Ashu Saini al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 1, Issue 2, Sept 2014, pp. 65-69

# A New Approach for Outlier Detection in WSNs

**Ashu Saini[1], Dr. Kamal Sharma[2], Sourabh Budhiraja[3]**

[1]Student, M. Tech, ESEAR, Ambala
[2]Professor, Dept. of ECE, E-Max group of Institutions, Ambala
[3]Assistant Professor, Dept. of CSE, E-Max group of Institutions, Ambala

***Abstract*— The problem of determining faulty readings in a WSN without compromising detection of important events will be studied. A correlation network will be built based on similarity between readings of two sensors. We will try to calculate the rank of the each sensor on the basis of correlation. In light of this SensorRank, an efficient in-network voting algorithm will be used to determine faulty sensor readings. To make SensorRank energy efficient, we will use clustering in which CH collect the outlier data from its cluster and send it to the Base Station. Now this Base Station aggregates the data and this data is sent to different clusters. Performance studies are conducted via simulation.**

***Keywords*— WSN, Outlier detection, Data mining.**

## I. INTRODUCTION

The term outlier, also known as anomaly, originally stems from the field of statistics (Hodge and Austin, 2003). The two classical definitions of outliers are: (Hawkins 1980): "*An outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism*".
Additionally outliers can be defined as, "*those measurements that significantly deviate from the normal pattern of sensed data*" [13]. This definition is based on the fact that in WSN, SNs are assigned to monitor the physical world and thus a pattern representing the normal behavior of sensed data may exist. Potential sources of outliers in data collected by WSNs include noise & errors, actual events, and malicious attacks.

## II. RELATED WORK

The literature of outlier has been studied in detail in [2]. Here is some related work.

### A. TECHNIQUES DESIGNED FOR WSN

In this section, we provide a technique-based taxonomy framework to categorize these techniques.
As illustrated in Figure 1, outlier detection techniques for WSNs can be categorized into different categories.

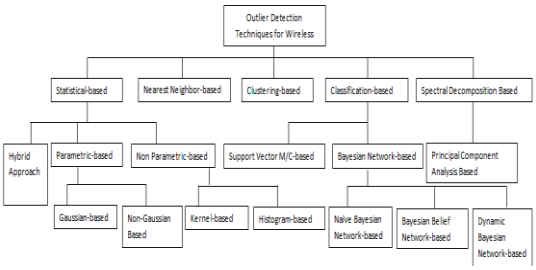


Figure 1: Taxonomy of Outlier Detection techniques for WSNs

### B. CLASSIFICATION OF THE STATE-OF-THE-ART OUTLIER DETECTION

Outlier detection techniques can be classified based on the following approaches used for outlier detection:

1. Distance based
2. Normal state model based (Machine Learning based)

### C. ANALYSIS OF DISTANCE BASED OUTLIER DETECTION TECHNIQUES FOR WSN

Distance based techniques, as shown in Figure 3, use some distance measure from statistical distribution, nearest neighbor or clusters to detect the outliers. We classify the distance based techniques into the following types:

- Statistical based
- Nearest Neighbor based
- Clustering based

This technique considers multivariate data and uses spatio-temporal correlations to detect local outliers.

- DFD Approach
- Reliable and Energy efficient schedule (REEF)
- History based approach
- Non-history based Approach
- Modified z-score method
- Classical majority voting
- Distance weighted voting
- An Interleaved Hop-by-Hop detection Scheme
- Statistical En Route Filtering (SEF) Scheme
- Commutative Cipher Based En-route Filtering (CCEF) Scheme
- Distribution-based method
- Using SensorRank Scheme

## III. PROPOSED WORK

### A. PROBLEM DEFINITION

Outlier detection in WSNs is needed in many applications that monitor abnormal behaviors, measurements, and events. For example, a sensor network, embedded in a highway bridge around beams or columns for detailed building structural monitoring, can give early warning of any structural weakness or deterioration, reducing the chance of unexpected failures. Outlier detection helps pinpoint the accurate locations of the weakening parts, especially in the early stage of the problem development. Chemical sensors deployed in the environment to monitor toxic spills and nuclear incidents gather the chemical data periodically. Outlier detection can trigger the alarm and locate the source

 

when abnormal data are generated. Habitat monitoring for endangered species is another application in which animals will be attached with small non-intrusive sensors. Outlier detection indicating abnormal behaviors suggests closer observation of an individual animal and maybe more human interactions.

B. OBJECTIVES

In this research we study few of outlier detection techniques to identify outlier in WSN. The summarized of goal of work for the dissertation as follows.

- The objective of our work is to find an Energy Efficient based method for outlier detection in WSN.
- As the SNs are limited in energy so we will try to use cluster head (CH) to save all the outlier detection reading found in the cluster.

Main focus will be to find the Inter-Cluster Outlier.

We will use MATLAB for simulating our work. MATLAB is widely used in all areas of applied mathematics, in education and research at universities, and in the industry. MATLAB stands for MATrix LABoratory and the software is built up around vectors and matrices.

C. PROTOCOL ASSUMPTIONS

In trust voting algorithm we have assumed that the sensors are distributed in a uniformly randomized manner throughout a field and the network has the following properties:

1. There exists a unique BS located away from network
2. Each SN has a unique identity
3. Sensors cannot move after being deployed

We propose a simple, static fault detection model which will improve the concept of SensorRank. In [3], the problem of determining faulty readings in a WSN without compromising detection of important events was studied. By exploring correlations between readings of sensors, a correlation network was built based on similarity between readings of two sensors. By exploring Markov Chain in the network, a mechanism for rating sensors in terms of the correlation, called SensorRank, was developed. In light of SensorRank, an efficient in-network voting algorithm, called TrustVoting, was proposed to determine faulty sensor readings. But to make SensorRank energy efficient we make use of clustering. The Concept of clustering is taken from LEACH [4]. Before we discuss about the concept of clustering we need to compute the value of SensorRank first.

D. LOW ENERGY ADAPTIVE CLUSTERING HIERARCHY IN THE WSN (LEACH)

To make above model energy efficient we make use of the concept of clustering. LEACH [4] was developed for application where there are SNs periodically collecting scalar data and send them to BS for further analysis. The physical scalar sensor measurements are processed by means of existing models or methods, with the aim of predicting the occurrence of events, such as flooding, fire or intruders. LEACH considers a network with the following characteristics:

- The SNs fixed, are energy-constrained and they have the same capability;
- The BS has not subject to energy restrictions and is located inside the sensing field;
- There is no batteries recharge after node deployment;

This general scenario may be used for various applications ranging from civilian and military areas. For example, monitoring in rainforest area to measure environmental factors, such as: temperature, humidity, and wind speed.These information can be used to predict event occurrence.

1. CH election

As mentioned earlier, in hierarchical architectures, the nodes are divided into clusters and a set of nodes is periodically elected as a CH. CHs are used for more complex tasks, such as: the management of each cluster, collecting data from non-CHs, data aggregation, and sending the collected data to the BS. In this context, it is important to use multiple metrics for CH election to provide an energy-efficient and load balance model. Furthermore, the cluster formation process can lead to poor energy use, if the CHs that are elected are only based on a single metric. In this context, LEACH proposes an equation, which is used by nodes to enable them to become a CH.

After adjusting the transmission power, each node generates a random number (μ), which ranges from 0 to 1. Then, the node decides to become a CH by comparing μ with the T(n), which is computed by means of Equation 2. If μ is less than T(n), the node becomes a CH for the current round.

$$T(n)=\frac{p}{1-p\left(r \bmod \frac{1}{p}\right)} \qquad (1)$$

Equation 1 uses a gauss function, due to the fact that has better result in terms of energy efficiency and representation in the context of an imprecise environment.

2. Cluster Formation

During this sub-phase, non-CHs select the best CH by considering a metrix, i.e. distance from non-CH to CH. The non-CH chooses the CH with a higher probability value and sends a *join message* to CH.

E. SENSORRANK

In the early work in the field, distances between SNs were taken into consideration when modeling the correlation of sensor readings. However, it is also possible that the readings of two geographically close SNs to have dramatically different readings. Thus, it's critical to truly capture the correlation of sensor readings rather than their distance. So a Correlation network is to be maintained for sensor readings.

The correlation network is modeled as a graph G = (*V*;*E*), where *V* represents the SNs in the deployment region and $E = \{(s_i; s_j) | s_i, s_j \in V; \text{dist}(s_i; s_j) < R \text{ and } \text{corr}_{i,j} > 0\}$. The weight of an edge $(s_i; s_j)$ is assigned to be $\text{corr}_{i,j}$.

1. SensorRank Calculation

SensorRank is to represent the trustworthiness of SNs. By our design, two requirements need to be met in deriving SensorRank for each sensor.

Requirement 1: If a sensor has a large number of neighbors with correlated readings, the opinion of this sensor is trustworthy and thus its vote deserves more weight.

Requirement 2: A SN with a lot of trustworthy neighbors is also trustworthy.

These two requirements ensure that:

1. A SN which has a large number of similar neighbors to have a high rank.

 

2. SN which has a large number of 'good references' to have a high rank. Given a correlation network G = (*V*; *E*) derived previously, we determine SensorRank for each sensor to meet the above two requirements. Based on the above setting, we can formulate SensorRank of $S_i$ , denoted as $rank_i$, as follows:

$$p_{j,i} = \frac{corr_{i,j}}{\sum_{k \in nei(i)} corr_{i,k}}$$
$$rank_i = \sum_{S_j \in nei(i)} p_{j,i}.rank_j \qquad (2)$$

where$p_{j,i}$ is the transition probability from state i to state j.
With the help of clustering instead of SensorRank exchange their rank to each other CH collect the sensor ranks of all SNs thereby energy is saved which is our main motive.

F. TRUSTVOTING ALGORITHM

Here we describe the TrustVoting algorithm, which consists of two phases: a) self-diagnosis; and b) neighbors diagnosis phase. In the self-diagnosis phase, each sensor verifies whether the current reading of a sensor is unusual or not. Once the reading of a sensor goes through the self- diagnosis phase, this sensor can directly report the reading. Otherwise, the SN consults with its neighbors to further validate whether the current reading is faulty or not. If a reading is termined as faulty, it will be filtered out.

1. Self-diagnosis Phase

When a set of SNs is queried, each sensor in the queried set performs a self-diagnosis procedure to verify whether its current reading vector is faulty or not. Once the reading vector of a SN is determined as normal, the SN does not need to enter the neighbor-diagnosis phase. To execute a self-diagnosis, each sensor $s_i$ only maintains two reading vectors: i) the current reading vector at the current time *t* (denoted as $b_i(t)$); and ii) the last correct reading vector at a previous time $t_p$(expressed by $b_i(t_p)$).$b_i(t_p)$ records a series of readings occurred in the previous time and is used for checking whether the current reading behavior is faulty or not. If these two reading vectors are not similar, $b_i(t)$ is viewed as an unusual reading vector. Once a SN is detected an unusual reading vector, this SN will enter the neighbor-diagnosis phase Note that when $b_i(t)$ is identified as a normal vector through the neighbor-diagnosis, $b_i(t_p)$ is updated so as to reflect the current monitoring state.

2. Neighbor-diagnosis Phase

If a SN $s_i$ sends $b_i(t)$ to a neighbor $s_j$, $s_j$ will compare $b_i(t)$ with its own current reading vector $b_j(t)$ and then give its vote with respect to $b_i(t)$. From the votes from neighbors, $s_i$has to determine whether $b_i(t)$ is faulty or not. Notice that some votes are from sensors with high SensorRank. A SN with high SensorRank has more similar neighbors to consult with and thus is more trust- worthy. Therefore, the votes from the neighbors with high SensorRank are more authoritative, whereas the votes from the neighbors with low SensorRank should cast less weight. When sensor $s_i$ sends $b_i(t)$ to all its neighbors for the neighbor-diagnosis, each neighbor should return its vote after determining whether $b_i(t)$ is faulty or not. If a neighbor $s_j$ considers $b_i(t)$ is not faulty by comparing the similarity of the two reading vectors (i.e., $corr_{i,j} \geq \sigma$)$s_j$ will send a positive vote, denoted $vote_j(i)$, to $s_i$. Otherwise, the vote will be negative. In addition, the vote from $s_j$ will be weighted by its SensorRank.

$$vote_j(i) = \begin{cases} rank_j, & corr_{i,j} \geq \sigma \\ -rank_j, & otherwise \end{cases}$$

After collecting all the votes from the neighbors, $s_i$has two classes of votes: one is positive class ($b_i(t)$ is normal) and the other is negative class ($b_i(t)$ is faulty). If the weight of the former is larger than the weight of the later, the most neighbors will view $b_i(t)$ as normal. Note that the weight of a vote represents how authoritative a vote is. It is possible that a neighbor $s_j$ of $s_i$ with a large SensorRank has a small correlation with $s_i$. In this case, these two SNs may not provide good judgments for each other. Therefore, each vote (i.e., $vote_j(i)$) has to be multiplied by the corresponding correlation, $corr_{i,j}$. Thus, we use the following formula to determine whether the reading is faulty or not.

$$dec_i = \sum_{S_j \in nei(i)} corr_{i,j}.vote_j(i)$$

If the weight of the positive votes is more than the weight of the negative votes, $dec_i$ will be positive which means that $s_i$'s reading is normal and the current reading can be reported. Otherwise, $dec_i$ is negative, implying that the current reading of $s_i$ is faulty.

Each CH find the outlier nodes with in the cluster, it will send data to the BS. BS will aggregate the data and send the aggregated outlier data to every cluster. Now each CH has aggregated outlier data of every other cluster. So whenever there is an Inter-Cluster communicates within the network, CH will check the local aggregated outlier data.

In this way we can detect the Inter-Cluster outlier nodes.

We are combining two techniques i.e. SensorRank [3] & LEACH [4].As per the result of these techniques, these two techniques are energy efficient. So we can say we will have an energy efficient system.

The Pseudo code of Proposed Model is as Follows:

Step1: Start

Step 2: Create a Network

Step 3:Create Clusters from network using LEACH [4].

a. A CH is selected from the SNs.
b. Based on last step, Non-CHs select the best CH by considering a metrix i.e. distance from non-CH to CH and Cluster is created.

Step 4: Rank of each node is calculated using SensorRank [3].

Step 5: With SensorRank, TrustVoting algorithm [3] is used which consists of two phases:

a. Self-diagnosis: performs a self-diagnosis procedure to verify whether its current reading vector is faulty or not.
b. Neighbor diagnosis phase: The votes from the neighbors are taken. Vote with high SensorRank are more authoritative, whereas the votes from the neighbors with low SensorRank should cast less weights.

Step 6: If a neighbor with a large SensorRank has a small correlation node, they may not provide good judgments for each other. Therefore, each vote correlation, $corr_{i,j}$, following formula is used to determine whether the reading is faulty or not.

 

$dec_i = \sum_{S_j \in nei\,(i)} corr_{i,j} . vote_j\,(i)$

if $dec_i$= +ve, node's reading is normal.

Otherwise, $dec_i$=-ve, implying that the current reading of node is faulty.

Step 7: Collection of outlier data within the cluster using CH, it will send data to the BS.

Step 8: Aggregated data from the BS can be forwarded to every cluster.

Step 9: Stop

## III. RESULTS

### A. Simulation Scenario

The presented model can be used for large WSNs, where network is divided into clusters. Initially there is a network in which nodes are distributed randomly as shown in Figure 2.

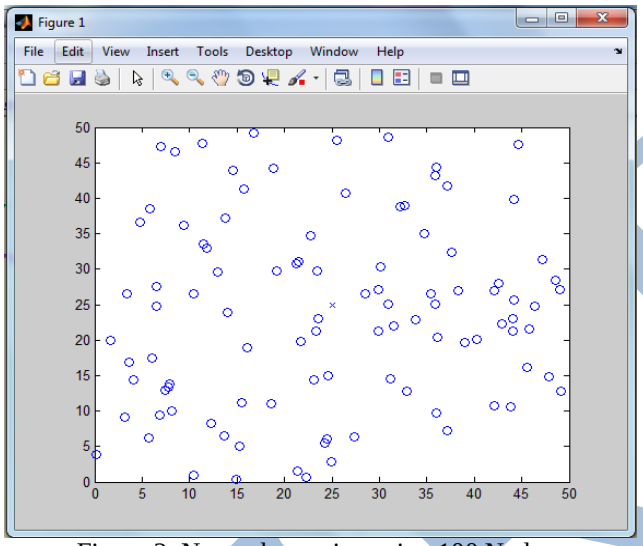


Figure 2: Network creation using 100 Nodes.

Now Clusters Heads are elected from the given network and clusters are made based on [4]. Figure 3 shows dark blue stars (*) which are marked as Cluster Heads.

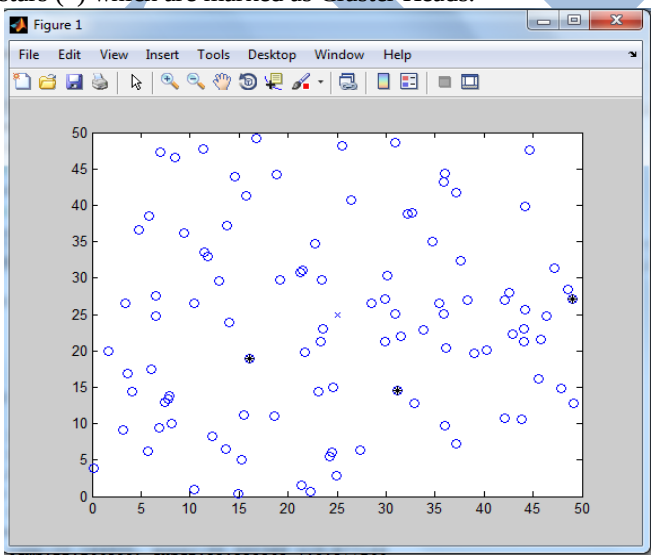


Figure 3: Election of Cluster Head

Each Normal node will elect its cluster head based on LEACH [4]. Now next simulation is for Sensor Rank Calculation and Outlier Detection. Figure 5.3 shows the Red colored (*) Faulty Nodes found in the network in each cluster.

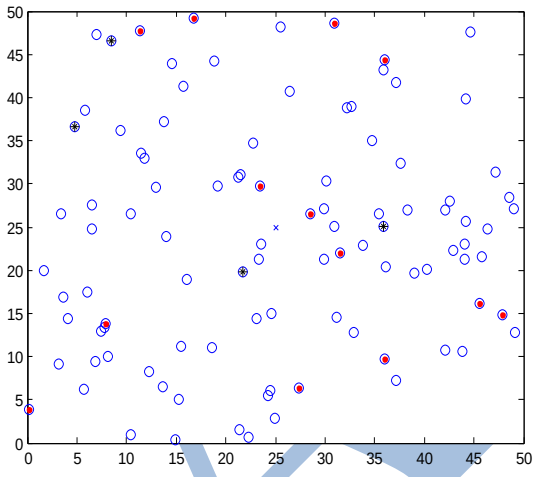

Figure 5.3: Faulty Nodes found in the network in each cluster.

### B. Performance Evaluation

The basic parameters used for simulations are listed in table 1.

Table 1: Parameters employed in Simulation

| Parameter | Value |
|---|---|
| Field Size | 50m X 50m |
| Location of Base Station | 25m X 25m |
| No. of Nodes | 100 |
| Probability of cluster | 0.1 |
| Initial Energy of sensor node | 20 J |
| The Data packet Size | 4000 bits |
| DeltaT | 10 |
| MinReading | 1 |
| maxReading | 10 |
| Efs | 10 J/bit/m2 |
| Emp | 0.0013 J/bit/m4 |

Based on these parameters author will carry out the simulations. These parameters are taken after studying different research papers used in Wireless sensor network.

Figure 5 showing outlier/Faulty Nodes detected in each round of simulation. It shows that outlier per round is more in Hybrid Technique as compared to old technique.

 

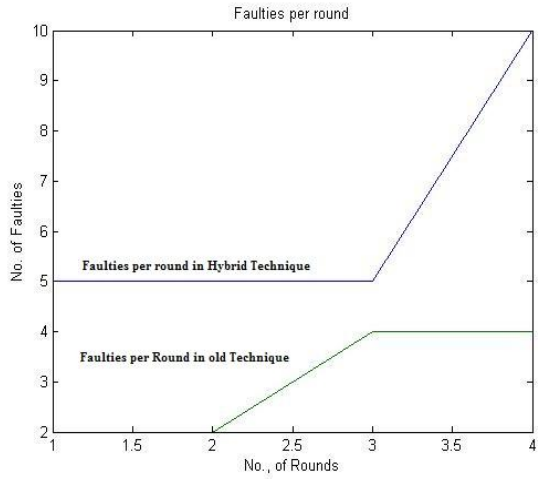


Figure 5: Outlier/Faulties per Round

Finally Figure 6 showing the energy consumption (joule) after completion of each round while finding the faulty node in a network. It shows that consumption of energy is very low after each round, which shows that the life time of network will increase.

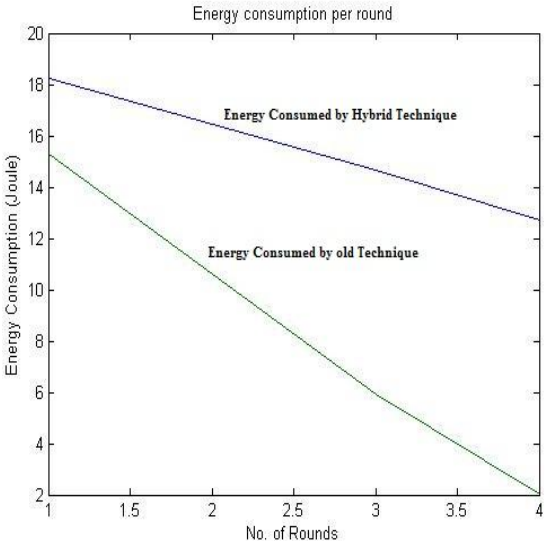


Figure 6: Energy Consumption per round

## IV. CONCLUSION

We applied our ideas into the SensorRank scheme and finally we achieved an efficient mechanism which is based on clustering. The cluster head of each cluster detect Faulty nodes from cluster and send this information to the Base Station. In this manner Base Station have information of all Faulty nodes in the network. Then Base Station can broadcast this information to all cluster heads. In this way Inter cluster outlier detection can be done. As each cluster head is associated in finding the faulty nodes, which results detection of faulty nodes in WSNs with low energy consumption. Energy consumption is affected by message communication between nodes, so our technique is efficient than traditional SensorRank scheme.

## REFFERENCE

[1] Chandola, V., Banerjee, A. and Kumar, V., "Outlier detection: a survey", Technical Report, University of Minnesota, 2007.

[2] Xiang-Yan Xiao, Wen-ChihPeng and Chih-Chieh Hung," Using sensor rank for in-network detection of faulty nodes in WSNs", in proceedings of MobiDE'07, pp. 1-10, June 2007.

[3] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Hawaii International Conference on System Sciences, Maui, Hawaii, January 4-7, 2000.

 