International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

A New Fuzzy Query Processing System in Wireless Sensor Networks

PARAND AKHLAGHI *

Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Email: p.akhlaghi@srbiau.ac.ir

Abstract

The task of acquiring information from sensor networks through generating queries is one of the most important issues in wireless sensor networks. The structure of traditional query processing systems requires defining query criteria in the form of crisp predicates with explicit and numerical thresholds, leading them to be processed in a certain manner. The inherent uncertainty and imprecision of sensor data call for a new approach towards them. Since fuzzy theory provides a toolbox to capture the imprecision associated with both data and query, in this paper, a new system for processing fuzzy queries in wireless sensor networks is introduced. In this system, in addition to presenting a new structure for fuzzy queries, a new algorithm is introduced for processing fuzzy queries in sensor networks. Simulation results indicate that accuracy and precision of the results obtained from fuzzy queries are higher than traditional ones, whereas there is no significant difference between the two regarding their energy consumption.

Keywords wireless sensor networks; query processing; in-network processing; flexible processing; fuzzy query; fuzzy proposition; is-predicate; correlation index

1. Introduction

Wireless sensor networks consist of a set of nodes equipped with different kind of sensors which communicate each other through wireless connections for such environmental monitoring applications. In these networks, the task of delivering sensor readings is done in two ways; proactive and reactive approach.

----

* Corresponding author.

In proactive approach, all sensor readings are sent to a base station periodically or event-based, and the raw data will be processed later at the base station. Whereas, in reactive (on-demand) method, the sensor readings are sent to the base station when they are requested by the user. For this end, user defines the desired information to acquire from network, through issuing a query. In respond to the query, only those sensors propagate their readings which are requested. It means that, the processing of data is performed inside the network in on-demand approach. Different kinds of systems such as TinyDB [1] and Cougar [2] has been introduced which are acquiring information from the sensor network based on query processing paradigm. In these query processing systems, user specifies what he needs from the network in a form of SQL-like query and propagates the query to the network. Sensors who receive this query, should first process it. If the sensors' readings satisfy the query criteria, it means that they could respond to the query, thus they send the requested readings to the base station. Whether, if the sensors' readings could not satisfy the query criteria, nothing is sent to the base station. Therefore, the reactive methods are more energy-efficient than the proactive ones. For instance, if the user wants to identify the sensors with irregular pressure value, he could issue a query like: "Q1: SELECT *nodeid* FROM *sensors* WHERE *pressure* >= 1025". In this example, the criterion of the query is a certain predicate "*pressure* >= 1025", in which the irregular pressure has been expressed by a numerical threshold 1025. Each sensor receives this query, process it and evaluates the predicate. If the pressure value of the sensor is exactly equal to 1025 or higher than 1025, it can send the requested data to the base station, which is the nodeid of sensor in this query.

The above example is an instance of classic queries in sensor networks. In traditional systems, the query criteria are expressed in the form of certain predicates with numerical thresholds. The processing of query is performed through evaluating the query predicates. If the truth-value of query predicate is true for a sensor, it means that sensor's readings satisfy the query criteria and the sensor could respond to the query. Since the truth-value of certain predicate is absolutely true or false, the sensor definitely decides to answer the query, regardless of any additional information. Moreover, the truth-value of crisp predicate sharply separates the result set from a non-result set. For instance, in the predicate "*pressure* >= 1025", if the sensor's pressure is equal to 1024.9, which is very close to desired threshold, since the truth-value of predicate "*pressure* >= 1025" for pressure value 1024.9 is not true, then the sensor node could not respond to this query. Also, the fact of exact processing of crisp predicate will be critical in the presence of sensing error measurement, which is common in sensor networks.

On the other hand, there are so many events in the real world that are difficult to be expressed by numerical values. For example, the event such as *hotness* does not have a certain definition. As in some domains the *temperature* value of 35 degrees of Celsius is interpreted as hot while in another domain, this value does not indicate the hotness. Since the linguistic labels are interpreted based on the situation, applying them to model the real world is more convenient. Therefore, a new tool is required to express the environmental events through linguistic labels, instead of numerical thresholds. Since fuzzy theory [3] provides a tool for applying linguistic labels and processing inexact data, we can handle imprecise data through issuing a fuzzy query. In fuzzy approach, each linguistic label is specified by a number of fuzzy membership functions, which can be vary in different situations.

Hence, in this paper, we propose a new fuzzy query processing system for sensor networks. We present a basic structure for fuzzy queries and a distributed algorithm for processing fuzzy queries in sensor networks. We propose to use fuzzy proposition in the query predicate rather than crisp proposition. By this way, the desired criteria could be expressed through linguistic labels. Furthermore, since the truth-value of fuzzy propositions is not definitely true or false and they have a degree of truth, the sensor node cannot definitely decide to answer the query. To this end, we propose a new component in the query structure, which is a correlation benchmark to specify which sensors should respond to the query. This benchmark is evaluated as the correlation between a node and a query. Given that, our query is a fuzzy query, we also express this benchmark as fuzzy. We propose a correlation index to specify the user-desired correlation in the query, through linguistic labels such as *weak*, *medium* and *strong*. Whenever the query is processed by sensor nodes, this index is used to evaluate the fuzzy proposition and to make a decision to answer the query.

In the following, we first review some literatures related to our work, in section 2. Then in section 3, our fuzzy query processing system will be presented in details. We describe the proposed structure and the processing mechanism in this section. The experimental analysis is described in section 4. Section 5 present concluding remarks.

2. Related Works

Inasmuch as our proposed approach is a new system for generating and processing fuzzy queries in wireless sensor networks, some literatures related to query processing, correlation-based query and fuzzy query processing in wireless sensor networks are reviewed in this section.

Recent researches use a database-like abstraction for acquiring information in sensor networks [1,2] and [4]. In effect, we can view the sensor networks as a distributed database into which data can be stored and from which desired data can be extracted by injecting queries. In-network processing has proved as one of the most energy efficient techniques to be used along with the database abstraction to retain query processing inside the sensor networks and close to the data sources. A number of query processing systems have been proposed to use in sensor networks such as TinyDB [1] and COUGAR [2]. TinyDB incorporates acquisitional techniques which in smart sensors have control over where, when, and how often data is physically acquired and delivered to query processing operators [1]. Yao et al. [2] evaluate the design of a query layer that accepts queries in a declarative language which are then optimized to generate efficient query execution plans with in-network processing which can significantly reduce resource requirements. Typically, these systems use the Structured Query Language (SQL) with the form “SELECT, FROM, WHERE” for specifying queries. Another aspect of acquisitional query processing is to maximize in-network query processing to reduce sensor resource usage while still meeting the query precision specifications. However, these query models usually require a precise, unambiguous specification of a query and the predicative thresholds in its WHERE-clause. Furthermore, in these systems all sensor nodes have an identical interpretation of data, which limits their application to homogenous settings. Moreover, since these models use two-valued logic in the querying process, small error in data values may consequence inadequacy of the selected data. Hence, they are not tolerable against imprecise data. Furthermore, it is not possible to use semantic concepts in their query structure, while it is more convenient for the user to express the preferences in conceptual terms.

Applying the correlation concept in processing queries has been subject of many research papers. The authors of [5] showed how to exploit correlations between attributes in a database system by modifying the query optimizer to produce conditional plans that significantly outperform plans produced by traditional database optimizers. Their approach is particularly useful in systems where the cost of acquiring some of the attributes is non-negligible and where correlations exist between one or more attributes. [6,7] use a correlation-aware probabilistic model for processing queries. Rather than directly querying the sensor network, they build a model from stored and current readings, and answer SQL queries by consulting the model. Such a correlation model in [5,6] builds one joint distribution table over all the sensors attributes and infers the probabilistic value of an attribute by conditioning all the other attributes on it. However, model-driven schemes are only applicable in scale systems with relatively low statistical dynamics. In contrast, [8] uses Bayesian networks to extract correlations between the attributes. Hence, for the estimation of a given sensor value instead of defining the model based on all other sensors, they use just those sensors that have a direct effect on that sensor. Our work is most related to the research work proposed in [9]. In [9], the correlation between a query and a node is calculated by the vector space model (VSM), and a query correlation indicator (QCI) is designed to quantify the priority of answering a query for individual nodes. However, a shortcoming associated with [9] as well as the aforementioned correlation-based schemes is that they do not explicitly factor in the imprecision of data or query in their design. Our technique on the other hand, utilize a fuzzy model to capture the correlation between a query and a node. Moreover, since our query is issued in the form of fuzzy query, our design can particularly handle imprecisions of both data and query.

Introducing flexibility into queries in sensor networks is a relatively new research topic. In [10] authors have proposed an active database approach employing a fuzzy Petri net model, which processes uncertain sensor data and handles flexible and continuous queries. However, this approach proposes no in-network processing of sensor data. It requires that all data be transmitted to a central gateway, thus treating the sensor network as an input data stream device, which may not be practical in large networks. Doman et al. [11] builds on the SwissQM platform [12] and extends it to implement a fuzzy query processing model. The proposed approach applies fuzzy membership functions to evaluate threshold-based predicates in a given query. However, [11, 13] did not explicitly support the fuzzy declaration of the query predicates in the more user-friendly form of “is-predicates”. Our work is also related to this research, however, we define a novel structure for fuzzy query in a more general form. In our proposed structure, query predicate can also be expressed as fuzzy predicate. This relieves the query issuer from dealing with numerical values in the query’s WHERE-clause. In addition, we introduce a new fuzzy-based query correlation index into the query structure, which can be specified by the user in fuzzy form to further restrict the results set according to user’s soft preferences. Also, we propose a comprehensive distributed algorithm for processing fuzzy queries in sensor networks.

3. Basic Fuzzy Query Processing System

In this paper, a fully fuzzy based system has been introduced in which queries are expressed and processed in a fuzzy manner. We first propose our new structure for expressing fuzzy queries and then present a comprehensive distributed algorithm for processing these queries. The incentive behind proposing a new structure is to provide the ability of expressing sensor queries as fuzzy. In the fuzzy queries, linguistic labels and fuzzy propositions can be applied instead of crisp propositions with certain numerical thresholds. While the truth value of fuzzy propositions is in the range of [0, 1], that of crisp propositions is certainly 0 or 1 [14]. Since the truth value of fuzzy propositions is not definitely true (1) or false (0), it is satisfied by more sensors and will therefore lead to collecting more results compared to crisp propositions. Consequently, using a mechanism to prune collected results is required. For this end, we propose to apply a node query correlation, which shows the correlation between a node and a query, as a benchmark for selecting more correlated results [15]. Since this benchmark should also be quantified as fuzzy, we have defined a fuzzy correlation index which indicates a desired correlation that the user considers for collecting the most related results. In the query structure, we have added a new clause for the user to express his desired correlation through a fuzzy term. This clause is used to filter the results in the processing phase. Thus, only those sensors could respond to the query which not only their readings satisfy the query criteria, but also their correlation to the query meets the desired index. In what follows, we first explain the suggested structure for fuzzy query in section 3.1. After that, in section 3.2 the proposed processing algorithm will be described.

3.1. The Proposed Structure For Fuzzy Query

In this section, we present our proposed basic structure for fuzzy queries. This structure is the enhanced form of the structure that has been used in TinyDB [1], which is also a clause-based. The proposed structure is shown in Figure 1. It consists of some clauses such as SELECT, WHERE, EPOCH and CORRELATION. As regards before, we added the CORRELATION-clause in the query structure for pruning results by correlation index. We explain each of these clauses by details in the following.

SELECT *attr1, attr2, …*
FROM *Sensors*
WHERE P_Q
EPOCH *t*
CORRELATION *desired-corr*

Figure 1: The proposed structure for fuzzy query.

In the SELECT-clause, the user specifies the list of sensor attributes that he would like to extract them from the sensor network. This clause starts with "SELECT" followed by a list of sensor attributes. The sensor attributes could be fuzzy or non-fuzzy attributes, such as *light* or *humidity* for fuzzy attributes and, *node id* or *position* for non-fuzzy attributes.

In the WHERE-clause, user stipulates the query predicate to determine the criteria for extracting data. This clause starts with "WHERE" followed by the query predicate P_Q. Since queries in the proposed system, have only a single predicate, we use simple fuzzy proposition for query predicate. The formal form of simple fuzzy proposition, according to "unqualified fuzzy proposition" [16] (Definition 5 in Appendix), is defined in below, in which V belongs to fuzzy attribute set. The notation ○ is used to indicate "is" or 'is not" phrase, and F is one of the fuzzy terms which has been defined for fuzzy attribute V. Based on fuzzy term's definition, the F could be a simple fuzzy term, or a composite fuzzy term that combined with some fuzzy modifiers (see Definition 3 in

Appendix) [17].

$$P_Q \overset{def}{=} V \circ F \quad (1)$$

In the EPOCH-clause, the sampling rate or duration of the query is expressed.

In the CORRELATION-clause, user expresses a fuzzy term to specify the desired *correlation index*. As regards before, we introduced a new index to determine by which intensity, the selected results should satisfy the query predicate stipulated in the WHERE-clause. So the *desired-corr* could be a fuzzy term such as, *strong, medium* or *weak*. For instance, when a user wants to select the sensors that their readings satisfy the query criteria with high correlation, he sets the correlation index as *strong*. This clause is started with "CORRELATION" followed by *desired-corr*. In the Table 1, definition of fuzzy terms for *correlation index* and their related fuzzy sets are shown.

Table 1: Fuzzy terms and related fuzzy sets for *correlation index.*

correlation index	Related Membership Functions
Strong	Gamma-Function(*a*,*b*)
Medium	Trapezoid-Function(*a*,*b*,*c*,*d*)
Weak	L-Function(*a*,*b*)

For instance, in *Q1* user wants to extract the nodeid and light values of sensors whose temperature values are hot every 15 *Sec* and he desires to select sensors with *strong* correlation. In the next section, we describe how sensor nodes process this query.

Q1: SELECT *nodeid, light*
FROM *sensors*
WHERE *temperature is hot*
EPOCH 15 *sec*
CORRELATION *strong*

3.2. Fuzzy Query Processing Algorithm

Here, we describe the overall processing that has been done in each sensor node. When a sensor node receives a query, it should check the query criteria. If the sensor readings satisfy the criteria, this node would respond to the query by sending the requested attributes. This task is performed in four steps, which are explained by details in the following.

- Query Predicate Evaluation phase:

In the first step, a query predicate, which is stipulated in the WHERE-clause, is evaluated through calculating its truth value. For this end, the sensor node should sample a sensor attribute, which is stipulated in the query predicate. If the query predicate is P_Q: $V \circ F$, then, the sensor node should sample the attribute value V which is v. If the ○ notation is "is" phrase, then the equation (2) is used to calculate the T_{PQ}, otherwise (it is "is not"), the equation (3) is utilized.

$$T_{P_Q} = \mu_F(V = v) \tag{2}$$

$$T_{P_Q} = 1 - \mu_F(V = v) \tag{3}$$

In the above equations, according to the definition of fuzzy terms (see Definition 3 in Appendix), the defined membership function for fuzzy term will be used, if that term is simple. Whereas, the modified form of the membership function will be used, for the composite fuzzy term. The output of this step, is the truth value of query predicate, T_{PQ}, which we call *node-degree*.

Since this *node-degree* does not definitely clarify whether a sensor reading satisfies a query criteria or not, we introduce a *correlation index*, which is a fuzzy correlation benchmark for demonstrating the correlation between a query and a sensor node.

- Correlation Calculation phase:

In the second step, this correlation is computed and the *node-corr* is given as result. The *node-corr* demonstrates, by which intensity the sensor readings satisfy the query criteria. The *node-corr* is a fuzzy term that has the maximum membership degree in the associated fuzzy sets. In order to find out this term, a membership degree of the *node-degree* in all three fuzzy sets that are associated with "*weak*", "*medium*" and "*strong*" terms, should be calculated and the maximum degree is chosen as *correlation-degree* (equation (4)). In this way, the fuzzy term related to the *correlation-degree* is considered as *node-corr*. The *query-corr* is calculated through equation (5).

$$correlation - degree = \underset{i \in \{weak, medium, strong\}}{Max} (\mu_i(node - degree)) \tag{4}$$

$$node - corr = \{i \mid \mu_i(node - degree) = correlation - degree\} \tag{5}$$

It should be mentioned that the fuzzy sets associated with "*weak*", "*medium*" and "*strong*" terms are orthogonal. It means that, the sum of membership degrees of all these three sets in a specific point, should be equal to 1 as can be seen in equation (6).

$$\mu_{weak}(x_i) + \mu_{medium}(x_i) + \mu_{strong}(x_i) = 1 \quad \text{(for all } x_i \in X\text{)} \tag{6}$$

- Decision Making Based on Desired Correlation phase

In the third step, the sensor node decides whether or not to respond to the query. The decision is undertaken through comparing a *desired-corr* (which is stipulated in the query) with the *node-corr* (which is calculated in the second step). If the *node-corr* is equal to the *desired-corr*, the processing goes to last step, otherwise the processing will be finished.

- Responding to Query phase

At the fourth step, the sensor node should sample requested attributes, which are listed in the SELECT-clause, and transmits them with the calculated *node-degree* value to the base station. If the outcome of third phase lead to responding to the query, sensor node will sample the attributes which are requested in the SELECT-clause, and will transmit them with a calculated *node-degree* to the base station. This processing will repeat in time interval which is specified by Epoch-clause.

4. Analytical Results

In this section, the performance of our fuzzy-based query processing approach is compared with classical non fuzzy-based mechanism. Classical schemes may entail almost all ordinary SQL-like query processors (e.g. TinyDB [1], Cougar [2], etc.) which rely on rigid predicate-based evaluation of sensor readings to determine the response set for a given query.

We have deployed our proposed query processing scheme in Castalia 3.2 simulation environment [18] which is based on OMNET++ [19] simulator. We have modified the sensor manager module to extend the single sensing device type to multiple sensing device types (such as *temperature*, *light*, *pressure*, etc.). The MAC layer's operation is managed by TMAC [20], and the routing process is handled by Multipath Ring Routing Protocol [21]. Both of our proposed fuzzy processor and classical querying scheme are implemented at the application layer on top of the aforementioned protocol stack. In the study, the network is comprised of a single sink node together with a number of resource-constrained immobile homogeneous sensor nodes with symmetric radio links in between. The query dissemination process by the sink is a simple reception-broadcast protocol, i.e. the sink relies on a basic multi-hop broadcasting scheme to diffuse each received query throughout the network.

4.1. Experiment Description

The purpose of this study is to demonstrate the robustness of our fuzzy querying scheme against varying degrees of device measurement error and environmental noise. The experimental setting is a homogenous sensing environment and we are interested in identifying the sensor nodes with temperature values in a specific range. We consider a wireless sensor network comprised of 100 nodes deployed in a $10m \times 10m$ grid-like environment. 30% of the nodes are located in a region with temperature values in the range (12°c, 22°c), 10% of nodes are in the 12°c region, and 10% of nodes are in the region with $22^{\circ}c$ temperature. We assume that the query issuer is interested in identifying the sensors located within the so-called *cool* region of the network with temperature values in the closed interval of [12°c, 22°c]. The classical query and its fuzzy counterpart are shown in below. Given the homogeneity of the environment, the fuzzy membership function is defined for *cool* temperature to be a trapezoid function (equation (7)) with identical parameters uniformly across all nodes. The *correlation index* of *strong*, *medium* and *weak* are characterized through equations (8) to (10) according to Table 1. The simulation parameters for this scenario are shown in Table 2.

Classical Query:

Q: SELECT *nodeid, temperature*

FROM *sensors*

WHERE *temperature* <= 22 *AND*

temperature >=12

EPOCH 200 *sec*

Fuzzy Query:

FQ: SELECT *nodeid, temperature*

FROM *sensors*

WHERE *temperature is cool*

EPOCH 200 *Sec*

CORRELATION *strong*

$$\mu_{Cool}(temp) = \begin{cases} 0 & temp \le 10.44, temp \ge 23 \\ \dfrac{temp - 10.44}{12 - 10.44} & a \le temp \le 12 \\ 1 & 12 \le temp \le 22 \\ \dfrac{23 - temp}{23 - 22} & 22 \le temp \le 23 \end{cases} \tag{7}$$

$$\mu_{medium}(x) = \begin{cases} 0 & x \le 0.2, x \ge 0.85 \\ \dfrac{x - 0.2}{0.2} & 0.2 < x \le 0.4 \\ 1 & 0.4 < x < 0.6 \\ \dfrac{0.85 - x}{0.25} & 0.6 \le x < 0.85 \end{cases} \tag{8}$$

$$\mu_{weak}(x) = \begin{cases} 1 & x \ge 0.2 \\ \dfrac{0.4 - x}{0.4 - 0.2} & 0.2 \le x \le 0.4 \\ 0 & x \ge 0.45 \end{cases} \tag{9}$$

$$\mu_{strong}(x) = \begin{cases} 0 & x \le 0.6 \\ \dfrac{x - 0.6}{0.85 - 0.6} & 0.6 \le x \le 0.85 \\ 1 & x \ge 0.85 \end{cases} \tag{10}$$

Table 2 Simulation parameter for measuring temperature in homogenous noisy environment.

Simulation Parameter	Value
Simulation Time (Sec)	*600*
Simulation Area (meter)	*10 × 10*
Node Deployment	*Grid 10 × 10*
Query Epoch (Sec)	*200*
No. of Sensor Nodes	*100*
No. of Nodes with Temperature in range (12,22)	*30*
No. of Nodes with Temperature is equal to 12	*10*
No. of Nodes with Temperature is equal to 22	*10*
Number of sensors we expected to answer	*50*
Standard Deviation of Device Noise	*0,0.25,0.5,0.75,1,1.25, 1.5,1.75,2*
Standard Deviation of Device Bias	*0,0.25,0.5,0.75,1,1.25, 1.5,1.75,2*

We assume that a sensor node's reading is subject to Gaussian error and environmental noise. Hence, we use equation (11) to express a given node's read-out value *x* [19] where *value* is the real value, *Device_Bias* is the measurement error introduced by the limitations of the device's hardware with a normal probability distribution $N(0,\sigma^2_b)$ (see equation (12)), and *Device_Noise* is the environmental noise with normal distribution $N(0, \sigma^2_n)$ (equation (13)).

$$x = Value + Device\_Bias + Device\_Noise \tag{11}$$

$$Device\_Bias \sim \mathrm{N}(0, \sigma_b^2) \tag{12}$$

$$Device\_Noise \sim \mathrm{N}(0, \sigma_n^2) \tag{13}$$

4.2. Discussion of Results

Figure 2- 5 depict the precision, recall and the accuracy of both the classical query scheme and our fuzzy-based method for varying degrees of measurement error and environmental noise. Overall, we observe that the fuzzy method turns out to be more accurate; more specifically, it outperforms classical query processing in terms of recall with only a slight detriment in precision. The classical scheme's higher precision can be attributed to its threshold hypersensitivity, which strictly excludes out-of-range sensors from the result set. The fuzzy method, on the other hand, exhibits more flexibility towards the specified thresholds, and accordingly achieves a higher recall by incorporating more relevant sensors into the result set. As the error values grow larger, both schemes show similar trends, in that they both lose their precision and accuracy with almost equal rates. The fuzzy scheme's superiority in terms of accuracy becomes more apparent at σ^2=0.25 for measurement error. We single out the case σ^2=0.25 in Table 3 as it deserves further discussion. A point of controversy is the slightly higher (energy) efficiency of the classical scheme (Figure 5); however, it should be noted that the energy consumed by our fuzzy method is roughly equal to what is actually needed to retrieve the ideal result set; in other terms, the classical scheme has no superiority over our proposed method once we aim for identifying the complete set of relevant sensors.

Table 3 Acquired results for standard deviation 0.25 for 50 nodes.

Query	Precision	Recall	Accuracy	Total Energy Consumption (Joule)	Accuracy/ Energy
Classic Query	0.99	0.79	0.89	0.00230	386.95
Fuzzy Query	0.98	0.96	0.97	0.00283	342.75

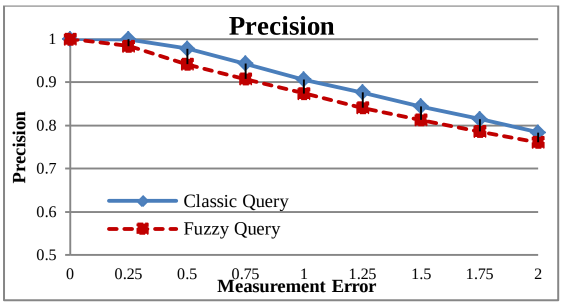

Figure 2: Comparison between the Precision of fuzzy and classic results

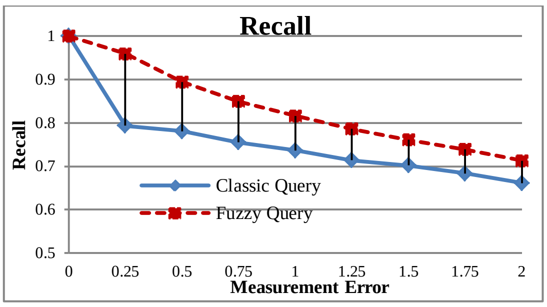

Figure 3: Comparison between the Recall of fuzzy and classic results

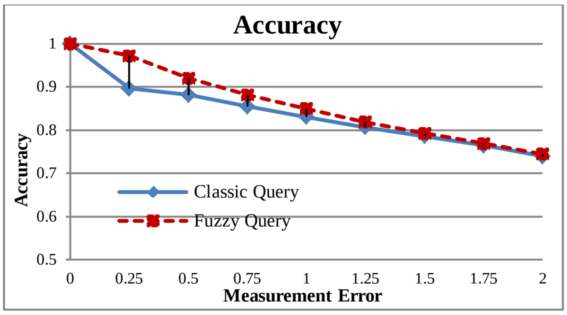

Figure 4: Comparison between the Accuracy of fuzzy and classic results

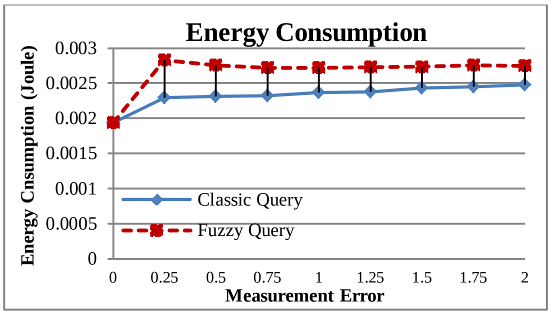

Figure 5: Comparison between the total Energy consumed in the fuzzy and classic query

5. Conclusion and Recommendations

In this paper, a new query processing system has been introduced for generating and processing fuzzy queries in wireless sensor networks. As regards, there has not been any comprehensive query processing system for fuzzy queries in wireless sensor networks previously, a new basic structure for generating fuzzy queries has been presented, in addition a new algorithm for processing these queries in the proposed system has been offered. In the presented structure, users can issue queries with linguistic labels and fuzzy propositions. Since, the truth value of fuzzy proposition is in the range of [0, 1], more sensor values are placed in the result set in compare with crisp proposition. To overcome the implosion of results, we recommended to use correlation as a benchmark for selecting more related results. For this end, the *correlation index* is introduced which is also expressed in a fuzzy manner. This index is provided the possibility of specifying a desired satisfaction degree for users, through some linguistic labels such as *strong*, *medium* or *weak*, to select more relevant results. The proposed system is a basic platform for fuzzy queries which can be improved in the future. The evaluation has shown that the completeness and the correctness of the fuzzy query, are higher in comparison with the classical query, which lead to achieving more accurate results while spending negligible energy.

In this paper, we apply simple fuzzy queries with a single predicate in the proposed fuzzy system, which is the constraint of our work. Therefore, we intend to improve it to handle more complicated fuzzy queries. In the future, we will enhance the query structure to support queries with multiple predicates. When the query has multiple predicates, the compound fuzzy proposition with aggregation operators “AND” and “OR” should be used. In addition, to process queries with multiple predicates, we will also propose an enhanced version of the fuzzy query processing algorithm.

Reference

[1] S. R. Madden, M. J. Franklin, J. M. Hellerstein and W.Hong (2005) ‘TinyDB: An acquisitional query processing system for sensor networks’, *ACM Transactions on Database Systems (TODS)*, ACM. Vol. 30, No. 1, pp. 122-173.

[2] Y. Yao and J. Gehrke (2002) ‘The cougar approach to in-network query processing in sensor networks’, *ACM Sigmod Record*, Vol. 31, No. 3, pp.9-18.

[3] L. A. Zadeh (1965) 'Fuzzy sets', *Information and control*, Elsevier, Vol. 8, No.3, pp. 338-353.

[4] S. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong (2003) 'The design of an acquisitional query processor for sensor networks', In SIGMOD 2003: *Proceedings of the 2003 ACM SIGMOD international conference on Management of data.* San Diego, California, USA, ACM. pp. 491-502.

[5] A. Deshpande, C. Guestrin, W. Hong and S. Madden (2005) 'Exploiting correlated attributes in acquisitional query processing', In ICDE 2005: *Proceedings of the 21st International Conference on Data Engineering, 2005,* Tokyo, Japan, IEEE, pp. 143-154.

[6] A. Deshpande, C.Guestrin, S. R. Madden, J. M. Hellerstein and W. Hong (2004) 'Model-driven data acquisition in sensor networks', In VLDB '04: *Proceedings of the Thirtieth international conference on Very large data bases*-Volume 30. Toronto, Canada, VLDB Endowment, pp.588-599.

[7] D. Chu, A. Deshpande, J. M. Hellerstein and W. Hong (2006) 'Approximate data collection in sensor networks using probabilistic models', In ICDE'06: *Proceedings of the 22nd International Conference on Data Engineering, 2006,* Atlanta, GA, USA, IEEE, pp.48-59.

[8] A. Jain, E. Y. Chang, and Y.-F. Wang (2007) 'Bayesian reasoning for sensor group-queries and diagnosis', *Advances in Databases: Concepts, Systems and Applications*. Lecture Notes in Computer Science. Vol. 4443, Springer. pp. 522-538.

[9] Q. Ren, and Q. Liang (2007) 'Energy and quality aware query processing in wireless sensor database systems', *Information Sciences*, Elsevier ,Vol. 177, No.10, pp.2188-2205.

[10] B. Bostan-Korpeoglu, A. Yazici , I. Korpeoglu and R. George (2006) 'A New Approach for Information Processing in Wireless Sensor Network', In ICDEW'06: *Proceedings of 22nd International Conference on Data Engineering Workshops, 2006,* Atlanta, GA, USA, IEEE, pp.34-40.

[11] M. Doman, J. Payton and T. Dahlberg, (2010) 'Leveraging fuzzy query processing to support applications in wireless sensor networks', In SAC'10: *Proceedings of the 25th ACM Symposium on Applied Computing, 2010.* Sierre, Switzerland, ACM, pp. 764-771.

[12] R. Mueller, G. Alonso and D. Kossmann (2007) 'SwissQM: Next Generation Data Processing in Sensor Networks', In CIDR 2007: *Third Biennial Conference on Innovative Data Systems Research, 2007,* Asilomar, CA, USA. pp. 1-9.

[13] M. Doman (2009). 'The design and implementation of fuzzy query processing on sensor networks', (Doctoral dissertation, University of North Carolina at Charlotte).

[14] J. G. Shanahan (2012). 'Soft computing for knowledge discovery: introducing Cartesian granule features' (Vol. 570). Springer Science & Business Media.

[15] Q. Ren and Q. Liang (2007, March). ‘A VSM-Based and Quality-Aware Query Processing Protocol for Wireless Sensor Networks’. *In Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE* (pp. 2648-2652). IEEE.

[16] G. Klir and B. Yuan (1995). ‘Fuzzy sets and fuzzy logic: Theory and Applications’. *(Vol. 4). New Jersey:* Prentice Hall.

[17] M. De Cock, (1999). ‘Representing the adverb very in fuzzy set theory’. *In Proceedings of the ESSLLI99 Student Session 11th European Summer School in Logic, Language and Information.*

[18] A. Boulis, Castalia. [Online]. Available: http://castalia.npc.nicta.com.au/index.php.

[19] OMNET++ discrete event simulator. [Online]. Available: http://www.omnetpp.org.

[20] T. Van Dam and K. Langendoen, (2003). ‘An adaptive energy-efficient MAC protocol for wireless sensor networks’, *In ACM SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems 2003, Los Angeles, CA, USA. ACM. pp. 171-180.*

[21] A. Pandya and M. Mehta, (2012). ‘Performance Evaluation of Multipath Ring Routing Protocol for Wireless Sensor Network’, *UACEE International Journal of Advances in Computer Networks and its Security, Vol. 2, No. 2, pp. 53-58*.

[22] J. Galindo (2006). ‘*Fuzzy databases: Modeling, design and implementation’*, IGI Global.

[23] Z. M. Ma and L. Yan, (2007). ‘Generalization of strategies for fuzzy query translation in classical relational databases’. *Information and Software Technology, 49(2), 172-180.*

[24] L. A. Zadeh (1975). ‘The concept of a linguistic variable and its application to approximate reasoning—I’. *Information sciences, 8(3), 199-249.*

Appendix

Regarding to our fuzzy based method for expressing and processing queries in sensor networks, we first elaborate some fuzzy definitions and terminologies in this section. Below we define most important fuzzy terminologies that are used in our discussions, such as fuzzy set, fuzzy term, linguistic variable and fuzzy proposition.

Definition 1- Linguistic label: linguistic label is a word in natural language that has semantic concept such as hot, strong, dark, high, etc. The intuitive definition of these labels not only varies from person to person and from time to time, but also varies within the context they are applied [20].

Definition 2- Fuzzy set: a fuzzy set A in a universal set X is associated with a membership function $\mu_A(x)$ which assigns to each point of x from domain X, a real number in the interval [0,1]. The value of $\mu_A(x)$

represents the grade of membership of *x* in *A*.

$$A = \{(x, \mu_A(x)) \mid x \in X, \mu_A(x) : X \rightarrow [0,1]\}$$

Depending on the type of membership functions, different kinds of fuzzy sets will be obtained. Zadeh [3] proposed a number of membership functions that could be classified into two groups: Linear (i.e., Triangular, L, Gamma, Trapezoid, etc.) and Gaussian (such as Gaussian, Gamma, S, etc.). With respect to the characteristics of each function and specifications of each linguistic label, we can choose the appropriate function for a given label. For example, Gamma and S-function are commonly used for linguistic labels such as strong or high [20]. The equation (14) expresses the membership function of the fuzzy set hot in the form of the Linear Gamma-function, also see Figure 6.

$$\mu_{hot}(temp) = \begin{cases} 0 & temp \leq 25 \\ \dfrac{temp - 25}{10} & 25 < temp < 35 \\ 1 & temp \geq 35 \end{cases} \tag{14}$$

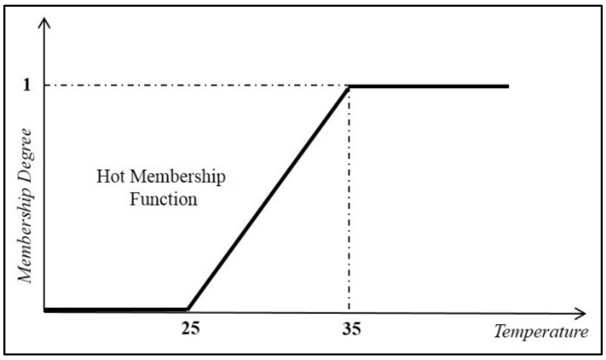

Figure 6: Fuzzy membership function for *hot*.

Definition 3- Fuzzy term: fuzzy term is a linguistic label such as hot that is identified through a fuzzy set such as hot which could have different membership functions in various times and places. For instance, there are some fuzzy terms such as cold, hot, moderate, warm, etc. which could be defined for the sensor's temperature attribute. Fuzzy terms are divided into two groups: simple and composite [22]. A simple fuzzy term is atomic, like hot, which is identified by a fuzzy set; while a composite (modified) fuzzy term, such as very cold, consists of fuzzy term with some modifiers *M* (Modifier = {*Very*, *Extremely*, *Fairly*, *Somewhat*, *More or Less*}). In composite fuzzy terms, the membership function is calculated through some rules which are shown below (equations (15)-(19)) [23].

$$\mu_{Extremely\ F}(v) = (\mu_F(v))^3 \tag{15}$$

$$\mu_{Very\ F}(v) = (\mu_F(v))^2 \tag{16}$$

$$\mu_{More\ or\ Less\ F}(v) = (\mu_F(v))^{1/3} \tag{17}$$

$$\mu_{Somewhat\ F}(v) = \mu_{Fairly\ F}(v) = (\mu_F(v))^{1/2} \tag{18}$$

$$\mu_{Not\ F}(v) = 1 - \mu_F(v) \tag{19}$$

Definition 4- Linguistic variable: a linguistic variable, is a variable like *temperature*, which its fuzzy value is determined through fuzzy terms such as *hot*, *cold* … and *warm*. In our proposed fuzzy query processing system, some sensor attributes are expressed by linguistic variables [24].

Definition 5- Fuzzy proposition: a fuzzy proposition consists of a linguistic variable and a fuzzy term, which its truth value can be true or false. The logic of fuzzy proposition is different from the logic has been defined for crisp proposition. Since the truth value of fuzzy proposition is a real number in the interval [0, 1], it has some degree of truth. This degree is related to the membership degree of the fuzzy set defined for fuzzy term. The canonical form of unqualified fuzzy proposition *P*, is *P: V is F*. Where *V* is linguistic variable such as pressure which could have a numeric value v from a universal set E_v, and F is fuzzy term which E_v is the domain of membership function defined for fuzzy term V. The truth value of the unqualified fuzzy proposition is calculated through equation (20)[16].

$$P \overset{def}{=} V\ is\ F$$
$$T_P = \mu_F(V = v), v \in E_V \tag{20}$$

For example, in the predicate "*temperature is high*", the truth value is calculated based on the membership function of fuzzy term high in Figure 7 and Figure 8.

$$T_{\text{temperature is hot}} = \mu_{hot}(temperature = 85) = 0.75$$

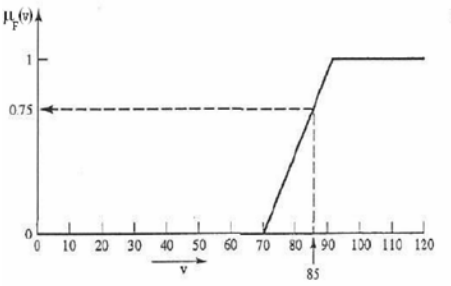

Figure 7: Membership function for hot.

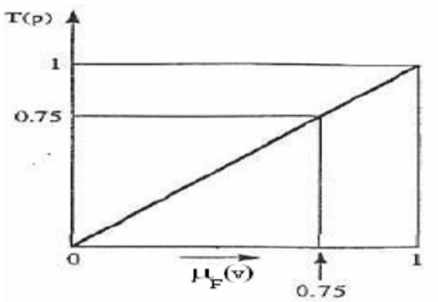

Figure 8: Truth value of unqualified fuzzy proposition