Event Detection in Wireless Sensor Networks – Can Fuzzy Values Be Accurate?*

Krasimira Kapitanova[1], Sang H. Son[1], and Kyoung-Don Kang[2]

[1] University of Virginia, Charlottesville VA, USA
krasi@cs.virginia.edu, son@cs.virginia.edu
[2] Binghamton University, Binghamton, NY, USA
kang@cs.binghamton.edu

Abstract. Event detection is a central component in numerous wireless sensor network (WSN) applications. In spite of this, the area of event description has not received enough attention. The majority of current event description 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 detection precision than a couple of well established classification algorithms.

A disadvantage of using fuzzy logic is the exponentially growing size of the rule-base. Sensor nodes have limited memory and 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 level of event detection accuracy.

Keywords: wireless sensor networks, fuzzy logic, event description, event detection accuracy.

1 Introduction

Event detection is one of the main components in numerous wireless sensor network (WSN) applications. 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. But just like 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 in WSNs, however, has not been explored much.

* This research work was supported by KOSEF WCU Project R33-2009-000-10110-0.

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 cooling 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 and 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 which can be classified as a false negative. The situation becomes even more convoluted 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 better address the problems that are challenging for crisp logic.

Fuzzy logic has a number of properties that make it suitable for describing WSN events: (i) it can tolerate the unreliable and imprecise sensor readings; (ii) 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%; (iii) 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 full 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 affect the event detection accuracy.

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 significantly compromising the accuracy of event detection.

2 Related Work

Event detection: Not much research has focused directly on providing methods for event description in WSNs that can support data dependency and collaborative decision making. The prevailing approach is to use SQL-like primitives [1,2,3,4]. The papers that employ this method vary in semantics. In [1] and [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 [4]. Li et al. define events using a sub-event list and confidence functions in SQL [3]. 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 support 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 design Petri nets that specify event logic. Petri nets were also used in MEDAL [7], an extension of SNEDL that supports the description of additional WSN specific features such as communication and actuation. Both SNEDL and MEDAL, however, use crisp values in the definitions of their Petri nets.

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].

Fuzzy logic: Fuzzy sets and logic were introduced by L. Zadeh in 1965. Ever since then, numerous fields have taken advantage of their properties. 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 et al. [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 precision 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 seconds after the fire ignition, the authors do not analyze the number of false alarms raised by D-FLER.

3 Overview of Fuzzy Logic

The structure of a general fuzzy logic system (FLS) is shown on Figure 1. First, 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 Cold and Warm. Second, 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 them in more detail in the following subsections.

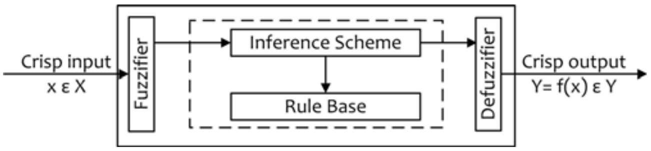

Fig. 1. The structure of a fuzzy logic system

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. Figure 2 shows an example of a temperature membership function. According to this membership function a temperature of -2℃ 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.

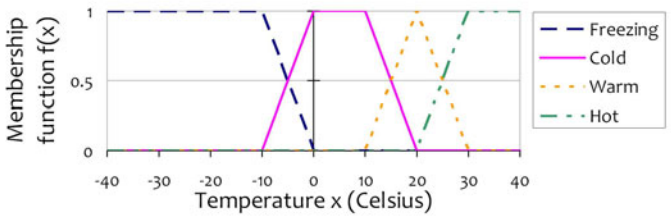

Fig. 2. Temperature membership function

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. Consider a t-input 1-output FLS with rules of the form:

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

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

$$\mu_{S_1^i}\left(x_1^{'}\right) * \mu_{S_2^i}\left(x_2^{'}\right) * ... * \mu_{S_t^i}\left(x_t^{'}\right) = T_{l=1}^{t}\mu_{S_l^i}\left(x_l^{'}\right)$$

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 [32].

3.3 Defuzzification

Executing the rules in the rule-base generates multiple shapes representing the modified membership functions. For example, rules designed to decide the probability of having a fire in a building 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* [32].

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

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
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
729	H	H	H	H	H	H	H

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 monitoring of the building is divided into floors and there is a master node on each floor. The rest of the sensor nodes on the floor send their readings to the master node and, based on these readings, it 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). 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 low. 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 is reporting 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 need to include the concept of location in the event detection logic. This can be achieved by augmenting the rules in the rule-base with a linguistic variable that would serve as a spatial guard. This variable would express the application requirements about the distance between the reporting sensors. For our fire detection scenario, we could call this variable *distance* and classify it as *Close* (C), *Distant* (D), and *Far* (F), for example. Incorporating the *distance* variable into the rule-base adds an extra column for *distance* which changes the format of the rules. Now we have rules such as:

IF T_1 is H and ΔT_1 is H and T_2 is H and ΔT_2 is H and S is H and ΔS is H and distance 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 confidence of event detection is higher if the temporal distance between the sensor readings is shorter and vice versa. Adding temporal semantics is especially important for WSNs because of the nature of sensor communication. It is very possible for messages in a WSN to be delayed because of network congestions or bad routing. Consequently, a reliable event detection rule-base should take into consideration the generation time 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 period 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 extended rule-base will have 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 the 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 us reduce the number of rules. Although such rule-base reduction techniques alleviate both the storage problem and the rule traversal process, they could also introduce a tradeoff between the time and space cost of the rule-base and the event detection accuracy. Therefore, maintaining high event detection accuracy was a key goal when designing the reduction techniques described in this section.

5.1 Separating the Rule-Base

The first thing we could do 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

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

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 only considers rules that are relevant to that node's sensor readings. This separation simplifies the decision process and makes the rule-base traversal faster. The smoke sensors' rule-base 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 on Table 2 applying such an optimization leaves us with 7 rules. The rules, however, have a slightly different syntax. Instead of:

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

some of the rules have the following different form:

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

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

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, only sensor readings that satisfy the temporal and spatial constraints can satisfy rules that could trigger an alarm. Therefore, the rules with *distance* variable Distant

Table 3. Reduced rule-base for a temperature sensor

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

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, e.g. rules that indicate that there is no fire, we will 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 should be stored by 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

6.1 Experiments

We used the FuzzyJ Toolkit for Java [33] to implement the necessary fuzzy logic functionality. Trace-based simulations were performed in order to avoid the danger, cost, and non-repeatability of creating fires. For our experiments we have used real fire data publicly available on the National Institute of Standards and Technology (NIST) website [34]. The study they performed provides sensor measurements from a number of different real fires. We have used two of the available scenarios: fire caused by a burning mattress and fire caused by a burning chair.

The membership functions for the input linguistic variables used in the experiments are shown in Figure 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. Figure 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, the probability that there is no fire is higher.

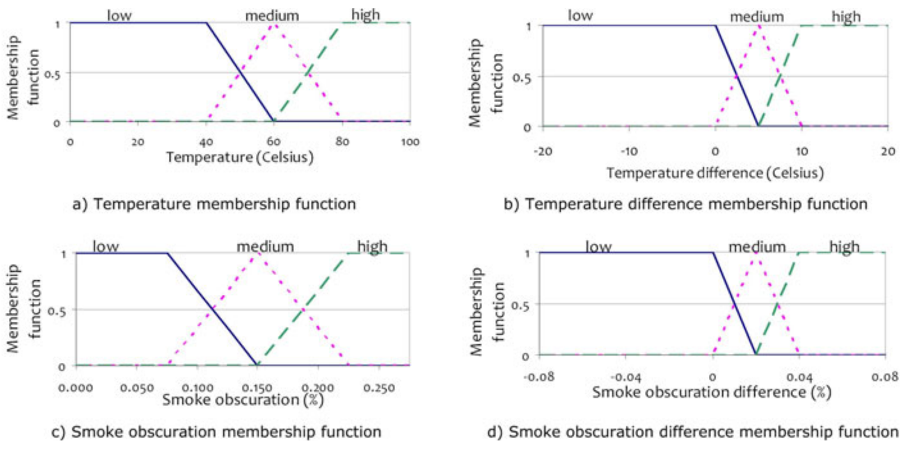

Fig. 3. Membership functions for the input linguistic variables

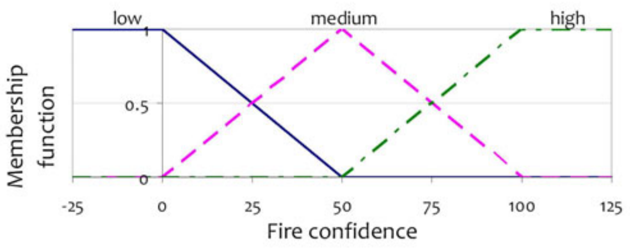

Fig. 4. Fire confidence membership function

To provide a baseline for our results, we performed crisp-value experiments with both the burning mattress and burning chair data. The temperature and smoke obscuration thresholds used in the crisp logic experiments are threshold values used in commercial smoke and heat detectors, 55℃ and 0.15 m^{-1}, respectively [35,36]. The membership functions in Figure 3 were also built according to these threshold values.

The results from the crisp-value experiment are shown in Figure 5 a) and b). In these and all following figures, the origin of the graph represents the time of fire ignition. As we can see from the two 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 affect 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. Since for a real fire detection system it is more important to decrease the number of false negatives than that

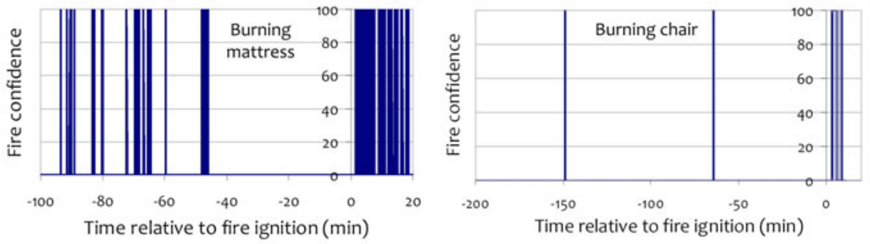

Fig. 5. Crisp value simulation: a) burning mattress b) burning chair

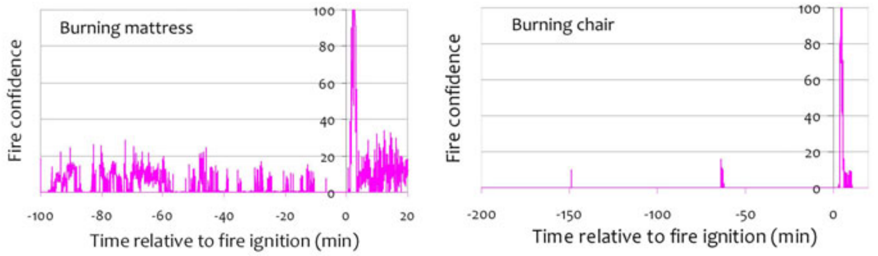

Fig. 6. Fuzzy value simulation: a) burning mattress b) burning chair

of false positives, 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 couple of fuzzy logic experiments a node decides if there is a fire based 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* (H), as shown by Figure 3 and Figure 4. We have used heuristics to build the rule-base for our fire detection experiments. In cases where this is not possible, e.g. when more complex events are to be detected, domain experts could be consulted for the definition of the rule-bases.

The results from our first couple of fuzzy logic experiments, a burning mattress and a burning chair, are presented in Figure 6 a) and b), 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. Both graphs show fire confidence around 0 before the ignition, except for a number of small peaks when the confidence increases to 25. 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

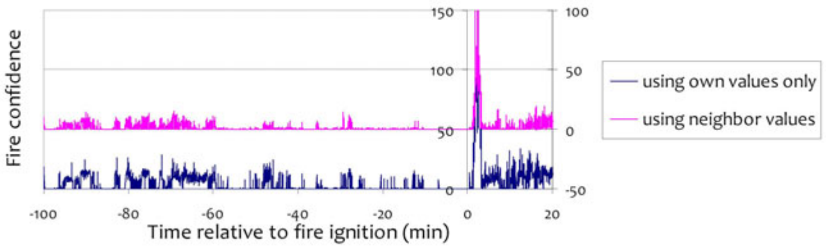

Fig. 7. Simulating a burning mattress: including neighbor readings in the decision. The results when only own values are used are plotted on the first y-axis. Including the neighbor values is plotted on the second y-axis.

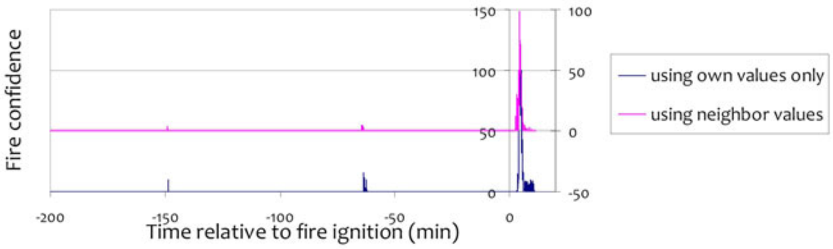

Fig. 8. Simulating a burning chair: including neighbor readings in the decision. The results when only own values are used are plotted on the first y-axis. Including the neighbor values is plotted on the second y-axis.

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 studied 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. The results in Figure 7 and Figure 8 show that fire is detected almost as quickly as when the decision process is based on only *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.

6.2 Analysis

Why does fuzzy logic perform better?. An interesting question is why fuzzy logic is more precise than crisp-value logic. From the considerable decrease in the number of false positives, it appears that fuzzy logic handles the fluctuating sensor readings much better. To see 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 minutes into the experiment. The values

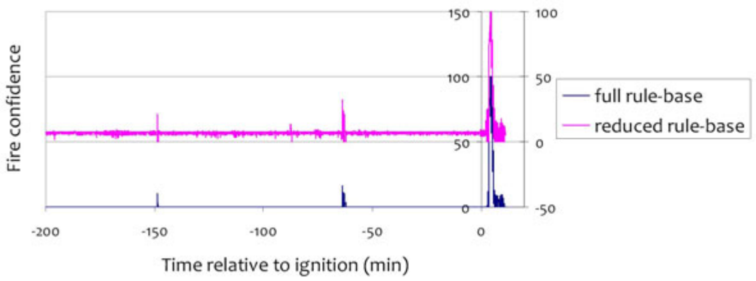

Fig. 9. Simulating a burning chair with a reduced rule-base. The results when the full rule-base is used are plotted on the first y-axis. Using the reduced rule-base is plotted on the second y-axis.

that caused the false alarm are; T = 25.21°C, ΔT = 0°C, S = 0.203 %, and ΔS= 0.109%. Since the smoke level and 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 Figure 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 0.109% is classified as 100% High. The decision making process checks which rules from the rule-base are satisfied, and the defuzzifier reports a fire confidence value of 22.5. This value maps to fire confidence which is 55% Low and 45% Medium. Since the Low confidence is higher, the logic determines that there is no need 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 only relying on exact values, which helps improve the accuracy of event detection.

Decreasing the rule-base. We applied our reduction techniques to the rule-base used in the simulation experiments. All nodes in the simulation are equipped with both a smoke and a temperature sensor which makes the first technique not applicable. Therefore, we only used the other two reduction techniques. The rule base initially had 81 rules. Combining the rules with similar outcomes reduced the number of rules to 30, which is a 63% decrease. 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 reduced the size of the rule-base to 24, which is 30% 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. Figure 9 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

Table 4. 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 5. 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

detections. For future work, we plan to perform deeper analysis of the memory requirements associated with using fuzzy logic.

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 [37] and a J48 decision tree which is an open source implementation of the C4.5 algorithm [38]. 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 nominal, it works with continuous values, which is exactly what the sensor readings are. In addition, specifying the membership functions is more intuitive and simpler than building a probability model.

We ran this set of experiments using the Weka data mining tool [39]. 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 4 shows the number of incorrectly classified instances for the two fire scenarios 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 what the fuzzy logic detection managed to achieve.

Fire detection delay. Table 5 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 slowed down.

7 Conclusions

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 to event detection in WSNs. Compared to using crisp values, fuzzy logic allows the detection algorithm to maintain 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 together reduced 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.

References

1. Cornell Database Group-Cougar, `http://www.cs.cornell.edu/bigreddata/cougar/`
2. Govindan, R., Hellerstein, J., Hong, W., Madden, S., Franklin, M., Shenker, S.: The sensor network as a database. Computer Science Department, University of Southern California, Technical Report 02-771 (2002)
3. Li, S., Son, S.H., Stankovic, J.: Event detection services using data service middleware in distributed sensor networks. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 502–517. Springer, Heidelberg (2003)
4. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: The design of an acquisitional query processor for sensor networks. In: SIGMOD, pp. 491–502 (2003)
5. Franklin, M.: Declarative interfaces to sensor networks. Presentation at NSF Sensor Workshop (2004)
6. Jiao, B., Son, S., Stankovic, J.: GEM: Generic event service middleware for wireless sensor networks. In: INSS (2005)
7. Kapitanova, K., Son, S.H.: MEDAL: A compact event description and analysis language for wireless sensor networks. In: INSS (2009)
8. Tapia, E., Intille, S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Pervasive Computing, pp. 158–175 (2004)
9. Wren, C., Tapia, E.: Toward scalable activity recognition for sensor networks. In: Location and Context-Awareness (LoCA), pp. 168–185 (2006)
10. Castro, P., Chiu, P., Kremenek, T., Muntz, R.R.: A probabilistic room location service for wireless networked environments. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 18–34. Springer, Heidelberg (2001)
11. Duarte, M., Hu, Y.-H.: Distance based decision fusion in a distributed wireless sensor network. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 392–404. Springer, Heidelberg (2003)
12. Chen, T.M., Venkataramanan, V.: Dempster-shafer theory for intrusion detection in ad hoc networks. IEEE Internet Computing, 35–41 (2005)

13. Wu, H., Siegel, M., Stiefelhagen, R., Yang, J.: Sensor fusion using dempster-shafer theory. In: Proceedings of IEEE IMTC, pp. 21–23 (2002)
14. Murphy, R.: Dempster-shafer theory for sensor fusion in autonomous mobilerobots. IEEE Transactions on Robotics and Automation, 197–206 (1998)
15. Wood, A., Virone, G., Doan, T., Cao, Q., Selavo, L., Wu, Y., Fang, L., He, Z., Lin, S., Stankovic, J.: Alarm-net: Wireless sensor networks for assisted-living and residential monitoring. University of Virginia, Technical Report CS-2006-13 (2006)
16. Lymberopoulos, D., Ogale, A., Savvides, A., Aloimonos, Y.: A sensory grammar for inferring behaviors in sensor networks. In: IPSN, pp. 251–259 (2006)
17. Ghasemzadeh, H., Barnes, J., Guenterberg, E., Jafari, R.: A phonological expression for physical movement monitoring in body sensor networks. In: MASS, pp. 58–68 (2008)
18. Amft, O., Kusserow, M., Tröster, G.: Probabilistic parsing of dietary activity events. In: BSN, pp. 242–247 (2007)
19. Gupta, I., Riordan, D., Sampalli, S.: Cluster-head election using fuzzy logic for wireless sensor networks. In: CNSR, pp. 255–260 (2005)
20. Kim, J., Park, S., Han, Y., Chung, T.: CHEF: Cluster head election mechanism using fuzzy logic in wireless sensor networks. In: ICACT, pp. 654–659 (2008)
21. Lee, H., Cho, T.: Fuzzy logic based key disseminating in ubiquitous sensor networks. In: ICACT, pp. 958–962 (2008)
22. Kim, B., Lee, H., Cho, T.: Fuzzy key dissemination limiting method for the dynamic filtering-based sensor networks. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS, vol. 4681, pp. 263–272. Springer, Heidelberg (2007)
23. Lazzerini, B., Marcelloni, F., Vecchio, M., Croce, S., Monaldi, E.: A fuzzy approach to data aggregation to reduce power consumption in wireless sensor networks. In: NAFIPS, pp. 436–441 (2006)
24. Kim, J., Cho, T.: Routing path generation for reliable transmission in sensor networks using GA with fuzzy logic based fitness function. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part III. LNCS, vol. 4707, pp. 637–648. Springer, Heidelberg (2007)
25. Chiang, S.-Y., Wang, J.-L.: Routing analysis using fuzzy logic systems in wireless sensor networks. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 966–973. Springer, Heidelberg (2008)
26. Ren, Q., Liang, Q.: Fuzzy logic-optimized secure media access control (fsmac) protocol wireless sensor networks. In: CIHSPS, pp. 37–43 (2005)
27. Munir, S.A., Bin, Y.W., Biao, R., Jian, M.: Fuzzy logic based congestion estimation for qos in wireless sensor network. In: WCNC, pp. 4336–4341 (2007)
28. Xia, F., Zhao, W., Sun, Y., Tian, Y.-C.: Fuzzy Logic Control Based QoS Management in Wireless Sensor/Actuator Networks. Sensors, 3179–3191 (2007)
29. Liang, Q., Wang, L.: Event detection in wireless sensor networks using fuzzy logic system. In: CIHSPS (2005)
30. Marin-Perianu, M., Havinga, P.: D-FLER: A distributed fuzzy logic engine for rule-based wireless sensor networks. In: Ichikawa, H., Cho, W.-D., Satoh, I., Youn, H.Y. (eds.) UCS 2007. LNCS, vol. 4836, pp. 86–101. Springer, Heidelberg (2007)
31. Zadeh, L.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics, 28–44 (1973)
32. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic: theory and applications. Prentice-Hall, Inc., Upper Saddle River (1995)
33. NRC FuzzyJ Toolkit, `http://www.csie.ntu.edu.tw/sylee/courses/fuzzyj/docs/`

34. Building and fire research laboratory, `http://smokealarm.nist.gov/`
35. WS4916 Series Wireless Smoke Detector
36. Geiman, J., Gottuk, D.: Alarm thresholds for smoke detector modeling, pp. 197–208 (2003)
37. Lewis, D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 4–15. Springer, Heidelberg (1998)
38. Quinlan, J.R.: C4.5: Programs for Machine Learning (1993)
39. Hall, M., Frank, E., Holmes, G., Pfahringera, B., Reutemann, P., Witten, I.: The WEKA data mining software: An update (2009)