Journal of Communications Technology, Electronics and Computer Science, Issue 5, 2016
ISSN 2457-905X

Information Theory Based Intrusion Detection in Wireless Sensor Networks

Nancy Alrajei[1], Huirong Fu[1], Fatma Mili[2,] Ye Zhu[3]

[1]*Oakland University, Computer Science and Engineering Department, Rochester, Michigan 48309*
[2]*Purdue University, Computer and Information Technology Department, West Lafayette, Indiana, 47907*
[3]*Cleveland State University, Electrical and Computer Engineering Department, Cleveland, Ohio 44115*
fu@oakland.edu

***Abstract*— Sensor networks are used for monitoring purposes in different environments. One of the biggest issues is to keep the network alive as long as possible. Another concern is to keep it safe from attacks. The limitations of sensor nodes make them particularly vulnerable to attacks from adversaries. The most damaging type of attack is Denial of Service (DoS) attack where parts of the network are overloaded with a flood of requests forcing them to deplete their power and die early. In this paper, we introduce a set of metrics by which intruders are identified among the other nodes. This approach is characterized by the fact that identification of intruders is based on the intrinsic behavior that is either harmful or not beneficial to the network. At the same time our approach saves the network power by taking advantage of network redundancy, and query minimum number of nodes without affecting the accuracy of the results. We tested different intruder detection metrics to see if we can accurately find intruders in the sensor network and how early to save the network from damage. Our results show the effectiveness of these metrics in detecting intruders with 100% accuracy and 0 error rate from some of them.**

Keywords—intrusion detection; wireless sensor network; metric; usefulness; usability; utility; power consumption convergence.

I. INTRODUCTION

A sensor network is a collection of low cost, small form factor, embedded devices called sensor nodes. Sensor network can provide access to information anytime, anywhere by collecting, processing, analyzing and disseminating data. Sensor nodes are great for deployment in hostile environments or over large geographical areas. This exposes them to attackers who capture and reprogram individual sensor nodes. Once in control of a few nodes inside the network, the adversary can extract private sensed information from sensor network readings [1].

Wireless sensor networks could be deployed in both civil and military applications such as volcanic eruption monitoring, target monitoring, security and remote surveillance [2]. Their deployment in remote and frequently hostile environments combined with the device constraints, makes them particularly vulnerable to Denial of Service (DoS) attacks from adversaries. Since no single node detains critical or private information, the most damaging type of attack is (DoS) attack where parts of the network are crashed or overloaded with a flood of requests forcing them to deplete their power and making them non-available for their primary function which is monitoring. They are especially vulnerable to these kinds of attacks because of their lack of a fixed infrastructure and their limited power, memory, and computation resources.

These networks are often deployed unattended for long periods of time. Therefore, it is important to guard against malicious outside behaviours. Typically, the security protection of networks consists of a collection of complementary tools and methods. The first line of protection consists of firewalls, which are "fences" built around the system directing all communication towards a small number of guarded gates. If an intruder succeeds in crossing the fence, a firewall is no longer useful, thus there is the need for intruder detection.

Intrusion detection is the process of discovering, analysing, and reporting unauthorized access, or damaging network or computer activities. It discovers violations of confidentiality, integrity, and availability of information and resources [3].

Existing approaches to intrusion detection know about patterns of past intruders. When dealing with malicious intrusions, the network is constantly at risk of new enemies that may use a different pattern from the ones in the catalogue.

Approaches based on the cataloguing of patterns associated with intruders suffer from a fundamental flaw: they only know about patterns of past intruders. Moreover, they depend on the extrinsic attributes like node authority, identity etc., and this characterization may result in false positives and false negatives.

False positive identification: Innocent nodes that behave in a way similar to that of intruder nodes are flagged based on assumed intentions.

False negative identification: Harmful intruders behave in a smart way and go undetected causing harm to the network.

Intrusion detection can be based on detecting any significant difference between intruder and legitimate nodes, such as their power consumption. Intruder nodes in DoS attacks tend to consume power more than a regular node due to their constant interactions with other nodes. The other way

is based on their contribution to the purpose of the network which is in the case of monitoring applications, responding to the query. In other words, intruders are nodes that consume a lot of energy that is not used for answering the query, rather in other unknown activates.

In DoS attacks, an intruder attacks other nodes within its range of communication and floods it with a request which as a result causes a huge power lost for that node and then an early death. Since all nodes are static, the intruder will not mobile to find another target if his target was set to sleep, because the intruder has no knowledge of the status of the other nodes, which will make the intruder lose power consistently whereas the power of the nodes are saved.

Our solution is based on the premise that intruders are nodes tend to be excessively busy doing nothing, they are not efficient nodes in the network and not contribute to the main function of the network, which is monitoring the network.

We introduce several metrics to detect intruders and compare the results. Using the simulation, we identify these intruders that act suspiciously in the network and then we take actions to minimize their harm as early as possible by setting a random time to shut them off.

Many of the traditional approaches to intrusion detection consist of a two-step approach: In the first step, a profile is created to characterize intruder behaviour. In the second phase, while the network is operating, the observed behaviour is compared with what has been catalogued and flagged if it matches catalogued abnormal behaviour [4][5][6] or if it deviates from catalogued normal behaviour [7][8].

Overall, the literature in sensor network intrusion detection can be divided according to what they protect. The resources typically targeted and protected include: data packets that can be maliciously dropped or changed [6], communication paths that can be intercepted and broken [5], communication signals that can be interfered with [10], normal behaviour that can be diverted by intrusion nodes [8], and data routing paths [11].

These approaches tend to be demanding in terms of storage and computation. The patterns that they catalogue tend to be generic and are not very effective in the very specialized, application specific context of sensor networks [4]. The issue of performance has been partially addressed by distributing the work among nodes and optimizing the codes required to identify intruders [12][17].

As a result many of the existing approaches have a high level of false positives and false negatives. On the other hand, using a wider net and looking for all unusual and rare behaviour may catch more intruders but would also result in a large number of false alarms.

The existing works on intrusion detection suffer from fundamental deficiencies:

They are confined to specific kind of attacks, like wormhole attacks, routing holes, or to particular operations, like routing, localization, etc. These methods either have been associated in the past with malicious intruders or are simply suspicious because they are rare or different.

These approaches also tend to be demanding in terms of storage and computation, and the patterns that they catalog tend to be generic and not application specific [5].

Moreover, we noticed intrusion detection approaches for DoS attacks in monitoring applications consume a lot of energy for data transmission and processing. In monitoring applications, energy should be managed wisely to extend the lifetime of the network.

So this brings up the need for an efficient intrusion detection that is low in false positive and false negative, and as efficient as expensive approaches in terms of catching the intruders as early as possible. We are looking also for less data transmitting and communications back and forth with the cluster head, and less data processing.

Our approach is based on the following premises:

Premise 1: In monitoring applications, single nodes hold no critical or private information. Malicious intruders will attack by depleting nodes of their power through purposeless activities.

Premise 2: Intruder nodes attack by engaging in intensive and purposeless activity.

The key then to identify intruder is to detect “intensive” and “purposeless” activity. In a nutshell, the intensity of activity of a node is measured in part through the node’s level of communication or power usage. The purposelessness of the activity of a node requires a bit more thought to define. Our challenge is then to identify activity and purposefulness metrics with the following characteristics:

- Accuracy: the metrics should allow us to identify all the harmful nodes (intruders).
- Fairness: the metrics should not label harmless (innocent) nodes as intruders.
- Minimal damage: the metrics should allow us to identify intruders fast enough, before the intruders cause too much damage to the network.
- Efficiency: the effort required (power used) to identify intruders should cause minimal overhead to the nodes.
- Distribution: the detection of the intruders should not rest on a small set of nodes.

We propose a set of metrics to characterize, respectively, intensity, purposefulness, and intensive purposefulness, and propose the architecture for monitoring these metrics.

In this research, we experiment information based intruder detection paradigm that does not rely on any extrinsic features of the net effect of that behaviour and on its compatibility

with the mission of the overall network. A node whose objectives are in conflict with those of the network is considered harmful to the network, irrespective of its intentions.

This paper is organized as follows: Section 2 describes the method we used. Section 3 talks about intrusion detection metrics. Section 4 explains network immunization. Section 5 has the simulation and results. Section 6 conclusion and future work.

II. Method

There is a general agreement in the research community that the efficient management of power is paramount to the success of sensor networks and to the realization of their full potential in practical application [14]. Therefore we used different strategies in our research to ensure efficient energy consumption methods to detect intruders, these strategies are combinations of:

A. Selective Querying

As explained in [15], the idea of this approach is to query the minimum number of nodes that is enough to tell us accurate answer about the query. The selective querying is based on the following premise:

Given a query Q computing some aggregate function f, given a set S of sensor nodes, it is possible to find a relatively small – subset S_0 of S such that $f(S_0)$ provides us with a good approximation of $f(S)$. The ideal of implementation of this premise is to select the S_0 that contains the right size of nodes and right contents of data. This subset should guarantee us two things, first, it should detect emerging event, and second, it should include nodes that have proven to be the most relevant in the recent past.

We define S_0 the subset of nodes from each cluster to be queried such that it includes the scrutiny set and the exploratory set -the scrutiny set makes 75% of S_0- contains nodes that show high relevance in the recent past, normally it includes nodes concentrated around the emerging event so that we ensure high accuracy of the query. On the other hand, the exploratory set - makes 25% of S_0 - is a randomly selected set of nodes that is to give an opportunity to other nodes to be queried where an event might take place. So it gives a wider picture of the whole network. The selection of nodes is done using the next strategy.

B. Information Value Based Trans-information

The idea of information value is to select a number of nodes to query in each iteration that shows high information value and terminating the other ones that show low information value. The information collected from one node is not useful by itself, instead the values of all the nodes aggregated together matters in monitoring applications.

We used the Usefulness Metric that will be explained later to identify the useful nodes. Then we sort them based on their usefulness and keep those of high usefulness values in our scrutiny set to query in the next iterations, and discard the nodes of low usefulness and replace them with randomly selected nodes to represent the exploratory set. In other words, the querying nodes S_0 consist of:

- 75% of the nodes with the highest Usefulness value, regardless of how many times a node was queried in the past, as long as it shows that it is an area where the phenomena take place then keep this node in the scrutiny set.
- 25% randomly selected nodes, get replaced every fixed number of iterations. Their purpose is to explore other areas in the network which might be overlooked, and not focus on the 75% set of nodes only.

By sampling nodes in this way, we ensure high accuracy to our query when we only query 12% of all the nodes. Using the Usefulness metric alone does not guarantee that we save power. We need to add other metrics to capture power consumption use in all nodes.

In monitoring applications, no single node hold a critical or private information, so the most damaging type of attack is DoS, when the goal of the attacker is to build a connection with the nodes and overload them with requests forcing them to deplete their power, hence die early.

III. Intrusion Detection Metrics

Based on premise two, we divide our metrics into three categories: measures of intensity, measures of purposefulness and finally metrics that combine the two together so that it measures intensive purposefulness.

A. Measures of Intensity

We came up with an approach to identify suspicious nodes based on the intensive activity knowledge of the nodes. Here we discuss the different ways in which we measure the intensity of activity. We can include:

1) Absolute Power Usage Outlier Based Detection

Intruders are more active than any other nodes in the network; we assume that Intruders lose power significantly as due to their intensive interaction with their surrounding nodes. Whereas legitimate nodes that are being queried lose the same amount of power in one time interval to respond to that query.

So we present the *Activity* $A(N_i, t, \delta)$ Metric that measures the difference of power for δ time period that measures the difference of power during time interval of interest, defined as:

$$A(N_i, t, \delta) = R(N_i, t+\delta) - R(N_i, t) \quad (1)$$

The *Activity* of a node N_i at time interval starting from t and having duration δ is denoted as $A(N_i, t, \delta)$, and where $R(N_i, t)$ is the residual power of node N_i at time t. So in one time

interval δ, the maximum power consumption for a queried node is *(δ *Query Cost)* for each node.

We conclude that this measurement is used to identify intruders in our network setup, since they tend to be highly active during one time interval comparing to other nodes that lose power no more than *(δ *Query Cost)* if they are interrogated.

The activity Metric cannot be used to identify intruders in other hierarchy like trees hierarchy, when the activity value decreases for the nodes as you go up the tree, assuming that a node can interrogate its children not its parents, yet it sends the result to its parents. So the activity decreases for nodes in a higher level of the tree. But for the sake of having a complete methodology that is compatible with other network setup we define another metric that is suitable for all assumptions.

2) Relative Power Usage

The metric of relative power usage measures the ratio of the power level of a node over the average power of its neighbors. The neighbors are defined as the nodes that are within the same communication range of a particular node. So a node can have neighbors that belong to other clusters as long as they are close to each other.

$$\textit{Relative Power Usage} = \textit{Power}N_i / \textit{Avg Power Neighbors} \quad (2)$$

where *Power Ni* is the amount of power consumed by Node *Ni*. As we see in Figure 3 that intruders have significant Relative Power Usage over the other nodes, the reason for this is that we assume that intruder never runs out of power, whereas any other node continue to lose power as a result to answering the query. The average power of neighbors decreases which results in increasing the relative power usage of intruders.

We can use this method to detect intruders where we can access the power level of each node, where intruders have infinite power supply.

3) Total Number of Messages Received

This metric relies on the number of messages received by each node, we assume that the intruder send one "Hello message" to its surrounding nodes and wait for response from them. In each iteration, an intruder sends one message to all and receives a number of messages equal to the number of its neighbors. As a result the total number of messages received by the intruder is higher than any other node. Because the nodes do not communicate among each other, they only communicate with the cluster head. On the other hand the cluster head sends one query message to all nodes and receives response messages from all of its members.

In Figure 4, we see that the cluster heads have the highest total number of messages received, since each head receives an approximation of *N/C* members, where *N* is the total number of nodes, and *C* is the number of clusters in the network, $C = 4$ in our setup.

The intruders come next since they communicate only with their neighbors. The neighbors are less than *N*/4, which explains why they lie in the middle. Regular nodes are two types, either suspicious or infected by the intruder. These nodes receive two messages in each iteration. One message is from the cluster head for the query and the other one is from the intruder.

We conclude that we can use the total number of messages received to detect intruders if we know the normal number of messages it receives from its cluster head per time unit t.

4) Difference between Messages Received and Sent

This metric measures the difference between numbers of messages that go out and that come in in each node. An intruder will have a big number of messages received comparing to a single "message-out" in a single iteration. As for a legitimate node this difference should remain zero, either infected or not, whatever messages they receive, they respond to them. Cluster heads should have a bigger difference than intruders. That is due to the number of nodes in one cluster is bigger than the total nodes that an intruder could approach within its range.

We keep track of this metric to allocate intruders, and plot each node based on its location. It is quick in detecting intruders but it falsely detects cluster heads as intruders for the same reason explained above. To avoid this we set a threshold for cluster-head values to be identified and then all the intruders will be easily pointed out based on this metric with 100% accuracy.

B. Measures of Purposefulness

We use metrics that measure how much each node is contributing to the purpose of the network. For example nodes with *MAX* values are considered to be purposeful because the query of interest is to find the maximum reading, whereas nodes with low values are not.

Intruders are not only highly active, but also they don't serve the main mission of the network, which is monitoring the phenomenon. Any node that contributes to the query will be considered a useful node, but if it does not add any value to the network, then a flag will be raised for being an intruder. We present *Usefulness* metric to measure the purposefulness of each sensor node.

Usefulness was explained in Section 2 as a method to selective querying. It is used as a metric to measure the *Usefulness* of a node. One reading from an individual node is not valuable, rather we are interested in knowing its contribution to answer the query. This notion is captured in the concept of Mutual Information or Trans-information denoted by *I(X, Y)* for two random variables *X* and *Y*. Trans-information measures the quantity of information that can be

obtained about *X* by observing *Y*. In our case, we use this concept to define *Usefulness*, a time-variable correlation between a query *Q* and a sensor node N_i, up to the current time *t*. We are interested in the amount of information that can be learned about the query result from interrogating node *Ni*. The trans-information of *Q* and N_i is denoted as:

$$U(Q, Ni, t) = -\sum_{[t-\delta t, t]} p(q, m_i) \log\left(\frac{p(q, m_i)}{p(q)\, p(m_i)}\right) \qquad (3)$$

where m_i is the message associated with node N_i and *p(q)* is the probability of query *q*, and *p(q,mi)* is *the* joint probability of *q* and *mi*, given the discrete probability distribution calculated over period Δt. We focus on recent history over a selected time interval of length δt. The relevance of a node to the query will vary over time, because natural phenomena are continuous over time and space. We assume that the relevance of a node at present time is highly correlated with its relevance over the recent history. This is explained in details in [15][16].

An example is shown in Table I, a snap shot of sensor data for only 10 nodes from iterations 11-51.The query is:

SELECT MAX
FROM ALL Nodes
EVERY 5 Periods

The event of interest is the maximum value, which means to select the maximum temperature if nodes are used to measure temperature. As seen in Table I, Max value in the 16th iteration is node 5 then it moves to node 9 in the 31th iteration, and so on. The relevance of the nodes is accurately reflected by their *Usefulness* values. Nodes that have values close to the max value, have higher Usefulness than others. *Usefulness* for the values in the selected iterations is shown in Table II. To calculate the *Usefulness*, we need a history of each node, *Usefulness* is being built over a history of 5 iterations back.

TABLE I. A SNAP SHOT OF NODES DATA

	T_{11}	T_{16}	T_{21}	T_{26}	T_{31}	T_{36}	T_{41}	T_{46}	T_{51}
N_1	0	0	0	80	0	0	0	0	0
N_2	0	0	8	44	0	0	0	0	0
N_3	0	4	22	22	0	2	0	56	1
N_4	0	5	0	0	0	1	0	54	0
N_5	0	29	38	51	0	6	44	32	29
N_6	0	0	0	0	0	0	58	0	0
N_7	0	0	0	0	0	0	0	0	0
N_8	0	0	0	0	0	0	0	0	0
N_9	0	0	0	0	36	0	0	0	0
N_{10}	0	0	2	0	0	0	0	0	0
Qry	0	29	38	80	36	6	58	56	29

TABLE II. USEFULNESS

	T_{11}	T_{16}	T_{21}	T_{26}	T_{36}	T_{41}	T_{46}	T_{51}
N_1	2.26	0	0	0.21	0.57	0	0	0.69
N_2	2.26	0	0.21	0.69	0.57	0	0	0
N_3	2.26	0.217	0.57	0.69	0.57	0.21	0	0.21
N_4	2.26	0.69	0.69	0	1.33	0.21	0	0.21
N_5	2.26	0.217	0.69	0.69	0.45	0.21	0.41	0.69
N_6	2.26	0	0	0	1.33	0	0.21	0.69
N_7	2.26	0	0	0	1.33	0	0	0
N_8	2.26	0	0	0	1.33	0	0.69	0
N_9	2.26	0	0	0	1.48	0.69	0	0
N_{10}	2.26	0	0.21	0.69	1.33	0	0	0

C. Measures of Intensive Purposefulness

These set of metrics capture the combination of intensity measures and purposefulness measures together. That means nodes should be characterized by their effect on the network rather than by some classification of their intentions. Therefore, an innocent node is positively contributing to the operation of the network, even if it is an intruder. An intruder node is the node whose operation is a burden on the network, even if the node is legitimate.

We have a set of metrics that capture intensive and purposefulness at the same time.

1) Utility

We use the concept of *Utility* to capture the combination of *Usefulness* and cost. The *Utility* of a node N_i, at the present time *t* for a query *Q*, is denoted by *f(Q, N_i, t)* and defined as a function that is directly proportional to *Usefulness* and inversely proportional to the cost of using N_i.

Utility (Ni, Q, t) = Usefulness (Ni, Q, t)/Cost (Ni, Q) (4)

where Cost *(Ni, Q)* is the cost of querying node N_i to compute query *Q*. Utility shows that after 50 iterations, intruders are clearly identified and their *Utility* values are lower than 1 (the *Usefulness* default value for all nodes is 1, and the cost value is 1 unit).

2) Usability

Sensor nodes die out when they exhaust their power. We give preference to nodes with a high power reserve and exclude nodes who have exhausted their power. The Usability of a node N_i , at the present time *t* for a query *Q*, is defined by

*Usability (Ni, Q, t) = Utility (Ni, Q, t) * Power (N_i)* (5)

A node N_i will have the highest *Usability* if it has the highest *Utility* and the highest residual power. A node with no residual power will have a *Usability* of zero, regardless of its *Utility*. The higher the *Usability* of a node, the higher is its probability of being legitimate node.

Simulation results have established the effectiveness of this approach. The life of the network is always multiplied by the inverse of the ratio S_0/S.

We can consider Usability as an efficient intrusion detection metric in this setting, but it has some limitations:

- Some intruders might have finite power supply. After it lunches its attack it dies soon after that. It is like an intruder commits a suicide and happily ends its attacks with some casualties. It is hard to catch this kind of intruder using our approach assumption, we assume that the intruder has infinite power which is the case in most DoS attacks.
- Smart intruders can deceive the network about their residual power level, so the intruder will stay hidden in this case.

3) Contribution

The *Contribution*, of a node N_i to the monitoring task, is measured by the power used in communication modulated by the *Usefulness* of the information communicated.

The Contribution of a node N_i to query Q at time interval starting from t and having duration δ is denoted $T(N_i, Q, t, \delta)$ and defined as the weighed sum of *Usefulness* $U(N_i, Q, t_k) * C(N_i, Q, t_k)$ where $C(N_i, Q, t_k)$ is the amount of power used by node N_i in participation to computing query Q at time t_k in the interval $[t, t + \delta]$. If node N_i was not interrogated at time t_k, then its contribution is zero. Contribution $T(N_i, Q, t, \delta)$ is defined by:

$$T(Ni,Q, t, \delta) = \sum_{t_k=t}^{t_k=t+\delta} U(N_i, Q, tk) * (Ni, Q, tk) \qquad (6)$$

A node with high *Contribution* value means it is often interrogated and it is useful when interrogated. A node with low *Contribution* value is never interrogated, or it is often interrogated while it has a low *Usefulness* or low *Contribution* value or both. Nodes that happen to have a low *Usability* are bound to have a low *Contribution*. Also nodes that are never interrogated are also bound to have a low *Contribution*. Nodes that have a low productivity but use no power are harmless nodes. Intruder nodes should have low *Contribution*, but they consume high amount of power. So the *Contribution* by itself is not an accurate measurement for intruder detection. Therefore we introduce the last concept, *Convergence* in which we use *Contribution* as an input.

4) Convergence

We define the concept of *Convergence* between a node's operation and the network's main function, via accurate computation of the query. We quantify the level of *Convergence* of a node *Ni* with the goals of the network by the extent to which the activity (measured by power) within that node was used to contribute to the accuracy of computation query *Q*.

We need another parameter along with Activity, to distinguish between nodes whose activity is supporting the main function of the network and nodes whose activity is not. That captures the combination between the *Contributions* of a node N_i, to the query Q during time interval that starts at t and lasts for δ units is denoted by $G(N_i, Q, t, \delta)$ and defined as:

$$G(N_i, Q, t, \delta) = T(N_i, Q, t, \delta) / A(N_i, t, \delta) \qquad (7)$$

The Convergence measures the ratio between the *Contribution* the node made to the query and the total power consumed. The higher the *Usefulness* of a node, the higher will be its *Convergence*, unless it consumed power in other tasks than computing the query.

The essence of this approach is that any node whose activity by far exceeds its contribution to the main function of the network can be counterproductive and safely considered as intrusive, whether it is or not.

This does not distinguish between the cases where the source of this activity is an external node that infiltrated the network or an internal node that was attacked by an intruder, and does not distinguish between accidental and malicious intrusions. All have the same potential effects, they need to be identified and managed so as to limit control and cease the damage that they are causing the network.

In other words, with this approach we no longer have a problem of false negatives or positives. Furthermore, our approach is unique in the sense that it takes its "order" from the activity on the ground rather than from some arbitrary attributes that have no necessary bearing on the function of the network.

IV. Network Immunization and Nodes Shut Down

This is our proposed approach for network immunization in order to save the power of the attacked nodes and hence, extend the life of the overall intruders. We consider a scenario where some of our regular nodes were hijacked and programmed to act like intruders.

The cluster head has a defined threshold for each metric. Once intruders are detected, the next question is what to do with them? Intruder nodes are not under the network's control anymore, and most likely will not be programmed to suicide just because they are told that their performance is low. We cannot force them to self-destruct, but we can protect the rest of the network by cancelling out or minimizing their impact.

Intruder has a relatively low Convergence value due to activity that is not related to answering the network query, and most likely consisting of sending messages to neighbouring nodes in an effort to deplete their power. Immediate neighbours are shown in yellow because their Convergence values are also declining.

There are two possibilities: first, after some time, the effect will propagate whereby all of the immediate neighbours are

flagged as intruders and they start "infecting" their immediate neighbours. Second, die early without affecting the neighbouring nodes, this assumption is less harmful to the overall network. In both cases our approach will detect intruders early as well as infected nodes and save the network by requesting them to sleep for a certain predefined time then wake up and resume their functions for another certain predefined intervals.

We cannot control the outsider that has been flagged as intruder; we can prevent and stop the propagation of its effect on the network. We were inspired by a biology strategy to apply in the sensor network; apoptosis, which is about self-initiated, shut down of a cell that recognizes that its current mode of operation may be harmful to the network.

The cluster head is the decision maker, so that it gives order to the attacked sensor nodes so that they shut down for a random time with a predetermined distribution whenever they recognize that they may be the subject of an attack.

Unlike the biological systems, the shutting down is not permanent for two reasons: The low Convergence metric of a node may be the result, not due to an intrusion, but due to lack of activity in that area, which is shown in Figure 8 during iterations nodes with low Convergence, because they were not centred around the event, but the history is being built up with time. The second reason, if we were to shut the nodes permanently we run the risk of "killing" the network prematurely, thus fulfilling the intruder's mission, to destroy and ruin the whole purpose of the network.

V. Performance Metrics

To evaluate our approach, it is important to build the following confusion metrics which represent the comparison between the detection and actual diagnoses of a node:

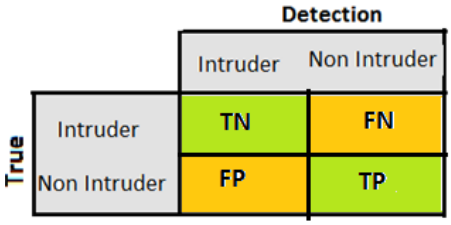

Fig. 1. Confusion Matrix

where FN is false negative, FP is false positive, TN is true negative, and TP is true positive.

An efficient intrusion detection requires high degree of accuracy and detection rate, and low false alarm rate. The performance is assessed in terms of accuracy, detection rate and false alarm rate as in the following performance metrics:

1) *Detection Rate = (TN/ Total Attacks)*100*
2) *False Negative: Percentage of Undetected Attacks = (FN / Total Attacks) *100*
3) *False Positive: Percentage of False Detection= FP/ Total Attacks) *100*
4) *Number of Errors = FP + FN*
5) *Accuracy = (TP + TN)/ (TP+ TN + FP + TN)*

VI. Simulation and Results

We used MATLAB simulation to study how accurate and effective our approach is in detecting intruders. We simulated the behaviour of one cluster. We used 200 nodes placed on a 100 x 100 grid. All input values are built by using

$$f(x,y) = h * e^{(-(x-a)^2 - (y-b)^2)/w} \qquad (8)$$

where h is the range of the phenomenon, or the height of the peaks in the data, w the radius of the phenomenon or the width. In this equation, the event is centered at (a, b), with a peak in value at that point and exponentially decline as we get further from the center (a, b). The smaller is w, the narrower the peak is, and the steeper the decline is. With a large w, the data changes more slowly. $(a,b$,the location of max or the center of the peak, moves with time along with h and w. In other words, a and b are in fact functions of time t.

We have used linear movement pattern, as shown in Figure 10. The interest is to find the MAX reading and the max value moving linearly in each iteration.

All the nodes are initialized with full battery of 128 power units. Each query costs one unit. The query we simulated is finding the maximum values among S_0 or the subset of nodes that include the scrutiny and exploratory nodes.

The simulation starts by initializing the sensor nodes values as discussed above, and randomly scatter them on the grid. At the first interaction, we pick our subset nodes S_0 randomly, we query the same nodes for 5 iterations then calculate its *Usefulness*, sort the nodes based on their *Usefulness* values, keep the highest 20 nodes in the scrutiny set, and pick 5 nodes randomly to replace the exploratory set and query this new S_0 for the next 5 iterations.

The *Activity* metric is calculated at the end of time interval which is 5, to compare the power lost during that time. The battery of a node loses one unit in each communication action with others.

During each of the iterations, we query the maximum values, then calculate the *Usefulness* for the nodes, then all the other metrics Utility, Usability, Contribution and Convergence are also calculated to keep track of these values for each node.

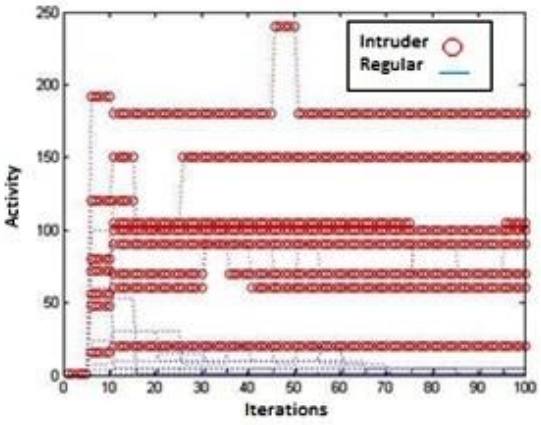

Fig. 2. Activity

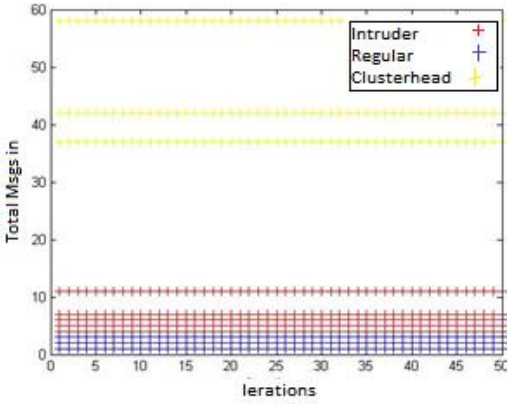

Fig. 3. Total Messages Received

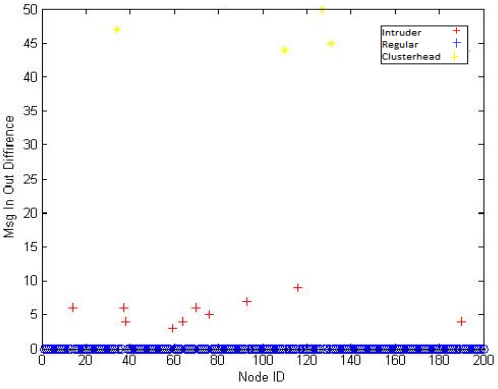

Fig 4. Message in – out difference

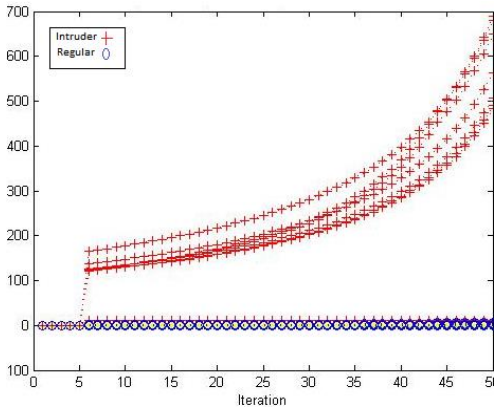

Fig. 5. Relative Power Usage

Figures 2–5 represent the intensity metrics. These metrics measure how active the network is. Intruders were easily flagged with 100% accuracy. Zero false negative, and false positive and no errors, except for total message received, our methods show 4 errors as it flagged cluster heads as intruders. This kind of error can be avoided by setting two thresholds or a range with upper bound and lower bond, for example [12-30] values within this range are intruders, less than that are legitimate nodes, more than that range are legitimate cluster heads.

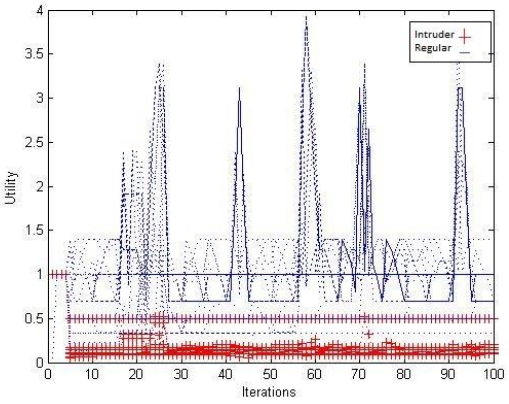

Fig. 6. Utility

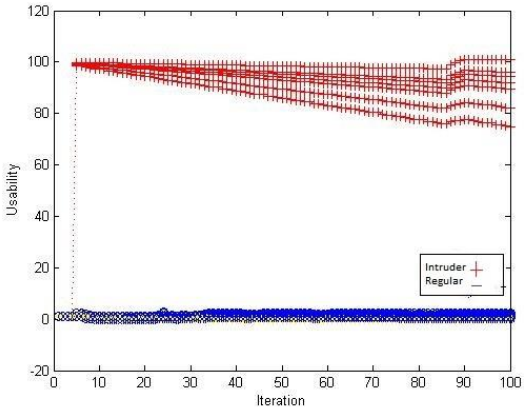

Fig 7. Usability

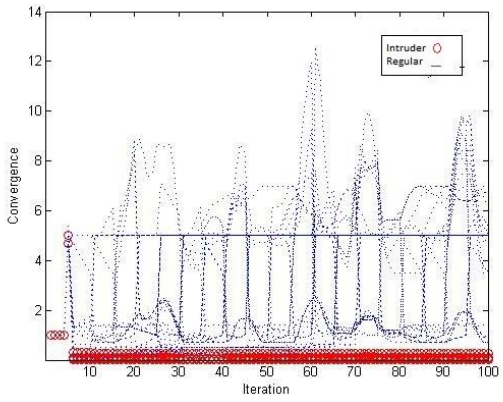

Fig.8. Convergence

Figures 6 - 8 represent the intensive purposefulness metrics. Figures 6 - 8 shows that it takes longer iterations to detect intruders than intensity metrics, which is expected because it takes into account two measures, how active a node is besides how useful that node is in the network. This approach shows all the legitimate but not useful nodes with low values in all intensive purposefulness metrics, values that are almost close to the intruders' values. This could be because, either these innocent nodes are not being active in participating in the query, or their *Usefulness* is low. The reason why we have straight lines of *Utility* in Figure 6 with value equal to one is that we query only 12 % of the nodes, so a lot of them are not being interrogated at all. The straight line represents those nodes. We use a threshold to separate intruders from others since it might catch some of these nodes that are not being interrogated, which explains the false positive values in Table III.

Our goal is to query the nodes of high usefulness. If the threshold catches those non-intruder nodes and flags them as suspicious, it will not affect the functionality of the network. Moreover, it does not matter if we lose few innocent nodes and avoid communicating with them, they are not useful anyway. They will naturally be avoided by our querying algorithm. For example, a node that give low readings every time we send a SELECT MAX query, at the same time it shows some high activities, that means it might have been infected by a malicious node and got engaged in unknown activities with it, such node will be flagged as intruder, which explains the error rate and false positive values in Table III.

Finally, *Usability* in Figure 8 shows clearly that it detects intruders in early iterations with 100% accuracy, as early as intensity metrics. At the same time, it considers the purposefulness of the nodes. Therefore, *Usability* metric in our simulation setup is the most effective metric that combines the best of the three measures.

Figure 9 shows the deployment of nodes. The nodes are randomly scattered on the field of interest, every dot represents one node. Any node within a node's communication distance is considered a neighbour. The circle shows an intruder and its direct neighbours inside the circle that got infected.

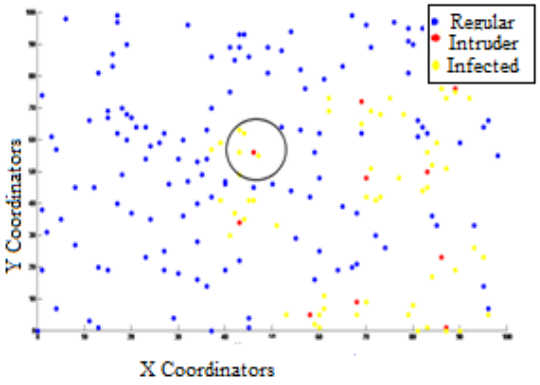

Fig. 9. All the nodes

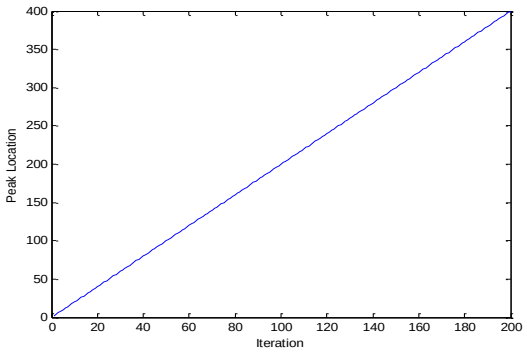

Fig. 10. Linear Event Movement

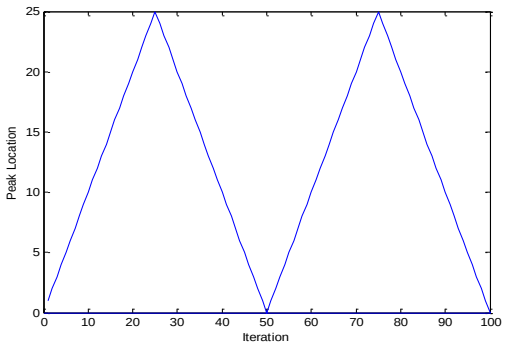

Fig. 11. Zigzag Event Movement

Figures 10 and Figure 11 show two kinds of sensor input data. We tested them both to see if we still get accurate results with different input data. In Figure 10 the event, which is MAX, moves linearly in the network. Figure 11 it moves like a zigzag in the network. No significant difference is observed in our results between the two kinds of data.

TABLE III. PERFORMANCE METRICS

	Accuracy	Detection Rate	FN	FP	Err
Activity	1	100%	0%	0%	0
Msg. Rec.	0.98	100%	0%	40%	4
Msg in-out	098	100%	0%	0%	0
Rel. Power	1	100%	0%	0%	0
Utility	0.90	90%	10%	100%	11
Usability	1	100%	0%	0%	0
Convergence	0.93	80%	20%	40%	6

Table III shows the performance metric results for each one of the intrusion detection approaches. As a result *Activity, Relative Power Usage* and *Usability* are the most effective and accurate with no error.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed efficient intrusion detection metrics that are based on three criteria: intensity, purposefulness and intensity purposefulness.

To detect intruders, we took two things in mind. First: intensity, or how active the nodes are in terms of communicating with each other, which can be measured using power consumption. Legitimate nodes have reasonable intensity of activities to do their job to monitor the network and to respond to queries, whereas intruders have more intensive activities, thus, more power consumption.

The other criteria is purposefulness of these nodes, in other words, legitimate nodes should have high purposefulness values whereas intruder's purposefulness values are always low. We defined and used a set of metrics to measure that.

The third criteria is the combinations of intensity measures and purposefulness together, which means nodes that are active in performing a meaningful service to the network, rather than active in the network doing nothing useful. We marked last kind of nodes as intruder. We injected intruders into the network that does not participate in anything to the query but perform one type of denial of service attack. In this kind of attack, intruders behave in a way to make other nodes busy by engaging them in useless communication and forcing them to respond to those messages. Intruders are highly active but not useful. We tested if we can detect intruders based on the information that is already in the cluster-head which sends out a query to all nodes to get their readings. We use this information and feed them to our matrices that result in classifying nodes as: intruder, not intruders, or suspicious. Unlike other intrusion detection approaches that require a lot of communication and calculations about the current status of each node, our approach is efficient, cheap and accurate to detect intruders.

We introduced different metrics and explained in what setup each one of these metrics work the best. A threshold should be set to distinguish between intruders and other nodes. Our simulation shows that *Activity, Relative Power Usage* and *Usability* metrics are most effective and accurate with no errors. In this paper, we have also suggested a strategy that is designed to immunize the network against the harmful effects of the intruders. This was designed to stop the propagation of the intrusion by disabling the set of nodes that were detected as infected nodes or the neighbours to the intruder for a randomly generated length of time.

For future work, we want to be able to detect other types of intruders such as mobile intruders that continue to move in the network and cause damage, so our immunization strategy will not be valuable in this case, we need to dynamic immunization strategy and keep track of the intruder path and take action as the intruder moves.

VIII. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grants CNS-1338105 and CNS-1343141. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] I. Akyildiz and V. Mehmet, "Wireless sensor networks," Vol. 4. John Wiley & Sons, 2010.

[2] N. Trigone, Y. Yao, J. Gehrke, and R. Rajaraman, "Multiquery optimization for sensor networks," In International Conference on Distributed Processing on Sensor Systems (DCOSS), 2005.

[3] C. Edith and H. Ngai, "Intrusion Detection for Wireless Sensor Networks," Computer Networks, 55:3224-3245, 2011.

[4] I. Demirkol, F. Alagoz, H. Delic, and C. Ersoy, "Wireless sensor networks for intrusion detection: packet traffic modeling," 10(1):22–24, Jan. 2006.

[5] J. Deng, R. Han, and S. Mishra, " Defending against path-based dos attacks in wireless sensor networks," In SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, pages 89–96, New York, NY, USA, 2005.ACM Press.

[6] I. Demirkol, F. Alagoz, H. Delic, and C. Ersoy, "Wireless sensor networks for intrusion detection: packet traffic modeling," vol. 10, no. 1,pp. 22–24, Jan. 2006.

[7] P. Dutta, M. Grimmer, A. Arora, S.Bibyk, and D. Culler, " Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," In IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks, Piscataway, NJ, USA, 2005. IEEE Press.

[8] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," J. High Speed Netw., 15(1):33–51, 2006.

[9] J. Deng, R. Han, and S. Mishra, " Defending against path-based dos attacks in wireless sensor networks," In SASN'05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, pages 89–96, New York, NY, USA, 2005. ACM Press.

[10] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher, "Rid: Radio interference detection in wireless sensor networks," in INFOCOM '05:24th Annual Joint Conference of IEEE Computer and Communications Societies, 2005.

[11] J. Deng, H. Richard, and S. Mishra, "Insens:intrusion-tolerant routing in wireless sensor networks: Dependable wireless sensor networks," in Proc. 23rd IEEE International Conference on Distributed Computing Systems, 2003.

[12] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad-hoc networks," 2004 IEEE Wireless Communications, 11(1): 48-60, Feb. 2004.

[13] S. Patil, and G. de Veciana, "Managing resources and quality of service in heterogeneous wireless systems exploiting opportunism," IEEE/ACM Transactions on Networking (TON) 15.5 (2007): 1046-1058.

[14] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy aware wireless microsensor networks," IEEE Signal Processing Magazine, 19(2):40–50, March 2002.

[15] C. Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, pp. 379–423,623–656, July, October 1948

[16] A. Yousuf, N.Alrajei and F.Mili, "Information Theory Based Intruder Detection in Sensor Networks," 3rd Indian International Conference on Artificial Intelligence, Dec 2007.

[17] W. Michener, G. Bonito, and T. W. Participants, "Areport from a national science foundation sponsored workshop: Environmental cyberinfrastructure needs for distributed networks," Technical report, Scripps Institute of Oceanography, Aug 2003.

[18] C. Intanagonwiwa, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," In 6th annual international conference on Mobile Computing and Networking, 2000.

[19] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: an acquisitional queue processing system for sensor networks," ACM Trans. Database Syst., 30(1):122–173, 2005.

[20] D. Robert and B. Gastel, "How to write and publish a scientific paper," Cambridge University Press, 2012.

[21] Y. Kotidis, "Snapshot queries: Towards data-centric sensor networks," In 21st International Conference on Data Engineering, 2005.

[22] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," In HICSS'03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, Washington, DC, USA, 2003. IEEE Computer Society.

[23] J. Meyer, F. Mili, I. Elhajj, M. Rossman, and I. Bass, "Distributed query optimization in wireless sensor networks," in Wireless and Optical Comm. Conf. (WOCC'06), Hangzhou, China, Oct. 2006.