Journal of
Sensor and Actuator Networks

MDPI

Article

Tracking a Jammer in Wireless Sensor Networks and Selecting Boundary Nodes by Estimating Signal-to-Noise Ratios and Using an Extended Kalman Filter

Waleed Aldosari * and Mohamed Zohdy

Department of Electrical and Computer Engineering, Oakland University, Rochester, NY 48309, USA; zohdym@oakland.edu

* Correspondence: wmaldosari@oakland.edu

Received: 8 October 2018; Accepted: 13 November 2018; Published: 15 November 2018

Abstract: This work investigates boundary node selection when tracking a jammer. A technique to choose nodes to track jammers by estimating signal-to-noise Ratio (SNR), jammer-to-noise ratio (JNR), and jammer received signal strength (JRSS) are introduced in this paper. We proposed a boundary node selection threshold (BNST) algorithm. Every node can become a boundary node by comparing the SNR threshold, the average SNR estimated at the boundary node, and the received BNST value. The maximum sensing range, transmission range, and JRSS are the main parts of this algorithm. The algorithm is divided into three steps. In the first step, the maximum distance between two jammed nodes is found. Next, the maximum distance between the jammed node and its unjammed neighbors is computed. Finally, maximum BNST value is estimated. The extended Kalman filter (EKF) is utilized in this work to track the jammer and estimate its position in a different time step using selected boundary nodes. The experiment validates the benefits of selecting a boundary when tracking a jammer.

Keywords: WSNs; Jammer Received Signal Strength (JRSS); Boundary Nodes Selection Threshold (BNST); Extended Kalman Filter (EKF)

1. Introduction

This work aims to define the threshold value for selecting boundary nodes during jamming attacks. In wireless communication, there are two types of interference: intentional and unintentional interference. Unintentional interference is caused by nearby wireless devices, such as wi-fi, microwave, and Bluetooth signals. A jamming signal is an intentional interference utilizing the same tools to block the wireless channel or wireless device, intended to interrupt a wireless transmission and make a legitimate node unable to use network resources [1]. Due to the shared communication medium of wireless sensor networks (WSNs), jammers efficiently disrupt the communication between nodes by emitting a signal towards the target channel using the same frequency band. The jamming signal harms the wireless channel based on the jammer's position and transmitting power.

Jamming attacks are organized into two levels: the elementary and the advanced, based on the functionality and strategies used [2]. The primary level includes constant jammers, deceptive jammers, and random jammers. A constant jammer transmits a random jamming signal continuously to make the target channel busy. All nodes inside the jammer's transmission range are unable to access the channel to transmit data while the jammer is transmitting its signal. A deceptive jammer sends a regular signal, which leads all sensors near the jammer to switch to received mode until the jammer's battery is depleted or the jammer is physically eliminated. A random jammer switches between sleep

and active mode. The jammer blocks a communication channel for a random time and then turns to sleep mode. The time between sleep and transmit modes is unknown and is randomly selected by the jammer. The advanced level of jamming attacks use a proactive jammer. This kind of attack runs on a frequency domain. The jammer senses all possible channels. Once it detects a channel used by another node, it immediately emits a signal to jam it. The jammer adapts its frequency from one carrier to another, so it can block all channels. This type is difficult to detect because it uses the same method used in the frequency-hopping spread spectrum (FHSS) transmissions and direct-sequence spread spectrum (DSSS) transmissions.

Localizing a jammer in WSNs is necessary to improve existing countermeasures [3]. WSNs are designed as multi-hop networks, in which each node forwards its collected data to the next hop node until reaching the sink node. The routing protocols need to determine the shortest path between the sender and the destination. When jamming attacks occur, the network topology is affected based on the jammer's transmission range and the jammed region. The network topology is classified into three main types based on the jamming signal [3]. Unjammed nodes are sensors located outside the jamming region and can transmit and receive data with neighbors. Boundary nodes may measure the jammer received signal strength (JRSS) directly from the jammer. They can communicate with unjammed and other boundary nodes. Jammed nodes are all sensors located inside the jamming region. These are entirely isolated from networks and cannot send or receive data from other sensors. All data sent to the jamming region fails to be received by the destination. A forced routing protocol is required to direct all data outside the jammed region and prevent resend requests and failed data delivery.

Tracking and localization algorithms are divided into range-free and range-base. Range-free algorithms utilize the geometric node's position and the change of network topology to estimate jammer location. Centroid-based location (CL) and weighted centroid-based location (WCL) are examples of range-free location algorithms [4,5]. Both techniques are sensitive to the sensor's location, the position of the jammer, and the density of nodes.

The existing boundary node selection algorithms are concerned with detecting nodes surrounding holes or the edge of the network. Based on our knowledge, there has been no research conducted to detect boundary nodes during jamming attacks.

Hsieh [6] introduced the Distributed Boundary Recognition Algorithm (BNST RA) to locate boundary nodes distributed around the hole and the network border. The algorithm is based on four levels. First, the author implemented the Virtual Hexagonal Landmark (VHL) to choose Closure Nodes (CNs) found close to the hole and on the border of the network, becoming landmark nodes (LNs). Second, all CNs join and are dubbed Coarse Boundary Cycles (CBCs). Each CBC is then assigned a unique ID to define each hole. Third, all boundaries circling the hole are connected. Fourth, the boundary availability is checked by sending a message to all 1-hop neighbors waiting for a reply.

Khan [7] proposed a Self-Detection Boundary Recognition (SDBR) which examines every node as if it has a closed cycle or a broken path. For the closed cycle, each node constructs a 2-hop neighbor, selecting any node and then moving in one direction to reach the starting point. If this node is a closed path, then it will mark the internal node. Otherwise, it is marked as a boundary node.

Dabba [8] proposed a Border Coverage Protocol (BCP) consisting of three steps. First, the nodes close to the network boundary are detected. Each node broadcasts its location, ID, and sensing range to the base station nodes on the border. Second, all nodes selected in the previous step send the message to their one-hop neighbors. Any node receiving this message mark becomes a transfer node. Third, the boundary nodes selected in the first step are replaced by transfer nodes, which are then considered the new boundary nodes.

In this paper, we have proposed a method to select boundary nodes. Selected boundary nodes have the responsibility of catching the JRSS. We applied an extended Kalman filter to estimate the position of the jammer at each time step. We utilized the signal-to-noise ratio to define boundary nodes while the jammer is moving.

2. Materials and Method

This work aimed to define the threshold value for selecting boundary nodes during jamming attacks. SNR is the difference between the signal received and the unwanted noise, including the jamming signal [9], since the jammer's signal itself is considered noise. SNR is measured in decibels (dB). SNR thresholds are determined by the wireless carrier, as these are system-designed. If the SNR is below the threshold value, the bit error rate (BER) increases, packet delivery ratio (PDR) decreases, and the system fails to decode the signal. It discards the signal and requests retransmission. The jammer is intended to lower the SNR below the threshold value, which disrupts communication between transceivers. Nodes closer to the jammer have a lower SNR, and the amount of SNR decreases until it drops below the threshold value, becoming a jammed node. Based on this observation, we designed an algorithm to compute another threshold value, called the boundary node selection threshold (BNST). The BNST is calculated based on a node's location, sensing range, and the JRSS.

SNR is computed as:

$$\mathrm{SNR} = \frac{P_r(\mathrm{dB})}{N_f(\mathrm{dB})} \;=\; P_r(\mathrm{dB}) - N_f(\mathrm{dB}) \tag{1}$$

The SNR, including the jamming signal, then becomes:

$$\mathrm{JNR(dB)} = \;\frac{P_r}{N_f + \mathrm{JRSS}} \;= P_r(\mathrm{dB}) - \Big(N_f + \mathrm{JRSS}\Big) \tag{2}$$

wherein P_r is the power received by the node, N_f is the noise floor, and JRSS is the jammer's received signal.

This work examines the path loss model. When signals propagate between a transmitter and receiver, they are affected by the surrounding environment. There are many factors that attenuate the signal while traveling through space, such as reflection, diffraction, refraction, and absorption [10]. The path loss model is usually expressed in decibels (dB) and described as follows:

$$L = \; 10\, n \;\log_{10} d, \tag{3}$$

The path loss is represented by L, and n is the path loss exponent, which varies by environment. Typically, the path loss exponent is between 2 and 4, based on the types of metal and objects affecting the signal as it propagates from the sender to the receiver. Variable d is the distance between the transmitter and the receiver. Therefore, the power received by a node becomes the following:

$$P_{ri} = P_t + K - 10\, n \;\log_{10} d_{ij} + X_\sigma \tag{4}$$

where

$$d_{ij} = \; \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5}$$

K is constant and depends on antenna characteristics. P_t is the node's transmission range measured at the reference distance. X_σ is a zero-mean Gaussian distributed random variable. Variables i and j are identifiers for the nodes.

In this network model, we assume that all nodes are randomly deployed and remain unchanged. Sensors have equal transmit power and are equipped with omnidirectional antenna. The data reach their destinations through a multi-hop network, in which each node forwards its collected data to its neighbors. Each node in the network is assumed to have a table of its neighbors' identification and locations. A constant jammer was adopted in this work and implemented using an omnidirectional antenna and fixed transmit power. Because our focus was selecting boundary nodes and tracking, we assumed all jammed nodes would be detected, and the system would know their location and identification.

Figures 1 and 2 illustrate the BNST algorithm.

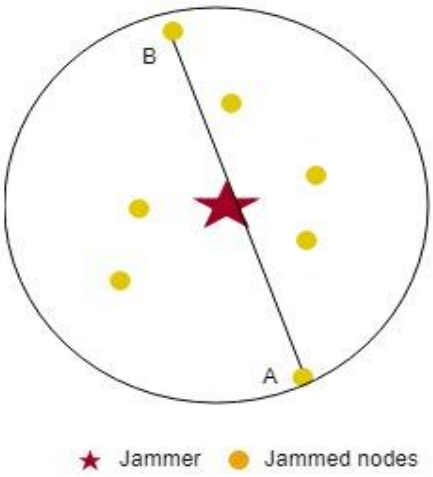

Figure 1. The maximum distance between two jammed nodes.

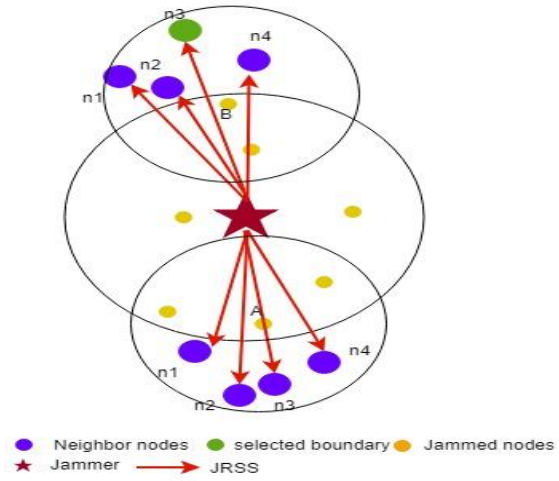

Figure 2. Selecting reference node to compute BNST value.

We calculate BNST as follows. After the jammer begins to function, the jammed node locations are denoted as $N_j = \{(x_i, y_i,)\} i = 1 \ldots m$, where m is the number of jammed nodes. To find the maximum distance between two jammed nodes, the Euclidean distance is used to compute the distance between all nodes inside the jammer region (see Equation (5) above). Therefore, the maximum distance between two nodes becomes $d_{max} = ID_i, ID_j$, where d_{max} is the maximum distance and ID_i and ID_j are the two jammed nodes' identification. In Figure 1, above, (a,b) are the two selected nodes with maximum distance.

Next, we find the farthest neighbor from each selected node using the same method of computing the max distance between two jammed nodes. The red circle n_3 is selected as a reference node to define the BNST as follows:

$$\text{BNST} = \frac{\text{Pr(SR)}}{N_f + \text{JRSS}} \tag{6}$$

Pr (SR) is the minimum power that could be received by nodes at the maximum distance, where SR is the node's sensing range. We used this value to compute the largest SNR at that point. At the first iteration, we select BNST as the maximum value. If the BNST computed at t time step is larger than the selected BNST, the BNST at the first iteration is set to the maximum BNST. Any node with an SNR lower than the BNST value can capture the JRSS. Each node computes the average SNR and compares it to the system threshold and BSNT. If the average SNR is between the system threshold and the BNST, then the nodes become boundary nodes by themselves. Boundary nodes continue to

update their average SNR, and when it drops below the threshold or rises above the BNST, the node will stop tracking the jammer.

$$Ave\ SNR_i = \frac{1}{m}\sum_{i=1}^{m}\frac{pr_i}{Nf + \text{JRSS}_i} \tag{7}$$

where m is the total of neighbor nodes, pr_i is the power received from neighbor i, and $JRSS_i$ is the JRSS at node i. The extended Kalman filter is based on a recursive approach, and it is used in this paper to estimate the jammer position. It is suitable for nonlinear state estimation, and it consists of two main processes: the prediction and update states [11]. The state transition model can be expressed as:

$$x_k = f(x_{k-1}, B_{k-1}) + w_k \tag{8}$$

where x_k expresses the state vector and w_k the process noise, with Q_k a covariance matrix with zero mean and normal distribution $w_k \sim N(0, Q_k)$. In addition, f represents the state transition for nonlinear equations. The measurement vector is expressed as:

$$z_k = \mathrm{h}(x_k) + v_k \tag{9}$$

where $v_k \sim N(0, R_k)$ is the measurement noise with covariance R_k and have normal distribution with zero mean. The nonlinear function is represented by h, and H is the Jacobian matrix. The prediction and update states are expressed as:

Prediction state

$$\hat{X}_{k|k-1} = A_k\hat{X}_{k-1|k-1} + B_k u_k \tag{10}$$

$$P_{k|k-1} = A_k P_{k-1|k-1} * A^T{}_k + Q_k \tag{11}$$

Update state

$$K_k = P_{k|k-1}H_k^T\ (H_k P_{k|k-1}H_k^T + R_k)^{-1} \tag{12}$$

$$\hat{X}_k = \hat{X}_{k|k-1} + K_k\left(Z_k - H_k\ X_{k|k-1}\right) \tag{13}$$

$$P_k = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{14}$$

We apply EKF to validate our algorithm. The position-velocity model is used to track the jammer under constant acceleration [12]. The jammer location is denoted by (x_j, y_j), and the velocity is (v_x, v_y). Therefore, the state vector is expressed as:

$$x = \left[x_j\ y_j\ v_x\ v_y\right] \tag{15}$$

The measurement vector becomes:

$$z = \left[v_x\ v_y\ Pj_1\ \ldots\ldots\ldots Pj_L\right] \tag{16}$$

The observation function and the Jacobian matrix are described as follows:

$$h = \begin{Bmatrix} v_x \\ v_y \\ h_1 \\ \cdots \\ \cdots \\ h_L \end{Bmatrix} \tag{17}$$

where

$$h_i = p_{tj} - 10{\cdot}n{\cdot}\lg\left(\frac{d_{ij}}{d_0}\right) \tag{18}$$

and p_{tji} is the jammer transmission power at reference distance

$$d_{ij} = \frac{\sqrt{x_i - x_j)^2 + (y_i - y_j)^2}}{d_0} \tag{19}$$

$$H = \frac{\partial h(i)}{\partial x(j)} \tag{20}$$

3. Results

To evaluate the performance of the suggested algorithm, 100 sensors were randomly deployed on an area 100 × 100 m. The jammer moved randomly, starting at point 40, 30 before constantly accelerating. The transmission range of the sensors was 15 m. The jammer had a transmission range of 17 m. The transmission power was set to −36 dBm and −35 dBm for the nodes and jammer, respectively. The noise floor was about −95 dBm and the SNR threshold value was 21.8, as presented in Figure 3.

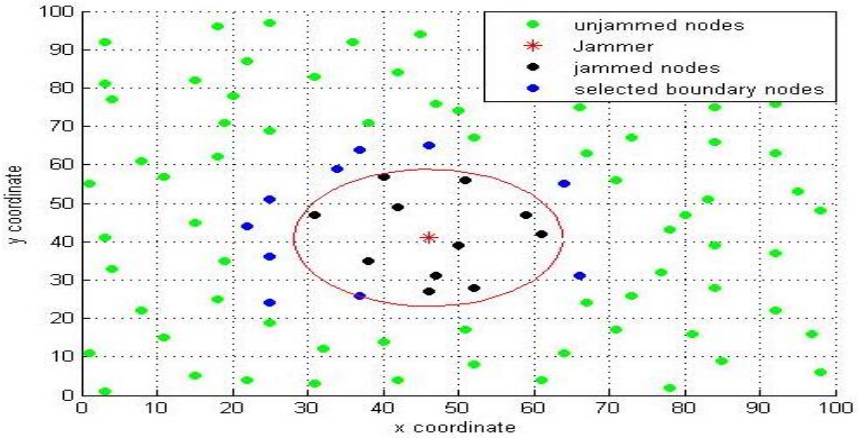

Figure 3. Network deployment.

Figure 4 shows the jammer, and the nodes within the jammer's transmit range are isolated and affected by the jammer's signal.

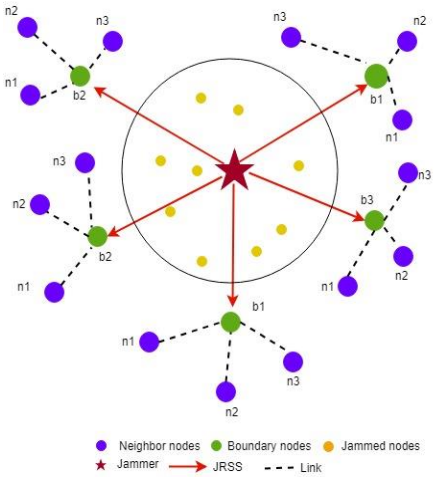

Figure 4. JRSS received by selected boundary nodes.

The red star is the jammer, the yellow circles represent the jammed nodes, green circles are the boundary nodes, and the blue circle denotes the boundary nodes' neighbors. The red arrow is the JRSS received by the boundary node, and the dotted line is the bidirectional link between nodes.

It is clear that the nodes located outside the jamming region are affected by JRSS. When the jammer is moving, the status of the node changes from unjammed to jammed, or a jammed node becomes an isolated node, as shown in Figure 5.

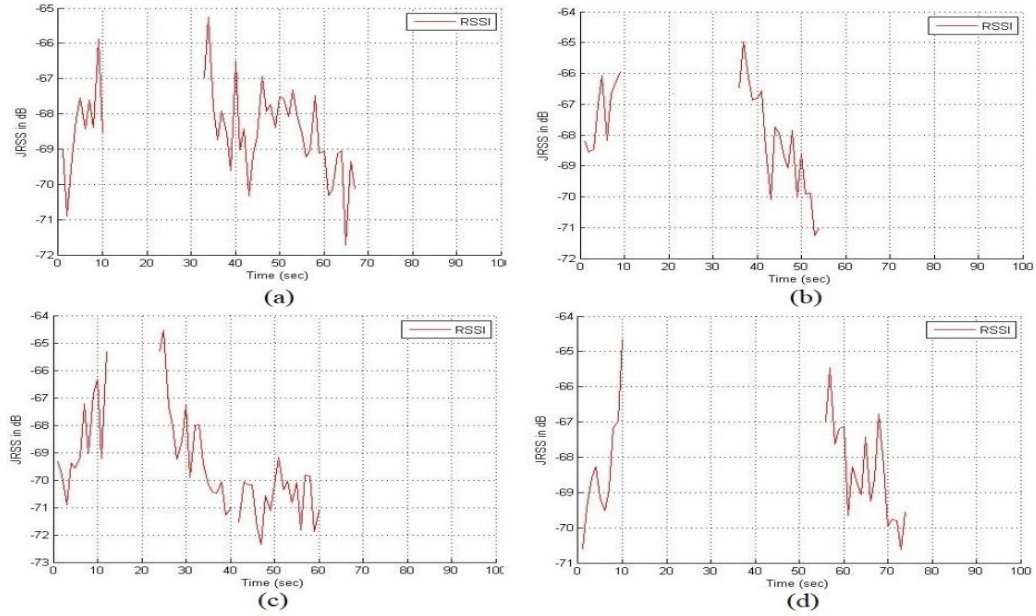

Figure 5. The jammer's signal received by boundary nodes while the jammer is moving: (**a**) tracking information missing in the period from t = 10 s to t = 23 s; (**b**) tracking information missing in the period from t = 10 s to t = 33 s; (**c**) tracking information missing in the period from t = 10 s to t = 22 s and; and (**d**) tracking information missing in the period from t = 10 s to t = 56 s.

This illustration represents boundary nodes affected during jammer movement. In (a), the node detects JRSS at t = 0 and becomes a jammed node in the period from t = 11 s to t = 23 s. In this period, we lost the jammer information. For the same reason, in b, c, and d, the boundary nodes are missing the tracking information while the jammer changes its position. The system stops tracking for the first time when encountering missing data, which t = 11 s shows in Figure 6. For this analysis, defining the boundary nodes while the jammer moving is mandatory. The unjammed nodes received the JRSS as noise, and the SNR decreased based on the amount of JRSS received by the node. Based on this observation, we defined a new threshold for boundary node selection.

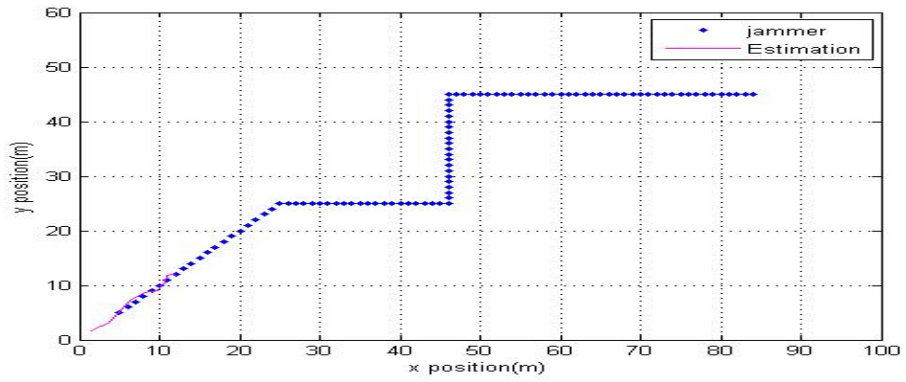

Figure 6. EKF output. The system stops tracking at t = 11 s.

We examined the BNST at time steps 10 s, 20 s, 30 s, and 50 s. Figure 7 represents the average SNR measured by each sensor. Figure 8 shows the BNST estimated with different jammer's transmission range 15 m, 20 m, 30 m and 35 m.

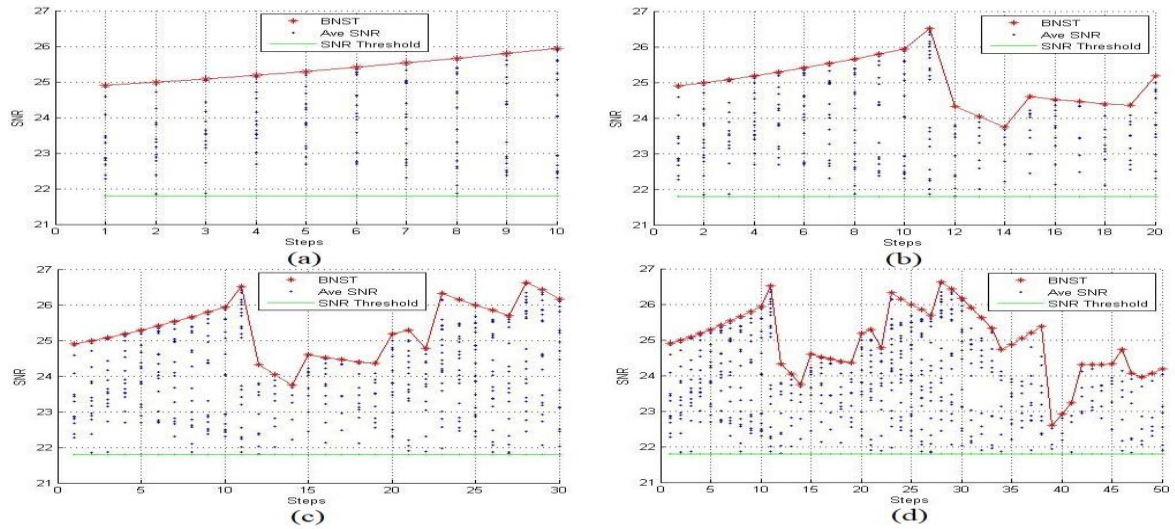

Figure 7. The average SNR computed at boundary nodes: (**a**) time steps = 10 s; (**b**) time steps = 20 s; (**c**) time steps = 30 s; and (**d**) time steps = 50 s.

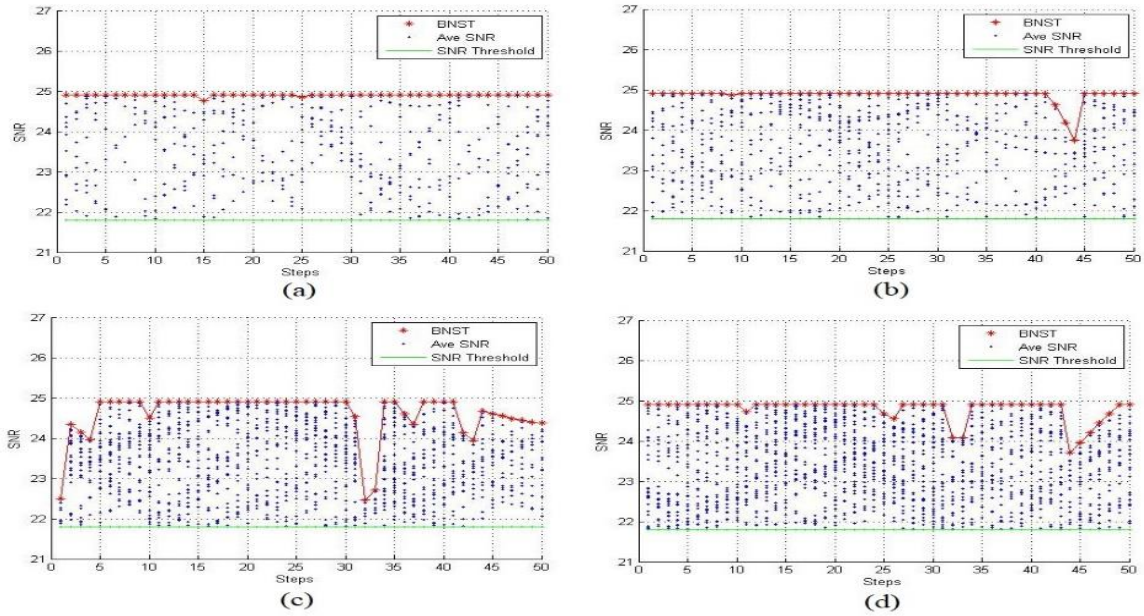

Figure 8. The BNST, the SNR threshold and the average SNR computed at each boundary nodes when the jammer's transmission range is changed: (**a**) jammer's transmission range = 15 m; (**b**) jammer's transmission range = 20 m; (**c**) jammer's transmission range = 30 m; and (**d**) jammer's transmission range = 35 m.

As described in the previous section, all nodes estimate the average SNR and match it with the system threshold, or the SNR threshold and BNST value. If the computed value falls between two these values, the sensor turns itself into a boundary node and begins tracking the jammer until its average SNR drops below the SNR threshold or rises above the BNST value. In Figure 7a, there are ten nodes that became boundary nodes by themselves and starting to read the JRSS. At time step 17 in (b), four elected boundary nodes had an average SNR between the SNR threshold and the BNST.

Figure 9 shows the jammer's estimated location and its real position, found by applying EKF to measure the JRSS by selected boundary nodes at every time step.

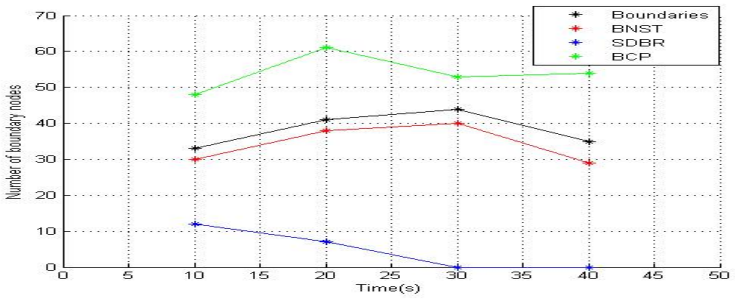

Figure 9. Boundary nodes recognized by BNST, SDBR, and BCP.

In this study, we compared BNST with SDBR and BCP. Unlike SDBR and BCP, because BNST does not depend on four neighboring locations and nodes degrees, BNST performed better and had a higher efficiency at detecting nodes whose transmission ranges intersected with the jammer's transmission range. As shown in Figure 9, at the time step of 10 s, 32 nodes were present around the jammer field, and they could catch JRSS. Please note that BNST recognized 30 nodes, SDBR recognized 12 nodes, and BCP recognized 45 nodes, including those nodes that were incorrectly detected. Because SDBR is based on nodes degree and limited to be equal to or larger than six nodes, it fails to recognize boundary nodes that have less than six nodes on the second hop. Because of the random deployment, location of nodes, transmission range, and neighboring nodes, BCP defined any nodes that had less than four nodes on each side as boundary nodes. BCP was successful in detecting all boundary nodes. Note that nodes that are located far away from the jamming region with their transmission range not intersecting with the jammer transmission range and having less than four neighbors on each side are defined as boundary nodes. Therefore, the number of boundary nodes detected by BCP is always larger than that by BNST and SDBR. When the jammer is closer to the network center, SDBR failed to locate any boundary nodes because the jammer is positioned around the center. Moreover, because of the jamming effect, all nodes deployed around the edge of the network lost most of their neighbors. Figure 10 shows the percentage error of the selected boundary nodes when the jammer's transmission range was adjusted to 15 m, 20 m, 30 m, and 35 m. Note that BNST has a higher efficiency and fewer boundary nodes recognition errors compared to SDBR and BCP. As shown in Figure 10, when the jammer's transmission range was adjusted to 35 m, which indicates more jammed nodes and a longer jamming region was covered by the jammer, SDBR failed to detect the boundary nodes. However, EKF is designed to receive tracking data at each time step by all the detected boundary nodes. As shown in Figure 11, EKF stopped tracking the jammer at t = 32 s because SDBR failed to detect the boundary nodes and EKF did not receive any data. When applied to BCP, EKF failed to track the enemy at all times because there are certain boundaries that were wrongly selected and there was no data feed provided to EKF. However, BNST detected most boundary nodes, and when this data was provided to EKF, it could track the jammer and detect its location at each time step.

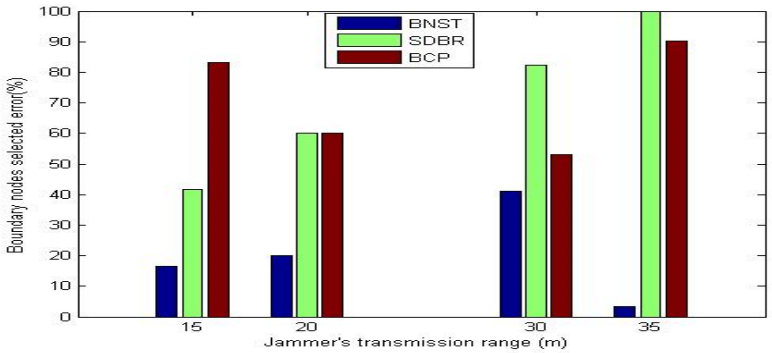

Figure 10. The percentage error of the selected boundary nodes when the jammer's transmission range was adjusted to 15 m, 20 m, 30 m, and 35 m.

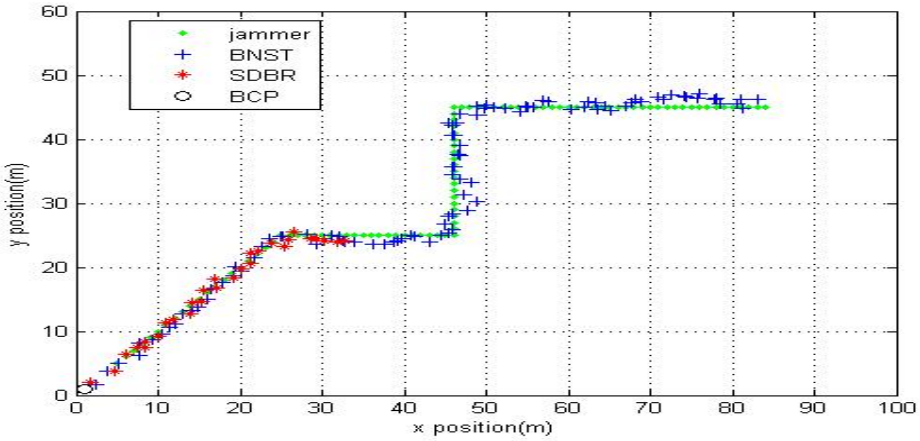

Figure 11. EKF output.

4. Conclusions

In this work, we proposed the BNST algorithm to select boundary nodes while a jammer is moving. The BNST is based on the maximum sensing range that can be reached by the node. A high BNST indicates a greater SNR at that position. By analyzing the SNR threshold and the proposed BNST value, nodes can decide if it is a boundary node, jammed or unjammed. When the sensor becomes a boundary, it starts tracking the jammer until its average SNR drops below the SNR threshold given by the system or becomes larger than the BNST. We evaluate our algorithm in different time steps and by adjusting jammer transmission range. The results show that the system can detect more than boundary nodes at each time step. The EKF tracks the jammer without any interruption or missing data.

Author Contributions: Conceptualization, M.Z. and W.A.; methodology, W.A.; software, W.A.; validation, W.A. and M.Z.; formal analysis, W.A.; investigation, W.A.; resources, W.A.; data curation, W.A.; writing—original draft preparation, W.A.; writing—review and editing, M.Z.; visualization, W.A.; supervision, M.Z.

Funding: Saudi Cultural Mission (SACM).

Acknowledgments: Member of ICCC group at Oakland University in SECS.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Liu, H.; Chen, Y. Error minimizing jammer localization through smart estimation of ambient noise. In Proceedings of the 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), Las Vegas, NV, USA, 8–11 October 2012; pp. 308–316.
2. Grover, K.; Lim, A.; Yang, Q. Jamming and Anti-jamming Techniques in Wireless Networks: A Survey. *Int. J. Ad Hoc Ubiquitous Comput.* **2014**, *17*, 197–215. [CrossRef]

3. Aldosari, W.; Zohdy, M. Localizing Jammer in an Indoor Environment by Estimating Signal Strength and Kalman Filter. *Wirel. Eng. Technol.* **2018**, *9*, 20–33. [CrossRef]
4. Oliveira, L.L.; Dessbesell, G.F.; Martins, J.B.; Monteiro, J. Hardware implementation of a centroid-based localization algorithm for mobile sensor networks. In Proceedings of the 2011 IEEE International Symposium of Circuits and Systems (ISCAS), Rio de Janeiro, Brazil, 15–18 May 2011; pp. 2829–2832.
5. Lee, S.H.; Kang, M.G. Motion tracking based on area and level set weighted centroid shifting. *IET Comput. Vis.* **2010**, *4*, 73–84. [CrossRef]
6. Kun, H.; Jang, S. Hole detection and boundary recognition in wireless sensor networks. In Proceedings of the 2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio, Tokyo, Japan, 13–16 September 2009; pp. 72–76.
7. Khan, I.M.; Khan, M.Z.; Mokhtar, H.; Merabti, M. Enhancements of the self-detection scheme for boundary recognition in wireless sensor networks. In Proceedings of the 2011 Developments in E-systems Engineering, Dubai, UAE, 6–8 December 2011; pp. 448–453.
8. Dabba, A.; Beghdad, R. BCP: A Border Coverage Protocol for wireless sensor networks. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 Augest 2014; pp. 632–640.
9. Misra, S.; Singh, R.; Mohan, S.V.R. Information Warfare-Worthy Jamming Attack Detection Mechanism for Wireless Sensor Networks Using a Fuzzy Inference System. Wireless Sensor Network and Its Application in Advanced Computer Science. *Sensors* **2010**, *10*, 3444–3479. [CrossRef] [PubMed]
10. Goldsmith, A. *Wireless Communications*; Cambridge University Press: Cambridge, UK, 2005; pp. 24–46.
11. Pappas, G.; Zohdy, M. Extended Kalman Filtering and Pathloss modeling for Shadow Power Parameter Estimation in Mobile Wireless Communications. *Int. J. Smart Sens. Intell. Syst.* **2014**, *7*, 898–924. [CrossRef]
12. Kilani, M.B.; Raymond, A.J.; Gagnon, F.; Gagnon, G.; Lavoie, P. RSSI-based indoor tracking using the extended Kalman filter and circularly polarized antennas. In Proceedings of the 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), Dresden, Germany, 12–13 March 2014; pp. 1–6.