Clustering Utilization for Coverage Improvement in Wireless Border Surveillance Networks

ZHILBERT TAFA
Dep. of Computer Sciences and Engineering
University for Business and Technology
L. Kalabria, Prishtina
KOSOVO
tafaul@t-com.me

Abstract: - In many border surveillance applications (such as military, homeland security, etc.), the wireless sensor networks cannot be deployed manually and the barrier coverage breaks can appear along a given surveillance line. This paper introduces a cluster-based algorithm and new metrics to determining the number and the positions of the additional nodes needed to be deployed by drones, robots, or moved in a network; in order to fill the gaps in a randomly deployed network. Simulation results show that the proposed algorithm optimizes the number of additional nodes and outperforms the alternative in 52,15% of cases while it performs similarly in the rest of the cases. The machine learning classification algorithms are used to show that the decision on choosing one or another algorithm is highly classifiable. Precisely, the proposed algorithm is shown to be the approach of choice in implementations where the sensing range is relatively small.

Key-Words: algorithm, barrier coverage, clustering; machine learning, decision trees, neural networks, support vector machines, wireless sensor networks.

1 Introduction

The WSNs have been widely employed in many long-term surveillance applications such as field surveillance, critical infrastructure protection and country border control [1]. Two fundamental WSN-based application requirements are the ability of the network to sense each event in the region of interest (ROI), and its ability to communicate the sensed events (from each point of interest) to the sink node. The ROI can be a specific point - when the sensing coverage is referred as a point or target coverage, a specific region – when it is referred as the area coverage, a specific path – which is called the path coverage or the border line between two areas – named barrier coverage.

In recent years, constructing sensor barrier is a very critical issue in wireless sensor networks for military and homeland security applications [2]. An example of WSN-based multi-layer border patrol architecture for national security is given in [3].

Barrier coverage describes the ability of the network to detect an object crossing from one side of the barrier (line) to another. It is distinguished from the traditional coverage models in a sense that it aims to protect an area of interest or points of interest by simply having a chain of sensor nodes surrounding them rather than continually monitoring the entire area or all the points [4]. In applications when the sensor nodes are placed at the expected locations and (ideally) are not supposed to fail, the barrier coverage can be easily guaranteed. However, in reality, the nodes are very prone to failures so the barrier gaps can appear, for example, due to the environmental changes or due to the hardware failure. Additionally, in actual applications, most cases of the ROI are in harsh environment, which make it difficult to obtain the expected locations and deploy the nodes there [5]. In these situations, the deployment will not guarantee the required sensing coverage and connectivity. Therefore, exploring the possibilities to assessing and healing the barrier holes in a quasi randomly deployed network is of a fundamental importance.

In [6] the authors have derived the critical conditions for the existence of the barrier coverage. But, these conditions are very strict and therefore economically unfeasible, since they require a large number of sensor nodes. Hence, instead of using only static nodes, recently, most of the approaches in topology control rely on the nodes' mobility, either by deploying fully mobile networks or by deploying hybrid networks that consist of static and mobile nodes. Although it can dramatically prolong the network lifetime, the deployment of fully mobile network largely increases the cost of sensor nodes and the complexity of the protocol design [7]. Hybrid networks, on the other hand, combine the advantages of both static and mobile networks. In the context of barrier coverage, mobile nodes are used to bridge the barrier breaks, if any, after the initial deployment with the aim of minimizing the number of mobile nodes.

This paper describes a new algorithm for the efficient barrier coverage gap bridging in hybrid networks where the final goal is creating the strong barrier coverage for border surveillance with as lower number of mobile nodes as possible. As compared to the previously used greedy algorithm, under the same conditions, the proposed algorithm showed better (52 %) or equal (48%) performances, but never underperformed it.

The reminder of the paper is structured as follows. The proceeding section reviews previous work on finding, mending, and predicting the barrier holes in quasi randomly deployed WSNs for barrier coverage. Section III presents the modelling details and methods, i.e., the network and sensing models, mathematical definitions and presumptions, the algorithms' description and functionality, and the evaluation methodology. The results are given in section IV. Finally, section V concludes the paper.

2 Related work

The problem of barrier coverage has been recently studied from different points of view. Most of the studies take advantage of mobile nodes in assisting the formation of the strong barrier coverage.

A study on the achievement of the barrier coverage in reconfigurable hybrid network is presented in [1]. The authors propose a greedy algorithm to provide the efficient barrier coverage under different weather conditions. A study on improving the barrier coverage by using sensors with limited mobility is presented in [8] while in [9] the authors show the barrier coverage construction based on sensor's density. Despite the predictable outcome, the authors show that the barrier coverage can be significantly improved with higher number of mobile nodes. Similar conclusions are derived in [2]. In this paper, the authors propose a network model and implement an algorithm that gives the number of mobile nodes to mend the barrier gaps in a given implementation. Although more focused on energy conservation, the authors in [10] use the same algorithm for mending barrier gaps via mobile sensor nodes with adjustable sensing ranges. A greedy algorithm for barrier constructing is presented in [11], while its variant and the idea of cluster based approach is given in [12].

In contrast to the mentioned works, our algorithm takes advantages of the fact that, in a randomly deployed (dense) network, clusters are likely to be created. The clusters compose connected graphs that are used by the algorithm as a fraction of barrier. We compare the outputs of the new algorithm with the outputs of the variant of greedy algorithm that is widely used in mentioned literature. In order to show the differences in performances between two algorithms, and to accentuate the most important parameters that influence these differences, to the best of our knowledge, we are the first to introduce the machine learning classification algorithms with the decision tree as a primary choice.

3 Models and methods

3.1 Network model

The experimental framework and setups model the network topology for barrier coverage in military inaccessible zones, where sensors cannot be deployed manually. Instead, the nodes are deployed quasi-randomly (from the aircrafts or artillery) on the region along a given line. Hence, the ROI is considered to be of a 2-D rectangular shape with the length much larger than the width, i.e., $l \gg w$.

As in similarly focused researches [13], [14], [15], [16], we also adopt the widely-used uniform distribution of the nodes' coordinates along the length and the width of the ROI.

3.2 The sensing model

Although, many other models have been proposed, the Boolean (or binary) sensing model ([17], [18]), is the mostly used one in modeling the sensing pattern. In real implementations, however, most of the nodes will not achieve the maximum (nominal) sensing range, because of the obstacles and other elements that influence the pattern of the sensing range. Furthermore, in practice, the sensing areas are never perfect disks. However, as stated in [19]; the disk model can provide lower and upper bounds for realistic irregular sensing areas. Therefore, we also adopt the Boolean sensing model but, as in Elfe's approach [20], and in contrast to most of the related work, we introduce some additional unpredictability that aims to model the influence of the environment on the sensing range of the nodes.

The sensing range for a given node k was randomized and defined as the circular area around the node with the center at the node's position and the radius between $R_m/2$ and R_m. The sensing radius of the node k is given with:

$$R_{sk} = \frac{R_m}{2} + r_k \qquad (1)$$

Here, R_m is the maximum nominal sensing range and r_k is a uniformly distributed random variable which takes values between 0 and $R_m/2$. A sensor k can detect an intruder if the intruder is within the distance R_{sk} from the sensor.

Nodes u and v are considered to be connected if the Euclidian distance between them is smaller or equal than the sum of their sensing ranges, i.e.,

$$d(u, v) \leq R_{su} + R_{sv} \qquad (2)$$

In Fig. 1, a simple two-node graph construction is depicted.

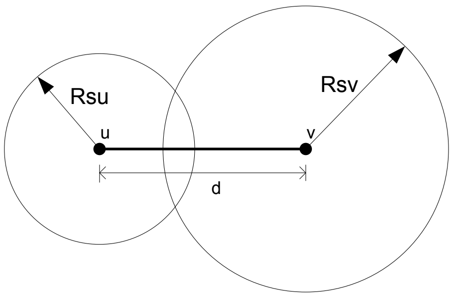

Fig. 1. Nodes *u* and *v* are considered to be connected.

In the given implementations, it is assumed that the communication radius is much greater than sensing range which is a realistic assumption in most cases bearing in mind that the sensing ranges are usually in order of tens of meters while the communication ranges are in order of hundreds of meters.

.

3.3 The algorithm description

After the initial deployment of n static nodes, if the barrier coverage is not achieved, the mobile nodes should be guided to fill the barrier gaps. As usually presumed in literature, in hybrid networks, mobile nodes can be deployed at the same time with stationary nodes, or they can be deployed afterwards by using robots or drones. Eventually, the network should provide at least one barrier that does not allow an intruder to cross from one site to another without being detected.

The coordinates of each stationary node are assumed to be known by combining the absolute positions from the on-board Global Positioning System (GPS) units and localization algorithms such as trilateration, triangulation, etc.

The aim of the algorithm is to find the positions in a randomly deployed network where the additional nodes can be added in order for the strong barrier coverage to be achieved. The algorithm is optimal if it results in minimal number of additional nodes.

In order to evaluate the efficiency of the proposed algorithm, we compare it with the existing greedy algorithm.

Let's suppose we have the topology of a randomly deployed network such as the one given in Fig 2.

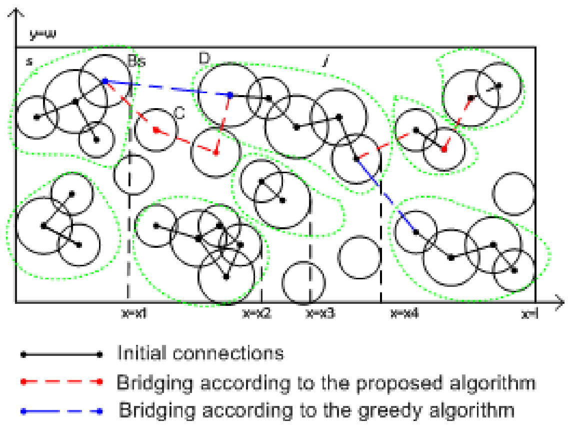

Fig. 2. A randomly deployed network. The lengthwise "paths" (barriers) as selected by two algorithms.

Both algorithms begin at the starting edge *x=0*. The rightmost node of the cluster that intersects the edge line *x=0*, and that reaches the farthest point towards the destination, is chosen to find its next hop towards the line x=l. In the given scenario, the starting cluster would be cluster *s*, with its rightmost node B_s.

The present, widely used greedy algorithms, would chose node C for $B_s's$ next hop by applying the following criteria: from all the nodes in communication range of B_s, among those whose x coordinates are greater than x_1 (Fig. 2), chose the closest one to B_s. Then fill the gap between B_s and the chosen node. The algorithm continues until a strong barrier is constructed between two vertical edges. The optimal positions for mobile nodes for creating strong barrier coverage in accordance to the existing greedy algorithm are depicted with the red lining in Fig. 2. Upon completing the strong barrier between *x=0* and *x=l*, no intruder can cross undetected from *y=0* to *y=w* (or vice versa).

The proposed algorithm uses different approach. It is based on the high probability of automatic cluster creation in the network. Precisely, in relatively densely deployed networks (such are usually WSNs), after the initial deployment, the nodes will get connected to each other in accordance to the criterion (2), and hence the network will be made of a set of connected sub graphs-clusters (e.g., such are clusters s, j, etc. in Fig.2), and a set of single nodes (e.g., node C in Fig. 2). In order to clarify the algorithm, we will introduce some new terms.

First, we define two important lines: the starting edge of the algorithm $x=0$ and the ending edge of the algorithm $x=l$.

Second, we define concepts related to the clusters as follows. After the initial deployment, for a given cluster p, among all the nodes that belong to the cluster, the node nearest to the line $x=l$ will be defined as the best node $B_p(x_{B_p}, y_{B_p})$ of the cluster p while the node nearest to the line $x=0$ will be defined as the worst node $V_p\left(x_{V_p}, y_{V_p}\right)$ of the cluster p.

We also define the best value and the worst value of the cluster. The best value of the cluster p is $\alpha_p = x_{B_p}$, while the worst value is $\beta_p = x_{V_p}$.

The algorithm works as follows:

1) Deploy the network and initialize the number of mobile nodes to zero, i.e., m=0.
2) If the strong barrier is achieved between $x=0$ and $x=l$, return m=0 and terminate the algorithm. Otherwise, continue.
3) Among all the clusters and single nodes that intersect x=0 with their sensing ranges, choose the one (s) that has the greatest best value.
4) In communication range of the node B_s, find the single node or the cluster t that meets the following criterion:
$$\min\left\{\frac{\beta_t - \alpha_s}{\alpha_t}\right\}, \beta_t > \alpha_s$$
5) Increment the number m as follows:
$$m = ROUNDUP\left\{\frac{\beta_t - \alpha_s - R_{V_t} - R_{B_s}}{{}^3/_2\, R_m}\right\} + m$$
Here, R_{B_s}, $R_{V_t}, and\ R_m$, are the sensing radius of node B_s, the sensing radius of the node V_t, and nominal sensing radius, respectively.
6) Include the cluster t in cluster s, i.e., merge s and t and expand the cluster s.
7) If the strong barrier coverage from x=0 to x=d is not achieved, repeat from 4. Else, terminate the algorithm and return m.

Regarding the calculation of the number m, it presents the quotient between the sensing gap between the nodes and the expected sensing range of the mobile nodes. This range was calculated as follows. Since the sensing radius varies between $\frac{R_m}{2}$ and R_m, the expected mean value is $\frac{3R_m}{4}$. So, the expected sensing range that occupies a mobile node could be estimated with $\frac{3R_m}{2}$.

The same metrics were used in case of both algorithms and the number of mobile nodes was acquired for all the input vectors.

While aware that the proposed algorithm might be less energy-efficient because it generally uses larger communication radius for searching and calculation (which could otherwise be adjusted to lower values), we explore the possibility to recognizing when both algorithms perform approximately equally (and hence the existing one would be more appropriate to be used) and when our algorithm outperforms the actual one for the given input parameters.

3.4 The machine learning approach to algorithm classification

Machine learning (ML) is the area of artificial intelligence that enables the automated discovery of patterns in data. The ML algorithms learn from experience, by inspecting the data structures, relations, and contents [21]. The ML algorithms are mainly used in classification, clustering, and regression.

In order to discover to what extent and in which cases the proposed algorithm may be appropriate for use, as well as to create the decision framework for using one or another algorithm, we conduct an analysis based on machine learning algorithms for classification. We use three algorithms with different classification philosophy, with the accent on decision trees.

The SVMs present one of the most efficient and widely used algorithms today. Based on the vectors from the train set, often by using special transformation functions (called kernels), the SVM tends to map the learning examples from input space to a new high-dimensional feature space in which examples are linearly separable. The aim is finding a hyper plane that maximizes its distances to the support vectors, i. e., to find the weights that gives minimum value of the error function:

$$\phi(w, \xi) = \frac{1}{2} w * w^T + C \sum_i \xi_i \qquad (3)$$

with the following constraint:

$$y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, \ldots n$$

where w is the matrix of coefficients, b is a constant, ξ is a slack variable (i.e., the error tolerance), n is the number of learning examples, and C is a regularization parameter. With the derived values of weights, the hyper-plane can be considered as optimal. This hyper plane now divides the space into two areas: one that is composed of (mainly) members of one class and another that contains (mainly) the members of another class.

Neural networks use complex, non-linear decision boundaries for data classifications. They consist of layered, feed forward, completely connected network of artificial neurons, or nodes. An example of a NN map is given in Fig. 3.

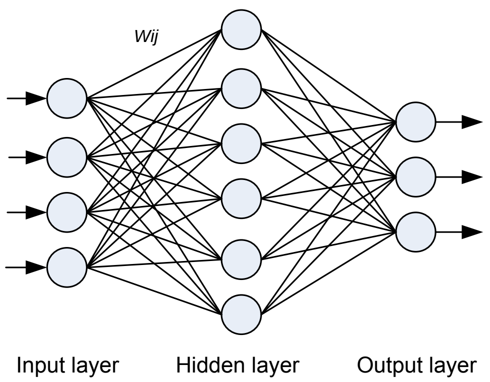

Fig. 3. A NN map.

Each connection has its weight associated with. The weights are randomly assigned to values between zero and 1 in initialization phase. Each node produces the linear combination of the inputs and the connection weights into a single scalar. By using back-propagation approach, the training set is used to set the weights. The derived set of weights is applied on the validation data in order to find the one that minimizes the sum of squared errors.

The NNs are very robust with respect of noisy data. However, unlike decision trees, which produce intuitive rules that are highly descriptive and understandable, neural networks are relatively opaque to human interpretation [22].

On the other hand, besides classification, decision trees are also widely used for data description as well. A variant of the decision tree algorithm, the C4.5 (with its successor 5.0) is a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date [23]. The C4.5 algorithm is based on entropy reduction to selection the optimal splits for the tree nodes. If X is a variable whose k possible values have probabilities $p_1, p_2, \ldots p_k$, the entropy of X is defined with:

$$H(X) = -\sum_j p_j \, log_2(p_j) \qquad (4)$$

Deeper tree usually gives higher precision regarding the training data. However, it can lead to over fitting which implies for the poorer performances on test data. Therefore, for the better generalization, tree is often pruned to the optimal level.

4 Simulation results

A simulation environment for algorithm implementation and testing is developed in Java. Under the same conditions, two algorithms were implemented: the variant of the greedy algorithm, and the cluster based algorithm.

The parameters that were made variable in simulations, along with their maximum and minimum values, are shown in Table 1.

Table 1. The range of the input parameters.

	Minimum	Maximum
Nominal sensing range (R_m)	10 m	40 m
Number of stationary nodes	40	140
The area length (*l*)	600 m	1200 m
The area width (*w*)	150 m	300 m

Even though the sensing range may vary from the order of millimeters to hundreds of meters (depending on the type of sensor), we focus on mid range sensors that are mostly used in similar analysis. The nominal sensing ranges of our sensors were varied from 10m to 40m with the step of 5m. The distribution of the number of sensor nodes were slightly biased between values 80 and 100. In similar analysis, various proportions were used for ROI dimensions, such as 200mx500m [13], 100mx1000m [1], 100mx2000m [14], 200mx2000m [4], etc. In our repeated simulations, for the area width more values were taken between 150m and 210m, while the values on the length of the region were chosen approximately uniformly in range from 600m to 1200m.

With the variation of the mentioned parameters, 160 experimental setups were established with 30 repeated simulations for each setup, which gives the total number of simulations to be 4800. Each setup were represented with the input vector $[R_m, n, w, l]$, and the output vector $[m_g, m_p]$, where m_g and m_p represent the average number of mobile nodes needed to mend the barrier gaps after 30 simulations, according to the existing greedy and proposed algorithm, respectively.

An algorithm is considered to be more efficient if it outputs the smaller number of mobile nodes.

In all of the experiment setups, the proposed algorithm has shown better or equal performances as compared to the existing one. Precisely, it outperformed the existing algorithm in 52,15 % of cases while it had the same performance in 47,85 % of cases.

In order to explore the predictability of when the proposed algorithm outperforms the existing one and when it gives the similar results, the difference $d = m_p - m_g$ were found for each of the 160 vectors. This provides two classes for the classification algorithm – class 1: when $d > 0$, and class 0: when $d = 0$. These two classes were sufficient, since the existing algorithm never outperformed the proposed one.

The main reason for classification is the energy consumption. The existing algorithm is expected to be more energy efficient since it uses smaller communication range. Therefore, in cases when the proposed algorithm does not bring the benefits in the sense of minimizing the number of mobile nodes, the existing algorithm should be used.

In order to explore the ability of the data for classification, as described in section III, we use three algorithms, namely C5.0, NNs and SVMs. All the algorithms used 10-fold cross-validation for performance evaluation and were implemented in MATLAB with the verification in Weka. The results are shown in Table 2.

Table 2. The ability of identifying when the proposed algorithm outperforms the existing one with high probability.

	C4.5	NNs	SVM
Overall accuracy	82.3	76.9 %	82.86 %
Precision (YES)	90 %	78 %	83.3 %
Recall (YES)	71.4 %	73 %	83.3 %
F-measure (YES)	79.6 %	75.4 %	83.3 %
Precision (NO)	77.5 %	76.1 %	83 %
Recall (NO)	92.5 %	80.6 %	83 %
F-measure (NO)	84.4 %	78.3 %	82.9 %

For better generalization, the outcomes of the C4.5 algorithm are given after postpruning. The best results on SVM are derived without space transformation while NNs use a usual three-layer construction with 10 nodes in hidden layer.

From the presented results, it is obvious that the SVM and C4.5 are best choices for classification. In all cases, classification on appropriateness of using one or the other algorithm can be made with the high accuracy of around 80 %. This practically means that the machine learning classification algorithms may be used for decision making on which algorithm is better to be applied for the given input parameters, for the given ratio of $\frac{R_c}{R_s}$, and for the given degree of the energy constraints.

In order to show the importance of the specific attributes on the advantages of the proposed algorithm, a non-pruned tree is shown in Fig. 4.

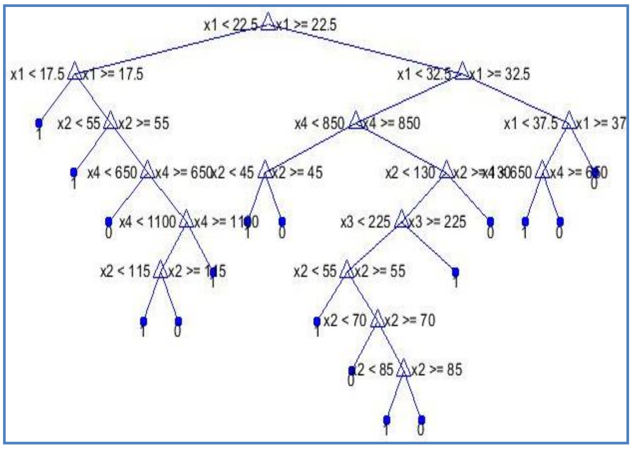

Fig. 4. Non-pruned decision tree shows the importance of the sensing radius.

In Fig. 4, parameters $x_1, x_2, x_3, and\ x_4$, represent *R*, *n*, *w*, and *l*, respectively. The output one means that the proposed algorithm outperforms the existing one, while the output zero means that both have the same output.

From the Fig. 4, it is obvious that the first parameter that influences the outputs is the sensing radius R. Particularly, the classification algorithm shows that, when the sensing radius if small enough, the proposed algorithm outperforms the existing one with a high probability. After pruning, when $R \leq 22.5m$, the introduced algorithm will show better performances in 87 % of cases. The percentage will be even higher if $R \leq 17.5m$. After *R*, the area length and the network density mostly influence the differences in outputs of the algorithms..

5 Conclusions

The paper shows that clustering in densely deployed WSNs can be utilized for optimization on the number of mobile nodes when they are needed to mend the sensing coverage gaps in barrier coverage applications.

The proposed solution can be used in WSN systems for border surveillance and homeland security, when the strong coverage has to be achieved in quasi randomly deployed networks.

The proposed algorithm is tested under the same conditions as the variant of the widely used greedy algorithm. The average output from simulations was taken for each setup.

The proposed algorithm has shown better or equal performances in all the simulations. Precisely, it outperformed the existing greedy algorithm in 52, 15 % of cases.

To the best of our knowledge, for the first time, the performance evaluation of this kind was conducted in context of machine learning algorithms, namely SVMs, NNs, and decision trees, with the accent on the latter one for a more understandable representation. As shown, the data were classifiable with the overall accuracy of nearly 80 % and high f-measures, which imply that the machine learning classification can be used in decision making on which algorithm to be used in a specific implementation.

The results also show that, when the sensing radius is small enough $(R_s < 22{,}5m)$, and especially when it is smaller than 17.5m, the proposed algorithm will almost surely outperform the existing ones.

The derived results are particularly interesting because, in most of the real implementations, the communication is based on IEEE 802.15.4 standard while the sensing radius is often smaller than 18m.

References:

[1] J. Tin, X. Liang, and G. Wang, “Deployment and relocation in mobile survivability-heterogeneous wireless sensor neworks for barrier coverage, *Elsevier Ad Hoc Networks*, vol. 36, no. 1, pp. 321-331, Jan. 2016.

[2] A. Saipulla, B. Liu, and J. Wang, “Finding and mending barrier gaps in wireless sensor networks,” In *Global Telecommunications Conference (GLOBECOM)*, 2010.

[3] Z. Sun, P. Wang, M. Vuran, M.-A. Al-Rodhaan, A.-M. Al-Djelaan, and I.-F. Akyildiz, “BorderSense: Border patrol through advanced wireless sensor network,” *Elsevier Ad Hoc Networks*, vol. 9, pp. 468-477, 2011.

[4] Xu, B., Zhu, Y., Kim, D., Li, D., Jiang, H., Tokuta, A. (2015). Strengthening barrier-coverage of static network with mobile sensor nodes. *Wireless Networks*, vol. 22, pp. 1-10, 2016.

[5] J. Shen, Z. Wang, and Zh. Wang, “Fault tolerant line-based barrier coverage formation in mobile wireless sensor networks,” *International Journal of Distributed Sensor Networks*, no. 11, 2015

[6] Liu, B., Dousse, O., Wang, J., Saipulla, A. (2008). Strong barrier coverage of wireless sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

[7] Li, M., Vasilakos, A. V. (2013). A survey on topology control in wireless sensor networks: taxonomy, comparative study, and open issues. *Proceedings of the IEEE, 101(12)*, 2538-2557.

[8] A. Saipula, B. Liu, G. Xing, X. Fu, and J. Wang, “Barrier coverage with sensors of limited mobility,” In *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010, pp. 201-210.

[9] C.-F. Cheng, T.-Y. Wu, H.-C. Liao, “A density-barrier construction algorithm with minimum total movement in mobile WSNS,” *Elsevier Computer Networks*, vol. 62, pp. 208-220, 2014.

[10] X. Deng, B. Wang, C. Wang, H. Xu, and W. Liu, “Mending Barrier gaps via mobile sensor nodes with adjustable sensing ranges,” *International Journal of Ad Hoc and*

Ubiquitous Computing, vol. 15, no. 1/2/3, pp. 121-132, 2014.

[11] A. Saipulla, B. Liu, and J. Wang, "Barrier coverage with airdropped wireless sensors," *in Proc. of IEEE MILCOM 2008*, pp. 1-7, 2008.

[12] Z. Tafa, "Algorithms on improving end-to-end connectivity and barrier coverage in stochastic network deployments," *In Lecture Notes of the Institute of Computer Science Social Informatics and Telecommunication Engineering*, vol. 89, pp. 81-92, 2012.

[13] T. Liu, H. Lin, C. Wang, K. Peng, D. Wang, T. Deng, and H. Jiang, "Chain-based Barrier coverage in WSNs: toward identifying and repairing weak zones," *Wireless* Networks, vol. 22, pp. 523-536, 2016.

[14] A. Chen, S. Kumar, and T.H. Lai, "Designing localized algorithms for barrier coverage," In *Proceedings of ACM MobiCom*, pp. 63–74, 2007.

[15] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier coverage of line-based deployed wireless sensor networks," *In Proceedings of IEEE INFOCOM*, pp. 127–135, 2009.

[16] W. Wang, V. Srinivasan, K.C. Chua, and B, Wang , "Energy-efficient coverage for target detection in wireless sensor networks," *In Proceedings of IPSN*, pp. 313–322, 2007.

[17] D. Tian and N.D. Georganas, "A coverage preserving node scheduling scheme for large wireless sensor networks," In *First ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32–41, 2002.

[18] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Peas: a robust energy conserving protocol for long-lived sensor networks," In *Proc. ICDCS*, 2003.

[19] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks*, no. 1, 2005.

[20] A. Elfes, "Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception," in Autonomous Mobile Robots: Perception, Mapping, and Navigation, S.S. Iyengar and A. Elfes, Eds. Los Alamitos, CA: IEEE Computer Society Press, pp. 6070, 1991.

[21] Z. Tafa, "Concurrent Implementation of Supervised Learning Algorithms in Disease Detection," *Journal of Advances in Information Technology* Vol. 7, No. 2, pp. 124-128, 2016

[22] D. T. Larose, *Discovering Knowledge in Data.* John Wiley & Sons, 2005.

[23] I. Witten, E. frank, and M. A. Hall, *Data Mining.* Elsevier, 2011.