A Robust Outlier Detection Scheme for Collaborative Sensor Networks

L.S.Jayashree*
Computer Science and Engineering Department
Kumaraguru College of Technology
Coimbatore-641 006. India
jayashreeofkct@yahoo.co.in

S.Arumugam
Directorate of Technical Education
Chennai. India
s_arumugam@vsnl.net

K.Vijayalakshmi
Computer Science and Engineering Department
Kumaraguru College of Technology
Coimbatore-641 006. India
vijopani@yahoo.co.in

JDIM
Journal of Digital
Information Management

***ABSTRACT**: In-networks, Data Aggregation is usually warranted for distributed wireless sensor networks, owing to reliability and energy efficiency reasons. Sensor nodes are usually deployed in unattended and unsafe environments and hence are vulnerable to intentional or unintentional damages. Individual nodes are prone to different type of faults such as hardware faults, crash faults etc and other security vulnerabilities wherein one or more nodes are compromised to produce bogus data so as to confuse the rest of the network in collaborative sensing applications. The availability of constrained resources and the presence of faulty nodes make designing fault tolerant information aggregation mechanisms in large sensor networks particularly challenging. In our work, we consider Byzantine type of faults, which encompasses most of the common sensor node faults [9]. Faulty nodes are assumed to send inconsistent and arbitrary values to other nodes during information exchange process. These values are termed as outliers and we use a statistical test called Modified Z-score method to reliably detect and remove outliers. We show by simulation that the proposed strategy works well for 2 major classes of collaborative sensor network applications viz. (i) Target/ Event detection and (ii) Continuous data gathering.*

Categories and Subject Descriptors
C.2.1[Network Architecture and Design];Wirelss communication: **C.4[Performance of Systems];**Fault tolerance: **E.1**[**Data Structures**]; Distributed data structures

General Terms
Sensor networks, Data aggregation, System performance

Key words: Distributed information processing, fault tolerance, outliers, data aggregation, detection accuracy, false positives

1. Introduction

Wireless Sensor Networks (WSNs) are networks of tiny, battery powered sensor nodes with limited on-board processing, storage and radio capabilities [7]. Nodes sense and send their reports toward a processing center that is called a base station or a sink. Designing protocols and applications for such networks has to be energy aware in order to prolong the lifetime of the network. Sensor networks, once deployed, are left unattended and expected to work for extended periods of time. This is true under many real world application settings, rendering battery replacement out of question - the life of the battery decides the life of the network. Owing to the importance of the problem, there is a significant body of research addressing different aspects of power control problem. [7] gives a detailed survey on sensor networks and the open research problems.

In a typical target detection WSN application, individual sensor nodes collaborate with each other to perform a common task like detecting enemy tank movements in defense applications, detecting possible survivors in disaster rescue operations, tracking animal movements in habitat monitoring applications etc.

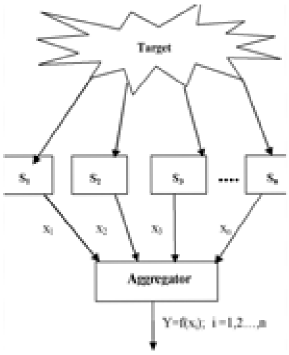

Figure 1. The Aggregation Process

Since each node has only a limited view of the sensing field and/or the sensing phenomenon, they send the sensed values to an aggregator as shown in figure 1. which then fuses/aggregates the collected reports from individual sensors and makes a higher-level decision regarding the presence/absence of the target of interest.

A WSN may be deployed in a potentially adverse or even hostile environment and potential threats may include depletion of batteries, accidental node failures, intentional tampering, failure of communication links and corruption due to noise. Therefore, sensor nodes have a high risk of being faulty. Inconsistent data can be reported by such faulty nodes, which can lead to false sensing reports (false positives/ false negatives). A system fails when its output deviates from the desired value. [17] identifies five main sources of error that influence performance results in WSNs.

When integrating sensor readings, robustness and reliability are crucial properties [14]. The presence of faulty sensor nodes affects the fusion process and can potentially corrupt the final result, thus requiring collaboration to be robust to node failures [17]. Since faulty nodes in a network can report inconsistently, thus misleading the other nodes in its neighborhood, dealing with such faulty nodes and making reliable decisions in the face of faulty nodes is a real challenging issue that needs proper investigation and hence taken as the subject of this paper. In this paper, we propose to use a statistical test called M*odified Z-score method* (Z-

* To whom all communications should be addressed

]

score test for large samples), a technique that uses median of absolute deviation about the median (MAD) to reliably label the outliers. This technique is well established in the field of statistics, which is based on robust regression to identify outliers in a normally distributed data. To the best of our knowledge, no reported work in sensor data aggregation problem makes a proper analysis of outliers and detects them. In [18]&[19], the authors suggest a technique to label the largest and smallest n values exchanged among sensor nodes as outliers and each sensor node drops those values before aggregation; here n is predetermined for a given number of sensor nodes. This may inadvertently cause dropping of legitimate values too, thus degrading the detection performance. We argue that making use of a well-established technique for identifying outliers would definitely improve the detection performance and we prove our hypothesis using simulation. We prove that the proposed method clearly outperforms [18]&[19] as discussed in section 5.

The remainder of the paper is organized as follows: In section 2 we present a brief review of related work. Section 3 gives a brief definition of outliers and elaborates the techniques to detect the outliers. The aggregation process is then explained in section 4. The results obtained are discussed in section 5 and section 6 ends the paper with conclusion and some discussion about future directions.

2. Related work

In a distributed system like WSN, when sensor nodes often cooperate to achieve a specified task, they often have to *agree* on a piece of data that is critical to subsequent computation. This is easily achieved in the absence of faulty nodes, for example by simple message exchange and voting. But special protocols need to be used to reach agreement in the presence of inconsistent faults**.**

The agreement problems are usually studied in terms of two broad categories namely, *consensus problem* and *Byzantine generals* problem.

The Byzantine generals problem is similar to a system of N nodes, some of which may be faulty. The faulty node not only makes wrong decisions but also attempt to make other nodes to disagree. The solution to this problem must ensure that, all fault free nodes agree among themselves on the content of a message received from node i, if node i is fault free.[9] presents a study about Byzantine generals problem and proposes the oral message algorithm to reach agreement among the fault free nodes. [9] proves that the algorithm is guaranteed to provide agreement only when at least two third of the total nodes are non faulty i.e., $N^{3}3t+1$ where N is the total number of nodes in which t nodes are faulty.

The *consensus problem* is concerned with reaching agreement on the system status by the non-faulty nodes in the presence of malicious nodes. [11] presents a detailed survey of 25 years of research on this problem and also classifies the node faults into various groups. The authors also study and compare system diagnosis and Byzantine agreement, which are two means to achieving consensus. The only difference is that system diagnosis identifies faulty nodes so that their impact may be avoided whereas Byzantine agreement uses protocols that mask any possible impact of faulty nodes. General techniques for reaching agreement are studied irrespective of the data being manipulated.

In applications, where processes hold an estimate of some global value, it may be sufficient to guarantee that the nodes agree on values that are not exactly identical but are relatively close to one another. This is known as the *approximate* or *inexact* agreement problem. [5] [10][11] present protocols for approximate agreement.

In distributed information fusion, important technical issues include the degree of information sharing between nodes and how nodes fuse the information from other nodes. There are many levels in which the sensed data can be shared and processed among nodes e.g. signal level, feature level and decision level [6]. At each of these levels, the information content is reduced, but this in turn reduces the required amount of data to be communicated between nodes. In short, processing is cheap and communication is expensive [6]. Therefore, one needs to consider the multiple tradeoffs between performance and resource utilization in collaborative signal and information processing using sensors.

Varying the size of the information shared between sensor nodes can derive different fusion algorithms. Two extreme cases are value fusion and decision fusion. The authors in [18] study the problem of collaborative target detection in a sensor network with and without faulty sensors using two methods viz. value fusion and decision fusion. They compare the two methods under various environmental conditions, particularly the number of faulty sensors on fusion performance. The paper [19] completes the preliminary study made in [18] and augments it with an analytical model to derive detection and false alarm probabilities. The authors in [19] prove that value fusion based algorithms perform better than decision fusion based algorithms in the absence of faults. However, in the presence of faults, both methods are claimed comparable. Though our work is based on [18] and [19] we use a more justifiable technique to detect the outliers present in sensor data and hence we get better detection performance as discussed in section 5**.**

The *influence field* of an object is the region within which the target is detectable by the sensors. [16] addresses the problem of deriving the necessary node density for reliably estimating the influence field of various object types. Their results can be made use of in estimating the subset of sensor nodes that are expected to participate in the aggregation process.

The authors in [3] propose Bayesian fault recognition algorithms that operate on sensor values transformed into binary equivalence for detecting and correcting faulty values in event detection applications. Their work is based on the conception that erroneous values are prone to be uncorrelated whereas correct values are spatially correlated. They show that the impact of faults can be reduced by as much as 85-95 percent for up to 10 percent of faulty nodes. Although our work does not take into account the spatial correlation of sensor values in identifying outliers, we work on actual sensor values and also the percentage of faulty nodes tolerated is much higher than 10 percent.

3. Detecting and accommodating outliers

3.1 Defining outliers

Outliers are the observations that appear to be inconsistent with the reminder of the collected data. The term outlier is used collectively for discordant observations and for contaminants. A discordant observation is defined as an observation that appears surprising or discrepant to the investigator [8]. A contaminant is defined as an observation from a different distribution then the rest of the data. Contaminants may or may not be noted by the investigator [2]. Possible sources of outliers are: recording and measurement errors, incorrect distribution assumption, unknown data structure, or novel phenomenon [8]. Recording and measurement errors are often the first suspected source of outliers. Incorrect assumption about the data distribution can lead to mislabeling data as outliers. In the context of distributed sensor networks, faulty nodes may deliberately introduce some malicious data so as to confuse the aggregation process. An attacker can either spoof numerous random or correlated data. Hence it can drastically change the aggregate data such as average, standard deviation etc.; it can also hide the extreme readings when genuine outlier (an extreme deviation from means) occurs.

3.2.Outlier test using Modified z-score method

The first step in data analysis is to label suspected outliers for further study. In a Modified z-score test, the z-score is determined based on outlier resistant estimators. The median of absolute deviation about the median (MAD) is such an estimator.

$$MAD = median \{|x_i - x_m|\} \qquad (1)$$

In z-score calculations, standard deviation is used to detect outliers whereas in Modified z-score, MAD is used in the place of standard deviation.

The method includes the following steps:

1. Calculate the sample median (x_m)
2. Calculate the absolute value of the difference between the observations and the median $|x_i-x_m|$
3. Calculate the median of the absolute deviation (MAD) about the sample median
4. Calculate the Modified z-score for each observation where z_i=0.6745*(x_i-x_m)/MAD
5. An observation is labeled an outlier when the $|z_i|$ is greater than 3.5

This is a reliable test since the parameters used to calculate the Modified z-score, are minimally affected by the outliers.

4. Details of the methodology

4.1 Sensor data aggregation model

In this section, we describe our proposed methodology. Our paper deals with aggregation of raw energy measurements obtained from individual sensor nodes to conclude on the status of a given target in target detection applications. We assume that all the nodes present in the influence field of a given target send their data to a common aggregator. This assumption is well suited for clustered sensor networks, wherein the cluster head can do the role of aggregator. Whenever an initiating event of interest occurs, each sensor node in the influence field takes a measurement and reports the observed value x_i to the aggregator. The aggregator's goal is to precisely detect and remove the faulty values and then compute an aggregate value y that summarizes the individual nodes' readings $x_1,\ldots,x_{n,}$ using an appropriate aggregation function f. Thus,

$$y = f(x_1,\ldots..,x_n). \qquad (2)$$

The aggregation is done to decide about the presence/absence of the target of interest. For the sake of comparison, we use the same aggregation function used in [18]&[19] i.e. the *arithmetic mean* of the values after removing the outliers. They propose a method in which the neighboring sensors exchange the measured values among themselves to reach an agreement. In order to alleviate the effect of outliers on the aggregation performance, they drop the largest and smallest n values of the data exchanged, before aggregation. The number of values to be dropped is fixed in advance for the given numbers of sensor nodes as given in table 1. Henceforth, we call it *drop extremes* method for brevity. But in the proposed work, we properly detect the outliers using a simple yet robust method that well suits to the resource-constrained nature of WSNs.

N	9	15	24	36	48	63	81	99
n		3	4	6	7	8	9	11 13

Table 1. Number of values to be dropped(n) for various values of N [19]

Another notable constraint in *drop extremes* method is its higher communication overhead which is O(k) for each sensor node, where k is the number of its neighbors. This is because their method requires the measured sensor values to be exchanged among the neighbours in order to reach consensus.

Thus, our approach differs from the former in,

(i) the way the outliers are detected and handled
(ii) eliminating the need for exchanging values among sensor nodes

Though our approach involves a slightly higher computational overhead i.e. O(nlogn), where n is the number of sensor data involved in the aggregation process, it offers a huge savings in communication cost by eliminating the need for data exchange. Note that in WSNs, the cost of transmitting a bit is many orders of magnitude higher than the cost of executing an instruction [13]. Thus, the slightly higher computational complexity is justifiable considering the savings in communication cost.

4.2 The aggregation process [20]

The aggregator performs the following steps:

1. Obtain raw energy values from the sensor nodes
2. Identify outliers using Modified Z-score and Z-score method
3. Remove the identified outliers
4. Compute the arithmetic mean of the remaining values
5. Compare the result of step 4 with the chosen threshold for final decision.

Generally, every target will have a signal energy range within which it is detectable*.* In this paper, we take the threshold to be the minimum of this range of energy and use this to measure the detection accuracy and the number of false alarms. The performance of the proposed approach is compared against the '*Drop extremes'* method in terms of two parameters namely, percentage of detection accuracy and the percentage of false alarms, for varying number of faulty nodes.

4.3 Data aggregation and outlier detection in continuous data gathering applications

We assume a clustered model of cluster size M in which each cluster member takes a measurement of a feature F (Temperature/light/pressure etc.), every T time units and a window of such measured values are sent to a common node for further aggregation. The window size is assumed as N. Under this setting, the measurement taken by a sensor node may produce outliers as in any of the following cases:

(i) A measurement error may occur in one or more sensor nodes leading to a faulty sensor reading in such nodes.
(ii) One or more sensor nodes may be tampered with, causing them to generate misleading values. These values may significantly deviate from the rest of the data distribution.

We label a value as an outlier if the difference between the $F_{current}$ and F_{pred} is greater than a threshold, where F_{pred} is extrapolated from the linear combination of the previously measured values as given in [1][4].

Assuming the above two possibilities, we now show how the modified z-score method employed at the sink node, is effective in removing the effect of outliers in a more precise way. If not attended to properly, these outliers will have a significant bearing on queries answered by the sink like range and median queries. The following graphs show the efficiency of modified z-score method in detecting the outliers present in the reported data.

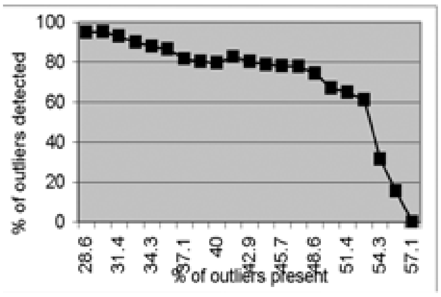

Figure 2.Performance of m-zscore for N X M=50

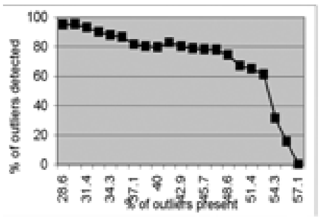

Figure 3. Performance of m-zscore for N X M =70

From the figures shown above, it is evident that the proposed outlier detection scheme resists up to 48% of faulty nodes and shows very good accuracy in correctly labeling the outliers. The scheme was tested for varying node densities and was found to give consistent performance. Two such instances are given in figures 2 and 3.

4.4 Time synchronization

All distributed systems need clock synchronization. In the scenario we have assumed, since data fusion takes place in a common aggregator, i.e. all nodes report to a single fusion point, they all need to synchronize with the fusion center and use the synchronized time to time-stamp all the data they send. Even in a hierarchical setup (multi hop network) as shown in figure 4, it is possible to extend this assumption, wherein, individual nodes in each level should synchronize only with the fusion nodes in the layer immediately above. Thus, the nodes s_1 and s_2 synchronize with s_4 in the next immediate layer, s_3 with s_5 and so on.

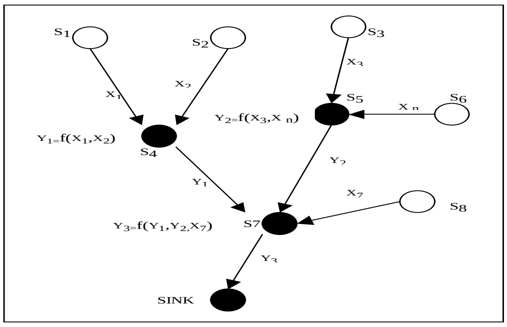

Figure 4. A multihop network with multiple aggregators

5 Performance evaluation

5.1 The simulation environment

To demonstrate the efficacy of this procedure we have simulated the scenario described in section 3, where N acoustic sensors (as per table 1) are uniformly distributed over a homogeneous region of 50X50 m, taking i.i.d. one-dimensional measurements corrupted by additive white Gaussian noise. We assume that the signal strength of an acoustic source measured at each sensor follows the model given below:

$$R_i = A.[D]^{-a}+w \qquad (3)$$

where,

R_i is the received signal strength of i th sensor
A is the strength of an acoustic signal from the target
D is the estimated distance between the target and the sensor node's positions
a is the attenuation coefficient
w is the white Gaussian noise

The zero mean Gaussian noise is generated with s^2=1 variance.

To measure detection accuracy, an acoustic target is placed in a random position. The target is made to emit a signal level of 90db. To measure the number of false alarms, no target is placed in the region. The Byzantine faulty behavior is generated as follows: In the absence of target, faulty nodes report high values and in the presence of target in the region, they all report low values. Simulations were repeated for variable number of faulty sensor nodes. The results shown in the graph are the averages of values obtained over 50 simulation runs.

5.2 Results and discussions

After collecting the measured values from all the associated nodes, the aggregator (i) drops the predetermined number of largest and smallest values (as shown in table 1) in *Drop extremes* method (ii) performs the outlier analysis and then drops the detected outliers in z-score and Modified z-score methods as discussed in section 3.

For the sake of completeness, we have also included the results obtained for z-score test. We now compare the performance of the three methods in terms of a) detection accuracy and b) number of false alarms.

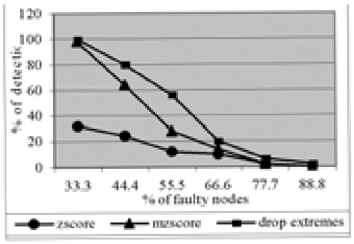

Figure 5. Detection accuracy for N=9

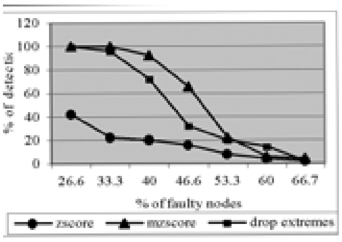

Figure 6. Detection accuracy for N=15

In table 1, N denotes the total number of nodes and n the corresponding number of largest and smallest values to be dropped. As shown in figure 5, when N=9, we find that the Drop extremes method gives slightly better performance than Modified z-score method whereas for all other values of N, the latter method is found to perform much better as evident from the graphs [Figs. 6-9]. When the total number of nodes is increased to 15 as shown in figure 6, the Drop extremes method guarantees detection up to 33% of faulty nodes,

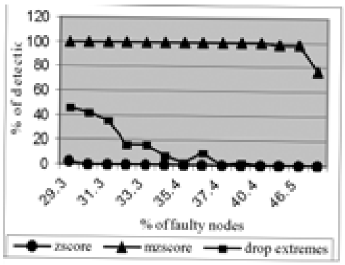

Figure 7. Detection accuracy for N=36

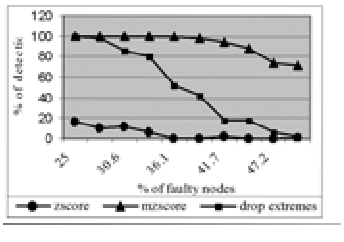

Figure 8. Detection accuracy for N=63

whereas Modified z-score method tolerates up to 40% of faulty nodes.

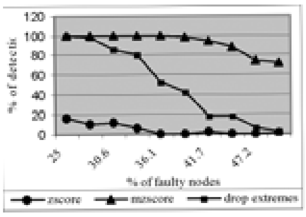

Figure 9. Detection accuracy for N=99

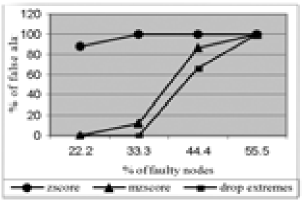

Figure 10. #False alarms for N=9

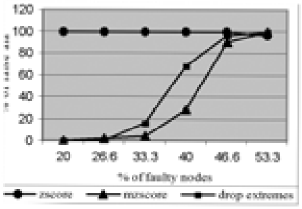

Figure 11. #False alarms for N=15

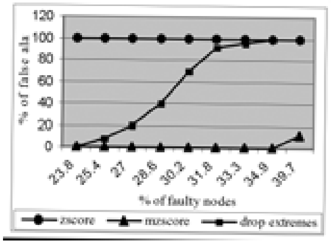

Figure 12. #False alarms for N=36

As the total number of nodes increases, the performance of Drop extremes method decreases whereas the detection accuracy of Modified z-score method substantially increases and remains stable i.e. the former method does not guarantee performance when the number of faulty values exceeds one third of N. But the proposed approach shows an average of 40% improvement in detection performance over Drop extremes method when the number of faulty nodes is 33% of the total number of nodes.

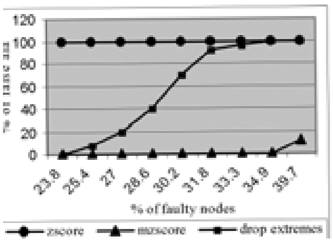

Figure 13. #False alarms for N=63

Next we discuss about the performance in terms of false alarms. As shown in figure 10, for a very low value of N, Drop extremes method offers a slightly better (12%) performance than the proposed approach. However, in all other cases, we find that the latter performs far better than the former. Thus when the number of faulty nodes reaches one third of the total node density, an average improvement of 48% reduction in false alarms was achieved when using Modified z-score method compared to Drop extremes method.

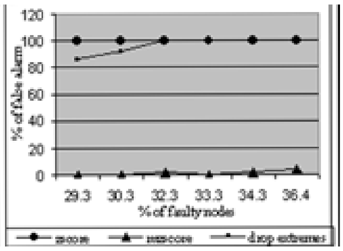

Figure 14. #False alarms for N=99

But in all the above cases, the z-score method shows no resistance to even 22% of faulty nodes**.** This is because it makes use of mean and standard deviation for outlier detection both of which are affected by outliers. Thus we conclude that this method is not a reliable one for target detection applications. The accuracy of detection is of crucial importance to most of the real world target detection applications. By means of precisely identifying the outliers in sensor data, our method clearly showcases its potential in improving the aggregation performance of collaborative applications. For easy empirical comparison, the graphs in figures 5-9 are presented in the form of table 2 given below.

No. of nodes (N)	Detection accuracy (%)											
	z-score				**Modified z-score**				**Drop extremes**			
	% of faulty nodes				% of faulty nodes				% of faulty nodes			
	33.3	**40**	**45**	**50**	**33.3**	**40**	**45**	**50**	**33.3**	**0**	**45**	**50**
9	32	-	24	18	98	-	64	48	100	-	80	68
15	22	20	16	12	100	92	66	42	96	72	32	26
36	6	0	0	0	100	98	88	72	80	42	18	2
63	0	0	0	0	100	100	94	80	20	0	0	0
99	0	0	0	0	100	100	98	76	16	0	0	0

Table 2. Comparison of detection performance of the three methods for varying number of faulty

5.3 Generality and scalability of the proposed approach

The major factor limiting the performance of any aggregation mechanism is the proportion of faulty values present during the aggregation process. Almost all the sensor network applications fall under either one of the following classes based on the data delivery model employed: (i) *Target/ Event detection (ii) Query driven* and *(iii) Continuous data gathering* [15]. All these are collaborative in nature and the proposed aggregation strategy is directly applicable for any of the above class of collaborative applications, which is evident from the performance graphs given in sections 4.3 and 5.2.

The simulation results show that the method scales well with increase in node density. Actually, the computational complexity of the aggregation process grows linearly with the size of the network. Yet, in a real setting, this can be managed by enhancing the processing power and storage capacity of the aggregator. Alternatively, multiple aggregators can be employed to share the load of fusing the reported sensor readings.

Essentially, the node doing the role of aggregator should have some special-purpose hardware that contains the in-built logic to perform the steps involved in modified z-score computation. Also, the aggregator is expected to have higher processing power and storage capacity than individual nodes. The application and the nature of the instrumented area decide whether the aggregator is wired/wireless. On the other hand, any commercially available class of sensor nodes can be employed as a cluster member with no modifications required in node hardware (e.g. Mica Mote).

6. Conclusion

This paper investigated the possibility of using a simple and effective statistical technique called Modified z-score method in distributed sensor networks applications that are collaborative in nature for the purpose of outlier detection. Outliers are normally discarded from majority of data. However, due to the unpredictable nature of observed phenomena, simply dropping fixed number of values may inadvertently cause losing of important observations. Owing to this simple observation and aiming towards improving the aggregation performance, we used a more justifiable method to precisely identify outliers in the sensor data and the results obtained are promising. Simulated experiments demonstrated the potential of the proposed approach in target detection and data gathering applications.

However, some of the improvements, we propose are the following:

(i) the current work does not fully make use of the processing power available at individual sensor nodes, as they are used only for taking measurements and reporting it to the fusion point. Local analysis of the measured values could be done to leverage the processing potential of sensor node and also to reduce the communication overhead.

(ii) current work used a single aggregator. Though the aggregator is assumed fault-free, it still suffers from the problem of single point of failure. One plausible solution to alleviate this effect is to make the aggregation process fairly distributed i.e., a set of leader nodes may be assigned the role of aggregator and they would all exchange their aggregation results among themselves to reach consensus. Faulty aggregator(s), if any, would possibly report inconsistent results but would be disregarded during a *majority voting* process towards making the final decision.

(iii) It should also be noted that significant deviation in the sensed values might also occur due to a *transient* or *persistent* change in the phenomenon being monitored. We are currently exploring the spatio-temporal correlation among the nodes to differentiate between the cases where outliers are produced by faulty measurements/nodes and those that are produced due to a fundamental shift in the phenomenon, which is a subject of the future work.

7. References

[1] Deshpande, Amol., Guestrin, Carlos., Madden, Samuel., Joseph, R., Hellerstein, M., Hong, Wei (2004). Model-based Approximate Querying in Sensor Networks. VLDB.

[2] Barnett, V., Lewis, T. (1984). Outliers in Statistical Data. New York : John Wiley & Sons.

[3] Krishnamachari, Bhaskar., Sitharama Iyengar (2004). Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks, *IEEE Transactions On Computers,* 53(3).

[4] Tulone, Daniela., Madden, Samuel (2006). PAQ: Time series forecasting for approximate query answering in sensor networks. *In:* Proc. of the 3rd European Conf. Wireless Sensor Networks. Feb.

[5] Dolev, D., et al. (1986). Reaching Approximate Agreement in the Presence of Faults. *Journal of ACM*.

[6] McErlean, Donal., Narayanan, Shrikanth (2002). Distributed Detection And Tracking In Sensor Networks, *In*: 36th Asilomer Conf. on Signals systems and computers.

[7] Akylidiz, Ian F., Shankarasubramaniam, (2002). A Survey on Sensor Networks. *IEEE Communications Magazine*.

[8] Iglewicz, B., Hoaglin, D. C (1993). How to Detect and Handle Outliers, American Society for Quality Control. Milwaukee. WI.

[9] Lamport, L., Shostak, R., Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*. 4 (3) 382-401.

[10] Mahaney, S., Schneider, F (1985). Inexact Agreement: Accuracy, Precision and Graceful Degradation. *In*: Proceedings of the Fourth ACM Symposium Principles of Distributed Computing.

[11] Barborak, Michael, Miroslaw., Dahbura, Anton (1993). The Consensus Problem in Fault-Tolerant Computing. *ACM Computing surveys.* 25 (2).

[12] Shrivastava, Nisheeth., Buragohain, Chiranjeeb., Agrawal, Divyakant., Suri, Subhash (2004). Medians and Beyond: New Aggregation Techniques for Sensor Networks. *In*: SenSys'04. November 3–5. Baltimore. Maryland. USA.

[13] Pottie, G.J., Kaiser, W.J(2000). Wireless Integrated Network Sensors. *Communications of the ACM*. 43 (5) 51–58. May.

[14] Richard, R. Brooks., Sitharama Iyengar, S., (1996). Robust Distributed Computing and Sensing Algorithm. *IEEE Computer Magazine.*

[15] Tilak, Sameer., Abu-Ghazaleh, Nael B., Heinzelman, Wendi (2002). A Taxonomy of Wireless Micro-Sensor Network Models. *Mobile Computing and Communications Review*. 6 (2).

[16] Bapat, Sandip., Kulathumani, Vinodkrishnan., Arora, Anish (2004). Reliable Estimation of Influence field for Classification and Tracking in Unreliable Sensor Networks. (OSU-CISRC-8/04-TR49). Columbus, The Ohio State University.

[17] Slijepcevic, Sasha ., Seapahn, Megerian., Potkonjak, Miodrag (2002). Location Errors in Wireless Embedded Sensor Networks: Sources, Models, and Effects on Applications. *Mobile Computing and Communications Review* 6 (3).

[18] Clouqueur, Thomas., Ramanathan, Parameswaran Saluja, Kewal K., Wang, Kuang-Ching (2004). Value-Fusion versus Decision-Fusion for Fault-Tolerance in Collaborative Target Detection in Sensor Networks. *IEEE Transactions on Computers*. 53.

[19] *Ibid*.

[20] Vijayalakshmi, K., Jayashree, L.S., Arumugam, S (2006). An Efficient and Fault Tolerant Aggregation Scheme for Distributed Sensor Networks using Modified Z-score Method. *Intl. Journal of Systemics, Cybernetics and Informatics*.

L.S.Jayashree completed her Bachelors degree in Electronics and Communication Engineering and Masters degree in Computer Science and Engineering in Govt. College of Technology in 1995 and 1997 respectively. She has over 10 years of academic experience. Her specialization includes Computer Networks Performance Optimization, Mobile and Pervasive Computing and Wireless Sensor Networks. She has about 20 publications in various national and international journals and conferences to her credit.

Vijayalakshmi Kothandapani obtained her Bachelor of Engineering (BE) degree in computer science and engineering from the Bharathiar University, India, in 2002. She is currently a PG student pursuing M.E. [CSE] degree from the Anna University, India. Prior to this she had been a Lecturer in the Department of computer science and engineering, Sri Krishna College of Engineering and Technology, Coimbatore, India, for two years. Her research interests include problems related to fault tolerant computing and sensor networks.

S.Arumugam completed his Bachelors degree in Electronics & Communication Engineering and and M.Sc(Engg) in Applied Electronics, both from PSG College of Technology under University of Madras and Ph.D in Computer Science & Engineering from Anna University. He has been serving in the Directorate of Technical Education since 1974 onwards. He is also serving as a member of various boards of studies. He is a senior member of IE, CSI, IETE and ISTE. He has so far published over 60 papers in various National, International journals and conferences.