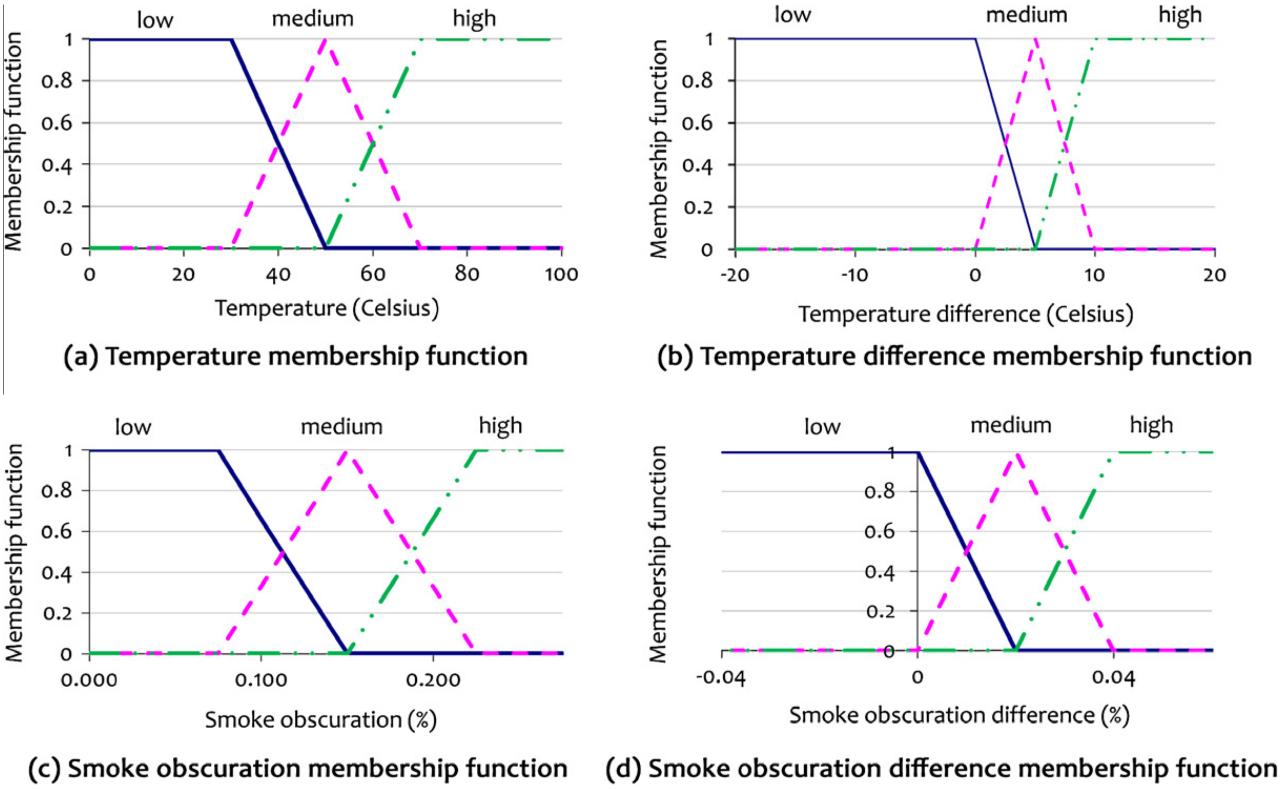

Fig. 3. Membership functions for the input linguistic variables.

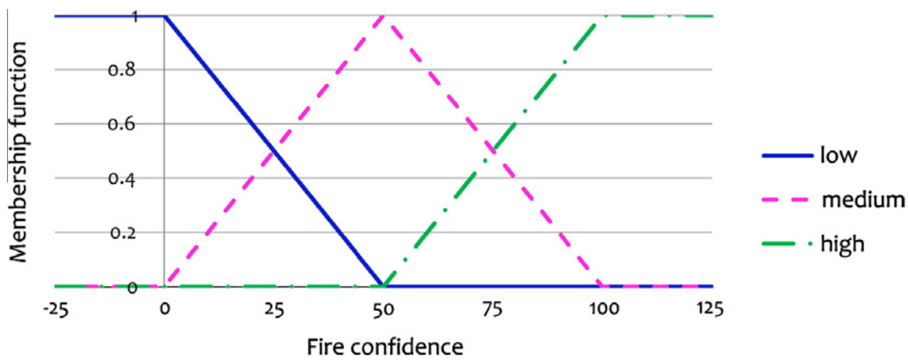

Fig. 4. Fire confidence membership function.

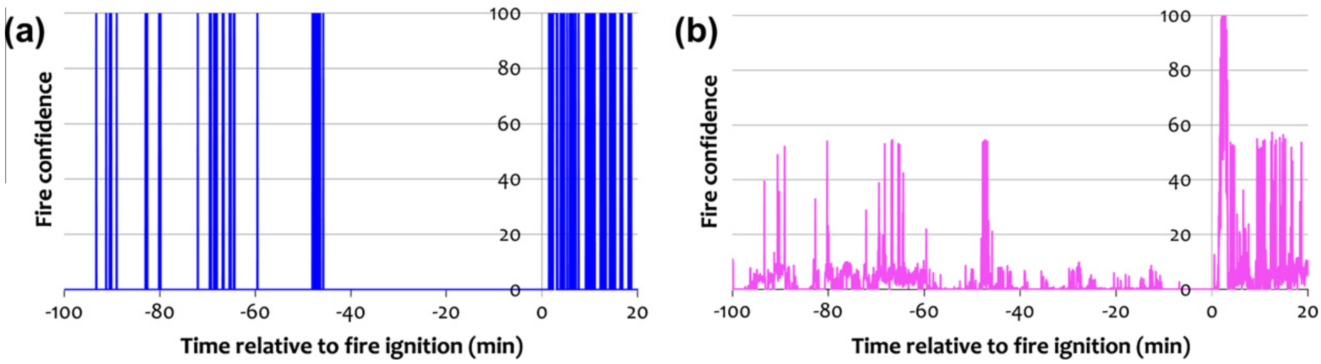

Fig. 5. Burning mattress simulation: (a) crisp value detection and (b) fuzzy value detection.

only on its own readings. The readings of neighboring sensor nodes are not considered as inputs to the decision process.

The values of the linguistic variables used in the decision process can be classified as Low (L), Medium (M), and High

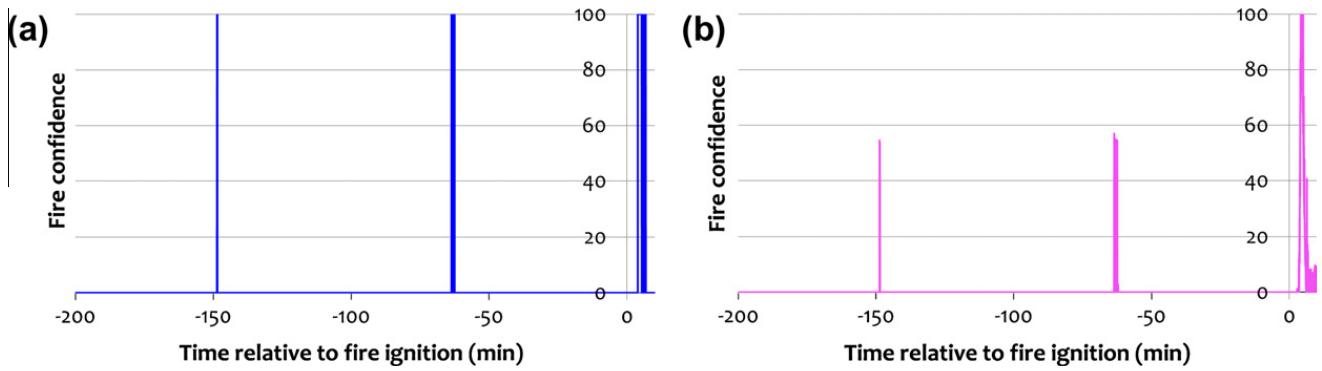

Fig. 6. Burning chair simulation: (a) crisp value detection and (b) fuzzy value detection.

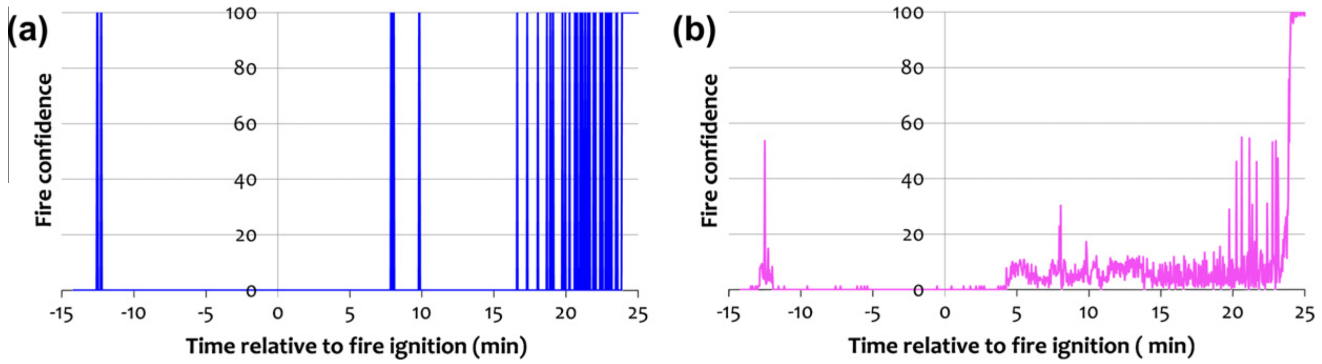

Fig. 7. Burning oil simulation: (a) crisp value detection and (b) fuzzy value detection.

(H), as shown by Figs. 3 and 4. We have used heuristics to build the rule-base for our fire detection experiments. In cases where this is not possible, for example, when more complex events are to be detected, domain experts could be consulted for the definition of the rule-bases. The rule-base for these experiments is shown in Table 4. Because of space limitations, instead of showing the complete rule-base, which has 81 rules, we show the rule-base after our second reduction technique has been applied.

The results from our first set of fuzzy logic experiments, a burning mattress, a burning chair, and cooking oil fire, are presented in Figs. 5b, 6b, and 7b, respectively. As we can see, the fuzzy logic event detection mechanism performs very well. It detects the presence of a fire shortly after the ignition. In addition, unlike the crisp-value fire detection, there are no false positives. All three graphs show fire confidence around 0 before the ignition, except for a number of small peaks when the confidence increases to 54, which is close to 100% Medium. At the same times when the fuzzy-value peaks occur, we can also notice crisp-value peaks but with much higher confidence. The raw sensor data revealed that the peaks were caused by a number of one-second-long reports of increased smoke values. This proves our hypothesis that fuzzy logic is able to accommodate the often imprecise sensor readings. Even in the cases when the nodes erroneously report the presence of smoke, the fuzzy logic mechanism keeps the fire confidence low enough so that a false alarm is not triggered.

We also evaluate how including neighbor node values in the decision process affects the detection accuracy. The average of the neighbor values is represented with an additional linguistic variable that we include in the decision rules. In addition, in order to meet the spatial and temporal requirements of the application, we only consider readings (i) received from neighbor nodes that are located close to the current node and (i) that have been generated within 1 s from the current reading of the node. The results in Figs. 8–10 show that fire is detected almost as quickly as when the decision process is only based on *own* sensor readings. Although the peak areas are still present, the corresponding fire confidence values are lower when the neighbor readings are included in the decision process. This shows that including the readings of neighbor nodes in the decision process positively affects the detection accuracy.

Fig. 10 allows us to make an important observation. In the burning oil scenario, when the fire detection is based on the readings of a single sensor, the system reaches fire confidence of 100 around 23 min after the stove has been turned on. However, when the readings of neighbor sensors are considered in the detection process, the maximum fire confidence never exceeds 71, which is approximately 60% Medium and 40% High. This is due to the fact that the neighbor sensors are located further away from the fire and, therefore, their temperature readings have lower values. These results come to show that sensor network designers should be careful when determining the size and radius of a sensor's neighborhood. Although including readings from neighbor sensors improves the event detection accuracy of the system, when these neighbor sensors

Table 4
Fire detection rule-base for the scenario where a node decides if there is a fire based only on its own sensor readings. The temperature, temperature difference, smoke, and smoke difference variables take Low (L), Medium (M), and High (H) values.

Rule	Temperature	Δ Temperature	Smoke	Δ Smoke	Confidence
1	L	L	⩽M	⩾L	L
2	L	L	H	⩽M	L
3	L	L	H	H	M
4	L	⩾M	L	L	L
5	L	H	L	M	L
6	M	L	L	⩾L	L
7	M	L	M	L	L
8	H	L	L	L	L
9	L	M	L	⩾M	M
10	L	M	M	⩾L	M
11	L	M	H	⩽M	M
12	L	H	L	H	M
13	L	H	⩾M	L	M
14	L	H	M	M	M
15	M	L	M	⩾M	M
16	M	L	H	⩽M	M
17	M	M	⩽M	⩽M	M
18	M	M	H	L	M
19	M	H	⩽M	L	M
20	H	⩽M	L	M	M
21	H	⩾M	L	L	M
22	L	M	H	H	H
23	L	H	M	H	H
24	L	H	H	⩾M	H
25	M	L	H	H	H
26	M	M	⩾L	H	H
27	M	M	H	M	H
28	M	H	⩾L	⩾M	H
29	M	H	H	L	H
30	H	⩾L	L	H	H
31	H	⩾L	⩾M	⩾L	H
32	H	H	L	M	H

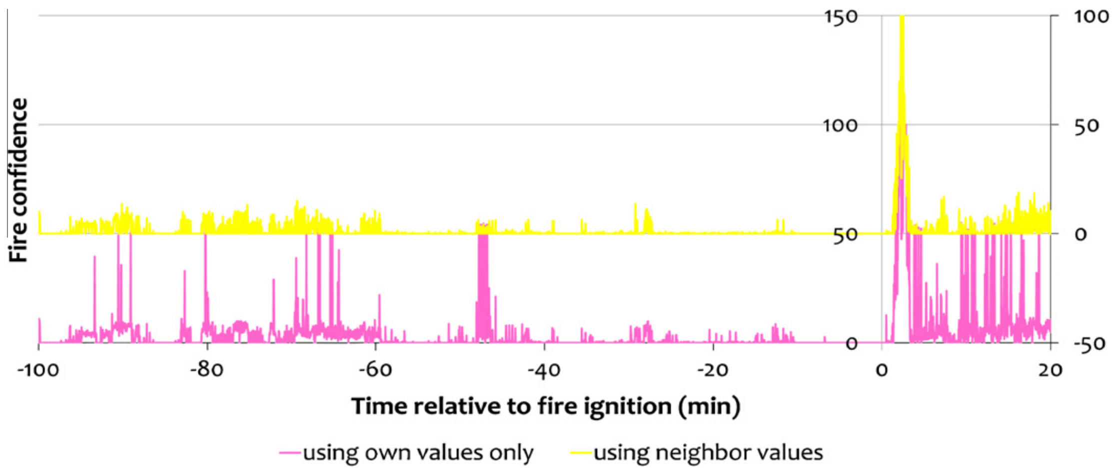

Fig. 8. Simulating a burning mattress: including neighbor readings in the decision. The results when only own values are used are plotted on the first y-axis. Including the neighbor values is plotted on the second y-axis.

are located too far from where the event has occurred, this might have a negative effect on the detection accuracy.

6.2. Experiments using nuisance fire data

The goal of these experiments is to study the behavior of our fuzzy-value fire detection mechanism when it is presented with a nuisance scenario. The Smoke Detector Operability Survey: Report on Finding [41] conducted by the US Consumer Products Safety Commission reported that about 50% of the 1012 participants indicated that they had experienced nuisance alarms, with 80% of those attributed to cooking activities, and an additional 6% citing steam from bathrooms. Dust and tobacco smoke are also mentioned sources. The survey also reported that for the alarms with missing or disconnected batteries, or

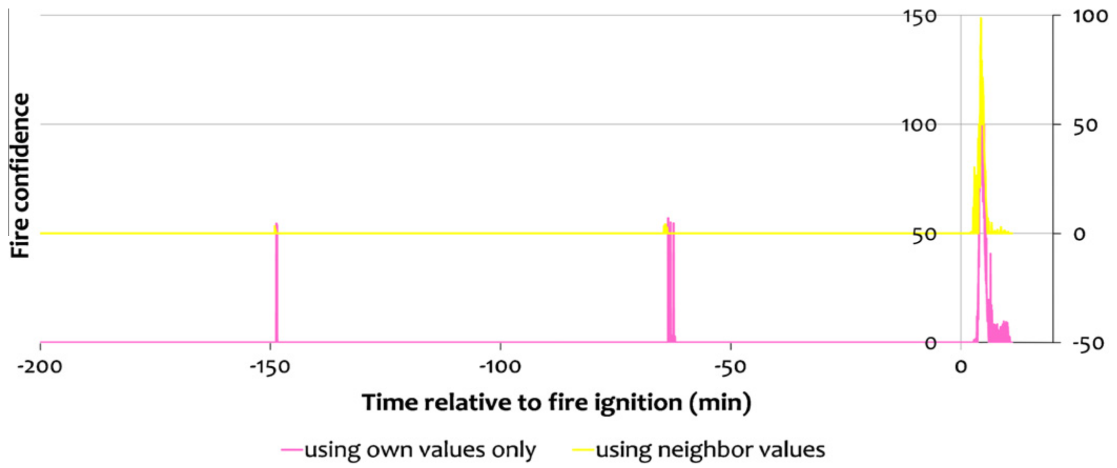

Fig. 9. Simulating a burning chair: including neighbor readings in the decision. The results when only own values are used are plotted on the first *y*-axis. Including the neighbor values is plotted on the second *y*-axis.

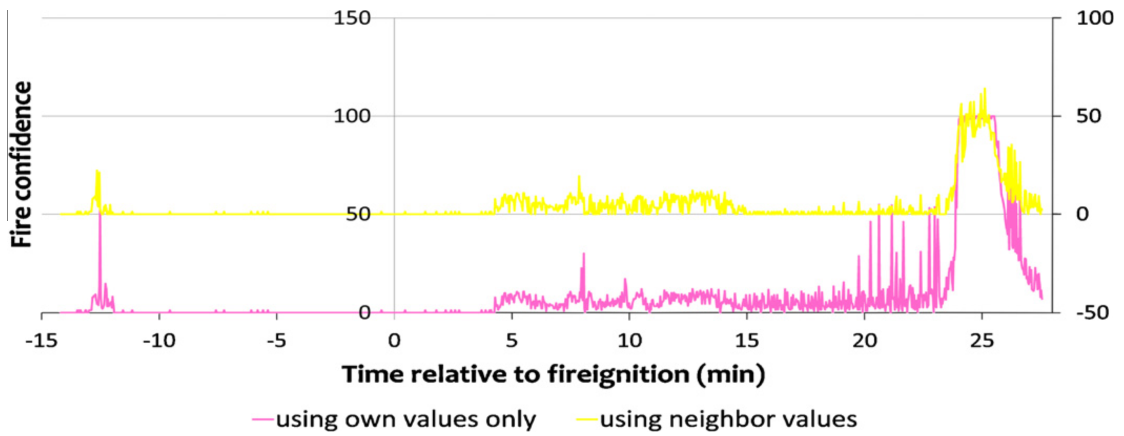

Fig. 10. Simulating burning oil: including neighbor readings in the decision. The results when only own values are used are plotted on the first *y*-axis. Including the neighbor values is plotted on the second *y*-axis.

disconnected AC power, more than one third of respondents indicated that power was removed due to nuisance alarms.

The NIST data provided for the nuisance scenarios differs from that for the actual fires in that the smoke obscuration measurements are not provided. Therefore, we have substituted the two input linguistic variables based on smoke obscuration with two new variables based on aerosol mass concentration. Similarly to the smoke obscuration, the aerosol mass concentration allows us to determine the amount of aerosol particles in the air. The thresholds for the mass concentration and mass concentration difference linguistic variables were chosen based on previous mass concentration alarm research [42]. Fig. 11 shows the membership functions for the two new linguistic variables.

As with the real fire scenarios, we use crisp-value detection as a baseline. Fig. 12 shows the results from the crisp-value experiments of frying margarine and grilling hamburgers, respectively. In both scenarios no actual fire occurred. However, as we can see from the figure, the number of false fire detections is high: 172 (34% of all readings) for the frying margarine scenario and 248 (16% of all readings) in the broiling hamburgers scenario.

The results from the fuzzy-value experiments are shown in Fig. 13. For both scenarios the fire detection confidence follows the same pattern as in the crisp-value experiments. The peaks that are present in Fig. 13a and 13b are also present in Fig. 12a and 12b, respectively. Similarly to the real fire experiments, the difference between the peaks is that in the fuzzy-value scenarios the peaks never reach high confidence. This means that, unlike the cases when crisp values are used, an alarm will not be triggered.

An interesting observation is that in Fig. 13a some of the fire confidence peaks reach levels as high as 50%. All of these peaks are grouped around the fifth minute of the experiment. At that time, the aerosol mass concentration increases above 100 mg/m^3 with maximum 214 mg/m^3. This is the time when the frying caused the highest level of smoke. However, despite these high mass concentration values, the fuzzy logic system manages to determine that no fire is currently present.

6.3. Analysis

6.3.1. Why does fuzzy logic perform better?

An interesting question is why fuzzy logic is more precise than crisp-value logic. From the considerable decrease

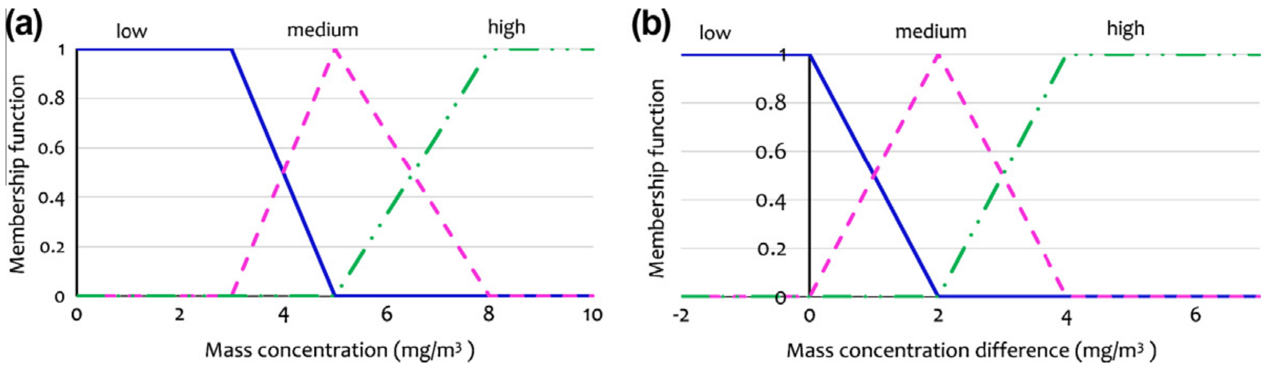

Fig. 11. Membership functions for the mass concentration input linguistic variables.

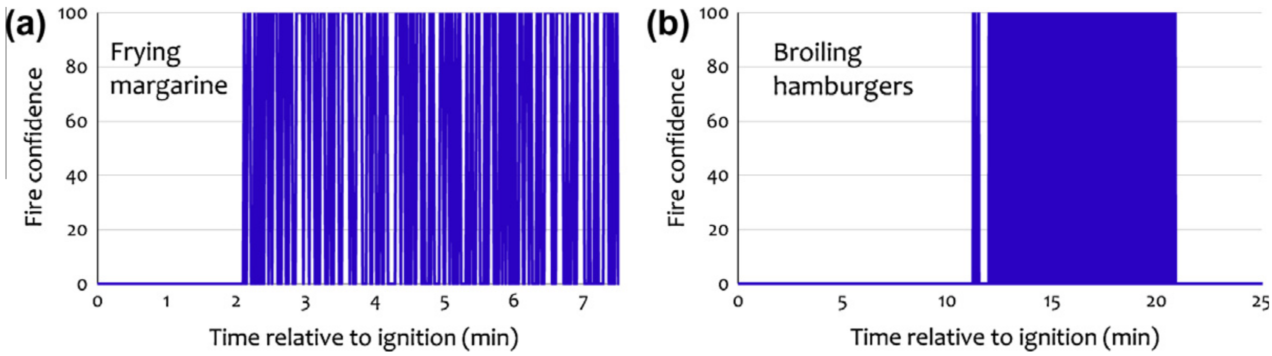

Fig. 12. Crisp value simulation: (a) frying margarine and (b) broiling hamburgers.

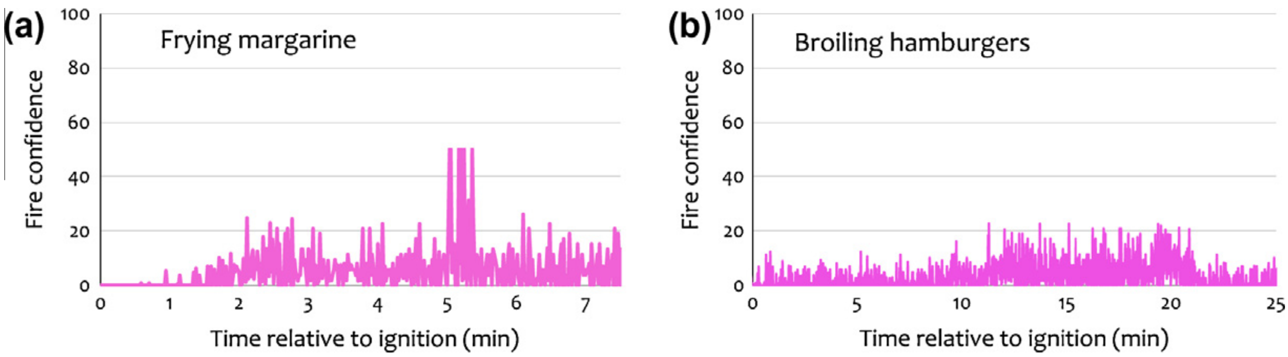

Fig. 13. Fuzzy value simulation: (a) frying margarine and (b) broiling hamburgers.

in the number of false positives, it appears that fuzzy logic handles the fluctuating sensor readings much better. To understand why this happens we take a closer look at the first false fire detection reported by the crisp-value logic. In the burning mattress scenario this occurs approximately 12 min into the experiment. The values that cause the false alarm are: $T = 25.21$ °C, $\Delta T = 0$ °C, $S = 0.203\%$, and $\Delta S = 0.109\%$. Since the smoke level (S) and the smoke change level (ΔS) are both classified as High, the crisp logic concludes that there must be a fire.

What does the fuzzy logic event detection do differently? According to the membership functions in Fig. 3, temperature value of 25.21 °C is classified as 100% Low; temperature change of 0 °C is classified as 100% Low; smoke obscuration level of 0.203% is classified as 33% Medium and 66% High; and smoke obscuration change of 0.119% is classified as 100% High. The decision making process checks which rules from the rule-base are satisfied. These are rules 1 and 3 from the rule-base in Table 4. Based on those rules, the defuzzifier reports a fire confidence value of 39.4. This value maps to fire confidence which is 20% Low and 80% Medium. Such level of confidence, however, is not enough to cause the system to report a fire.

This example illustrates why a fuzzy logic event detection system tends to perform better than a crisp one in the presence of short-lasting inaccurate sensor readings, which often occur in WSNs. Fuzzy logic takes into account the certainty with which an event occurs, instead of making binary decisions based on crisp values and fixed thresholds, which improves the accuracy of event detection.

6.3.2. Decreasing the rule-base

We applied our reduction techniques to the full version of the rule-base shown in Table 4. All nodes in the

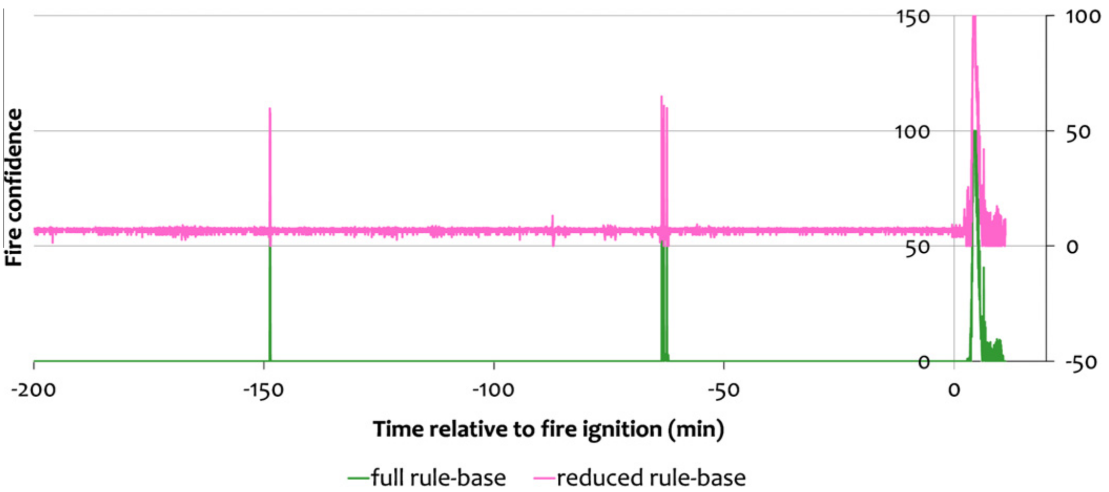

Fig. 14. Simulating a burning chair with a reduced rule-base. The results when the full rule-base is used are plotted on the first *y*-axis. Using the reduced rule-base is plotted on the second *y*-axis.

simulation are equipped with both a smoke and a temperature sensor which makes the first technique not applicable. Therefore, we only used the second and third reduction techniques. The rule-base initially has 81 rules. Combining the rules with similar outcomes reduces the number of rules to 32, as shown in Table 4. This evaluates to a decrease of 60%. In general, when there are more than two input linguistic variables, applying the second method decreases the rule-base by approximately two thirds. Excluding the rules that result in *Low* fire confidence additionally reduces the size of the rule-base to 25, which is 31% of the original rule-base.

We have compared the behavior of the fire detection system when the full and the reduced rule-bases are used. Fig. 14 shows the results for the burning chair scenario. The fire confidence is consistently higher when the reduced rule-base is used. However, since this confidence remains Low, this does not cause false fire detections. For future work, we plan to perform more detailed analysis of the memory requirements associated with using fuzzy logic.

6.3.3. Detection accuracy

To further understand the behavior of our fuzzy logic approach, we have compared it to two well established classification algorithms: a naive Bayes classifier [43] and a J48 decision tree which is an open source implementation of the C4.5 algorithm [44]. Fuzzy logic is more suitable than these two algorithms for WSN event description since, unlike Bayes classifiers and decision trees where values are considered to be discrete, it works with continuous values, which is exactly what the sensor readings are. In addition, specifying the membership functions is simpler and computationally more efficient than building a probability model.

We ran this set of experiments using the Weka data mining tool [45]. The input values to the classification algorithms were the same as the ones used in the fuzzy logic experiments – temperature, temperature difference, smoke obscuration, and smoke obscuration difference. We performed a 10-fold cross validation for both classification algorithms. Table 5 shows the number of incorrectly classified instances for the first two fire scenarios, burning mattress and burning chair, as well as what percentage of the total instances was incorrectly classified. Both algorithms produce a number of inaccurate classifications. Although the percentage of the erroneously classified instances is low, it is higher than the number of misclassifications introduced by fuzzy logic.

Table 5
Number of incorrect classifications by a Naive Bayes classifier and a J48 tree.

	Naive Bayes		J48 decision tree		Fuzzy logic	
	Number	Percent (%)	Number	Percent (%)	Number	Percent
Burning chair	105	1.56	7	0.13	0	0
Burning mattress	89	2.35	5	0.13	0	0

Table 6
Fire detection delay in seconds.

Scenario	Crisp values	Fuzzy values	Plus neighbor readings	Reduced readings
Burning chair	236	236	248	236
Burning mattress	103	97	117	97
Cooking oil fire	1431	1431	1443	1431

6.3.4. Fire detection delay

Table 6 shows the delay incurred by the different fire detection mechanisms. Fire is detected just as fast, and in the burning mattress scenario even faster, when fuzzy values are used. In addition, decreasing the size of the rule-base does not delay the fire detection. We also notice that including the readings of neighbor sensors in the decision process slightly slows down the detection. This is not surprising since not all sensors are located at the same distance from the fire, and, therefore, they start registering abnormal values at different times. Consequently, if a sensor is waiting for its neighbors to also detect the fire, and those neighbors are located further away from the fire source, the detection might be slightly delayed.

7. Conclusions and future work

A disadvantage of the current event detection approaches used in WSNs is that they cannot properly handle the often imprecise sensor readings. In this paper we show that fuzzy logic is a powerful and accurate mechanism which can successfully be applied not only to fire detection but to any event detection sensor network application. Compared to using crisp values, fuzzy logic maintains a high accuracy level despite fluctuations in the sensor values. This helps decrease the number of false positives, while still providing fast and accurate event detection. Our experiments support the hypothesis that incorporating the readings of neighbor nodes in the decision process further improves the event detection accuracy.

The evaluation also shows that the rule-base reduction techniques we have developed are efficient and preserve both the correctness and the timeliness of event detection. Using two of these techniques, namely, *combining rules with similar outcomes* and *incomplete rule-base*, reduces the size of our experimental rule-base by more than 70%. Further, compared to two well-established classification algorithms, fuzzy logic provides more accurate event detection.

For future work we plan to perform experiments on a sensor testbed. This will allow us to better evaluate how using fuzzy logic influences the accuracy and speed of event detection when the decision logic is run on sensor nodes. In addition, it will help us study the effect of applying temporal constraints on the accuracy of event detection.

References

[1] Cornell Database Group-Cougar, 2010. <http://www.cs.cornell.edu/bigreddata/cougar/>.

[2] R. Govindan, J. Hellerstein, W. Hong, S. Madden, M. Franklin, S. Shenker, The sensor network as a database, Technical Report 02-771, Computer Science Department, University of Southern California, 2002.

[3] S. Madden, M. Franklin, J. Hellerstein, W. Hong, The design of an acquisitional query processor for sensor networks, in: SIGMOD, ACM, New York, NY, USA, 2003, pp. 491–502.

[4] S. Li, S.H. Son, J.A. Stankovic, Event detection services using data service middleware in distributed sensor networks, in: Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN), Springer-Verlag, Berlin, Heidelberg, 2003, pp. 502–517.

[5] Michael Franklin, Declarative interfaces to sensor networks, Presentation at NSF Sensor Workshop, 2004.

[6] B. Jiao, S. Son, J. Stankovic, GEM: Generic Event Service Middleware for Wireless Sensor Networks, in: INSS, Institute of Electrical and Electronics Engineers (IEEE), USA, 2005.

[7] K. Kapitanova, S. Son, MEDAL: a compact event description and analysis language for wireless sensor networks, in: Networked Sensing Systems (INSS), 2009 Sixth International Conference on, Institute of Electrical and Electronics Engineers (IEEE), USA, 2009.

[8] E. Tapia, S. Intille, K. Larson, Activity recognition in the home using simple and ubiquitous sensors, in: Pervasive Computing, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 158–175.

[9] C. Wren, E. Tapia, Toward scalable activity recognition for sensor networks, Location Context-Awareness (LoCA) (2006) 168–185.

[10] P. Castro, P. Chiu, T. Kremenek, R.R. Muntz, A probabilistic room location service for wireless networked environments, in: Proceedings of the 3rd International Conference on Ubiquitous Computing, Springer-Verlag, London, UK, 2001, pp. 18–34.

[11] M. Duarte, Y. -H Hu, Distance based decision fusion in a distributed wireless sensor network, in: Proceedings of the 2nd International Conference on Information Processing in Sensor Networks, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 392–404.

[12] T.M. Chen, V. Venkataramanan, Dempster–Shafer theory for intrusion detection in ad hoc networks, IEEE Internet Comput. 9 (2005) 35–41.

[13] H. Wu, M. Siegel, R. Stiefelhagen, J. Yang, Sensor fusion using Dempster–Shafer theory, in: Proceedings of IEEE Instrumentation and Measurement Technology Conference (IMTC), Institute of Electrical and Electronics Engineers (IEEE), USA, 2002, pp. 21–23.

[14] R. Murphy, Dempster–Shafer theory for sensor fusion in autonomous mobilerobots, IEEE Trans. Robot. Automat. (1998) 197–206.

[15] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, ALARM-NET: wireless sensor networks for assisted-living and residential monitoring, Technical Report CS-2006-13, University of Virginia, 2006.

[16] D. Lymberopoulos, A.S. Ogale, A. Savvides, Y. Aloimonos, A sensory grammar for inferring behaviors in sensor networks, in: Proceedings of the 5th International Conference on Information Processing in Sensor Networks, ACM, New York, NY, USA, 2006, pp. 251–259.

[17] H. Ghasemzadeh, J. Barnes, E. Guenterberg, R. Jafari, A phonological expression for physical movement monitoring in body sensor networks, in: Mobile Ad Hoc and Sensor Systems, 2008, MASS 2008, 5th IEEE International Conference on, Institute of Electrical and Electronics Engineers (IEEE), USA, 2008, pp. 58–68.

[18] O. Amft, M. Kusserow, G. Tröster, Probabilistic parsing of dietary activity events, BSN (2007) 242–247.

[19] I. Gupta, D. Riordan, S. Sampalli, Cluster-head election using fuzzy logic for wireless sensor networks, in: Proceedings of the 3rd Annual Communication Networks and Services Research Conference, IEEE Computer Society, Washington, DC, USA, 2005, pp. 255–260.

[20] J. Kim, S. Park, Y. Han, T. Chung, CHEF: cluster head election mechanism using fuzzy logic in wireless sensor networks, ICACT (2008) 654–659.

[21] H.Y. Lee, T.H. Cho, Fuzzy logic based key disseminating in ubiquitous sensor networks, in: 10th International Conference on Advanced Communication Technology (ICACT), Institute of Electrical and Electronics Engineers (IEEE), USA, 2008, pp. 958–962.

[22] B. Kim, H. Lee, T. Cho, Fuzzy key dissemination limiting method for the dynamic filtering-based sensor networks, ICIC (2007) 263–272.

[23] B. Lazzerini, F. Marcelloni, M. Vecchio, S. Croce, E. Monaldi, A fuzzy approach to data aggregation to reduce power consumption in wireless sensor networks, in: Fuzzy Information Processing Society, 2006, NAFIPS 2006, Annual Meeting of the North American, Institute of Electrical and Electronics Engineers (IEEE), USA, 2006, pp. 436–441.

[24] J.M. Kim, T.H. Cho, Routing path generation for reliable transmission in sensor networks using ga with fuzzy logic based fitness function, in: Proceedings of the 2007 International Conference on Computational Science and Its Applications – Volume Part III, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 637–648.

[25] S.-Y. Chiang, J.-L. Wang, Routing analysis using fuzzy logic systems in wireless sensor networks, KES (2008) 966–973.

[26] Q. Ren, Q. Liang, Fuzzy logic-optimized secure media access control (FSMAC) protocol wireless sensor networks, in: Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS), Institute of Electrical and Electronics Engineers (IEEE), USA, 2005, pp. 37–43.

[27] S.A. Munir, Y.W. Bin, R. Biao, M. Jian, Fuzzy logic based congestion estimation for QoS in wireless sensor network, in: Wireless Communications and Networking Conference (WCNC), Institute of Electrical and Electronics Engineers (IEEE), USA, 2007, pp. 4336–4341.

[28] F. Xia, W. Zhao, Y. Sun, Y.-C. Tian, Fuzzy logic control based QoS management in wireless sensor/actuator networks, Sensors (2007) 3179–3191.
[29] Q. Liang, L. Wang, Event detection in wireless sensor networks using fuzzy logic system, in: Computational Intelligence for Homeland Security and Personal Safety (CIHSPS), Institute of Electrical and Electronics Engineers (IEEE), USA, 2005, pp. 52–55.
[30] M. Marin-Perianu, P. Havinga, D-FLER: a distributed fuzzy logic engine for rule-based wireless sensor networks, in: Proceedings of the 4th International Conference on Ubiquitous Computing Systems (UCS), Springer-Verlag, Berlin, Heidelberg, 2007, pp. 86–101.
[31] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Syst Man Cybernet. SMC 3 (1) (1973) 28–44.
[32] S.-i. Horikawa, T. Furuhashi, Y. Uchikawa, On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, IEEE Trans. Neural Networks 3 (5) (1992) 801–806.
[33] J.-S.R. Jang, Self-learning fuzzy controllers based on temporal backpropagation, IEEE Trans. Neural Networks 3 (5) (1992) 714–723.
[34] A. Arslan, M. Kaya, Determination of fuzzy logic membership functions using genetic algorithms, Fuzzy Sets Syst. 118 (2) (2001) 297–306.
[35] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1995.
[36] Ch. Schmid, Course on Dynamics of Multidisplicinary and Controlled Systems, 2005. <http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/main.html>.
[37] G. of Canada. National Research Council Canada, Institute for Information Technology, FuzzyJ ToolKit from the Java(tm) Platform & FuzzyJess – IIT-ITI- NRC-CNRC, 2011. <http://www.nrc-cnrc.gc.ca/eng/ibp/iit/past-projects/fuzzyj-toolkit.html>.
[38] Building, Fire Research Laboratory, 2009. <http://smokealarm.nist.gov>.
[39] Home Security Store Inc., WS4916 Series Wireless Smoke Detector, 2005. <http://www.homesecuritystore.com/downloadmanual.aspx?mnl_id=496&prod_id=992>.
[40] J. Geiman, D.T. Gottuk, Alarm thresholds for smoke detector modeling. Fire Saf. Sci. 7 (2003) 197–208, doi:10.3801/IAFSS.FSS.7-197.
[41] C. Smith, Smoke detector operability survey: report on findings, US Consumer Product Safety Commission, 1994.
[42] C.D. Litton, Laboratory evaluation of smoke detectors for use in underground mines, Fire Safety J. 44 (3) (2009) 387–393.
[43] D.D. Lewis, Naive (Bayes) at forty: the independence assumption in information retrieval, in: Proceedings of the 10th European Conference on Machine Learning, Springer-Verlag, London, UK, 1998, pp. 4–15.
[44] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
[45] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (2009) 10–18.

Krasimira Kapitanova is a Ph.D. student in Computer Science at the University of Virginia. She received her B.S. degree in Computer Science and Technologies from Technical University Sofia, Bulgaria, and an M.C.S. degree from the University of Virginia. Her research interests include formal event description in sensor networks, QoS management, and information management and security.

Sang Hyuk Son is a Professor at the Department of Computer Science of University of Virginia. He received the B.S. degree in electronics engineering from Seoul National University, M.S. degree from KAIST, and the Ph.D. in computer science from University of Maryland, College Park, in 1986. He is on the executive board of the IEEE Technical Committee on Real-Time Systems, for which he served as the Chair during 2007–2008. He is currently serving as an Associate Editor for IEEE Transactions on Computers and Real-Time Systems Journal. His research interests include real-time and embedded systems, database and data services, QoS management, wireless sensor networks, and information security.

Kyoung-Don Kang is an Associate Professor in the Department of Computer Science at the State University of New York at Binghamton. He received his Ph.D. from the University of Virginia in 2003. His research interests include real-time data services, wireless sensor networks, and security and privacy.