Çankaya University Journal of Science and Engineering
Volume 14, No. 1 (2017) 099-124

Evaluation of Range-Free Localization Algorithms Against Node Compromise Attack

Seyed Saber Banihashemian, Fazlollah Adibnia*, Mehdi Agha Sarram

Department of Computer Engineering, Yazd University, Yazd, Iran,
e-mail: s.banihashemian@stu.yazd.ac.ir, fadib@yazd.ac.ir, mehdi.sarram@yazd.ac.ir
**Corresponding author*

Abstract: Different range-free algorithms are proposed for location estimation in Wireless Sensor Networks. In these algorithms, the network is assumed to have no error and false data. This article attempts to evaluate and compare the effect of malicious data produced by node compromised attacks in some of the range-free algorithms: DV-hop, LSVM, and NN. The false data may be produced by the malicious anchor nodes or compromised sensor nodes. The resistance of these algorithms against node compromise attacks is compared. The results show that although DV-hop has less localization error compared to the two other algorithms in a normal condition, in the case of attacks LSVM has less localization error. Further, in this research work, a new criterion is proposed for studying and comparing the border problem issue in the localization algorithms. Using the simulation results from various algorithms, the outcomes have been used for comparison, where it can be considered that LSVM has better performance in the border problem compared with the other studied algorithms.

Keywords: Localization, Wireless Sensor Networks, Range-free, Node compromise attack, border problem.

1. Introduction

Wireless sensor networks are an important technology, and many research works have been done in this field. Recent advances in wireless communications and electronic industries have made it possible to develop multi-functional sensors with low cost and low energy consumption. These sensors are small and can communicate with each other over short distances. Inexpensive, intelligent and networked sensors have introduced new opportunities to control houses, cities and the environment around. Further, sensor networks have a wide

range of functionalities in the military industry and have introduced new capabilities in identifying, investigating and other strategic applications [1].

Supplying all nodes in the network with GPS devices is infeasible due to limitations in the size of sensors, cost, and energy consumption. Sensor nodes with unknown locations can estimate their positions through localization algorithms and by means of available knowledge of distance measurements to some nodes with known locations. The nodes with the known locations are called Anchor or Beacon. These nodes can obtain their locations using the Global Positioning System or setting up with a fixed known coordinates. The nodes with unknown location are called sensor nodes, non-anchor or normal nodes and we use these terms interchangeability. The locations of normal nodes are estimated with the help of the localization algorithms. In this paper, we use normal nodes and sensor nodes interchangeability. Also, we use the term 'nodes' for all sensor nodes and beacons. To obtain the location coordinates of normal nodes, the localization procedure is divided into two parts. In the first part, data such as distance, connectivity, angles between nodes and also the location of anchor nodes are collected [2]. The distance between the neighbors can be measured by the Received Signal Strength [3], Time of Arrival [4] and the Time Difference of Arrival. The distance between multi-hop nodes can be measured by DV-hop [4] and DV-distance [4]. In the second part, the location of an unknown node is determined by means of the collected data.

The localization systems can be categorized in different ways. They can be divided into node-centric and infrastructure-centric [5, 6]. In the former, the sensor nodes compute their locations on their own. In the latter, infrastructure (base station) or trusted nodes identify the locations of sensor nodes. These systems can also be divided into one-hop and multi-hop. In the first approach, normal nodes are localized based on the one-hop neighbors of anchor nodes. In the second approach, distant anchor nodes are also used. Localization systems can also be categorized into range-based and range-free [7]. In the ranged-based systems, geographical distances or angles between nodes are measured in the data collection stage but in range-free systems, there is no need to obtain these measurements. For the resiliency of these algorithms against attacks, some algorithms are proposed for securing the localization algorithms [8, 9, 10, 11, 12, 15].

In the previous proposed multihop range-free algorithms such as DV-hop, LSVM and NN [4, 16, 17], the localization performance is studied and evaluated only in normal conditions. It is assumed that the error free data is used for localization and there is no attack on the network. Although many algorithms are proposed for securing the localization algorithms [8, 9, 10, 11,

12, 15] and for reducing the effect of attacks on the localization process, no studies have been conducted regarding the effect of data errors caused by attacks on the range-free multihop localization algorithms mentioned above. Here, the term attack means the node compromise attack. For instance, in multihop range-free algorithms, if a malicious anchor node announces a wrong location or if a malicious sensor node increases or decreases the number of hops, it will result in false data for the use in the localization process. The initial objective of this article is to analyze the effect of false data on the localization process. This study can be used in later studies to overcome the weakness of the localization algorithms caused by the node compromise attacks. The next objective is to propose a criterion that can be used for studying the border problem issue in isotropic sensor networks. The border problem is an important issue in the localization process and is partially dealt with in [16]. The utilized method in [16] cannot be used for comparing the performance of localization algorithms in the border problem. The structure of this article has been organized as follows: in the next section, the previous related methods are reviewed. In Section 3, the problem statement is highlighted and essential criteria for the evaluation are then argued. In Section Four, results of the simulations are presented and at the end, the conclusion is discussed.

We have used some notations through the paper. Table 1 explains notation used throughout this paper.

TABLE 1. Used notation

$[D_x * D_y]$	dimensions of the deployment area
D_x	the length of the x axis
D_y	the length of the y axis
CX_i	class i on X axis in LSVM and NN algorithms
CY_i	class i on Y axis in LSVM and NN algorithms
N_x	number of location classes on X axis in LSVM and NN algorithms
N_y	number of location classes on Y axis in LSVM and NN algorithms
(x_a, y_a)	the location coordinate of node A
(x'_a, y'_a)	false location coordinate of a malicious node A
$h(A_i, A_j)$	number of hops between the anchor node i and j
$h'(A_i, A_j)$	manipulated hop count of the path between the two anchor nodes i and j
$hopsize_i$	hop size of anchor node i
m	number of malicious anchors
K	set of malicious anchor nodes
C_i^x	location class of anchor node i on X axis
C_i^y	location class of anchor node i on Y axis

2. Related Work

Tran *et al.* [16] proposed LSVM which is a range-free localization algorithm. In this algorithm, the localization is done only using the connectivity data (i.e., only the hop counts); therefore it is simple, and there is no need to use any distance measurement hardware or auxiliary tools, unlike many available range-based techniques. The LSVM algorithm is also based on Support Vector Machines that are used in a suitable approach for localization so that the localization error is kept low. Also, LSVM addresses coverage-hole and border problems in an efficient way. On the other hand, LSVM presents a quick localization method in a distributed fashion which is efficient in using both processing and communication resources.

In this technique, the nodes are randomly placed in a geographic location $[0, D_x] * [0, D_y]$, where $D_x, D_y > 0$. The location classes of nodes are divided into two class sets [16] that have N_x and N_y members on x and y axes, respectively.

- $N_x - 1$ classes for the x axis $\{CX_1, CX_2, \ldots, CX_{N_x-1}\}$ in which each class CX_i includes nodes with $X \geq i\frac{D_x}{N_x}$.
- $N_y - 1$ classes for the y axis $\{CY_1, CY_2, \ldots, CY_{N_y-1}\}$ in which each class CY_i includes nodes with $Y \geq i\frac{D_x}{N_y}$.

Subsequently, a Support Vector Machine is trained for each class, and its coefficients are used for the localization process.

In [17], Chatterjee et al. suggested a localization algorithm denoted as NN, based on multihop connectivity and using Fletcher-Reeves Conjugate Gradient Neural-Network. In this algorithm, first, beacon nodes obtain the distance to each other based on the number of hops and send the collected data along with their locations to a head beacon node. A neural network is trained using the received data by the head node, and then the achieved model of the neural network is used to estimate the location of sensor nodes. Sensor nodes can estimate their positions with the use of the achieved weights and biases. The neural network has two outputs that show location classes on the x and y axes.

In [18], Wang *et al.* used a kernel function to find the degree of similarities among the sensor nodes. The kernel matrix can normally be defined based on the matrix of signal strength. The authors indicated that the relative position of the sensor nodes can be obtained by solving the dimension reduction problem. By means of Kernel Spectral Regression (KSR), relative

locations of sensor nodes can be estimated, and if the number of anchor nodes is sufficient, KSR can be adjusted to a global location.

Chaurasiya *et al.* [19] used a network that consists of normal nodes and anchor nodes. Anchor nodes are equipped with the global localization system. The localization algorithm is split into three parts. First, the algorithm estimates the distance among all the nodes in the network. After that, the algorithm estimates the local coordinates of the nodes within the network by means of MDS. And at last, the local coordinates of nodes are converted to global coordinate systems. The authors state that each node determines the distance to its neighbors and sends it to a central server [19]. After location estimation by the server, the global coordinate is calculated and then sent to the nodes. The received signal strength indicator (RSSI) is used for estimating the distance. The results show that algorithm will not be able to produce results if the node density is lower than a specific value. Besides, the execution time of the algorithm increases exponentially when the number of nodes increases.

A positioning method is suggested by Afzal *et al.* [20] based on machine learning and connectivity data for wireless sensor networks. The deployment area is divided into a number of cells. Anchor nodes gather connectivity data and send this data to the head anchor. The head anchor builds a SVM model based on the received data. The constructed model is sent to all the nodes, and each node calculates its own class. Likewise, all the anchor nodes calculate their own class based on the model and compare these results with their real class location. If an estimated class differs from a real class, it means that the nodes have changed their locations, making the prediction model old and obsolete. Besides, if the anchor node changes its location beyond the range *T*, the network repeats the prediction model. This research work [20] is also assumed that both anchor and normal nodes are dynamic. Since the nodes are dynamic, the movement of the nodes should be tracked. The localization can be done by two methods of location analysis and distance study.

In [21], a localization method was presented which uses the semi-supervised Laplacian regularized least squares. Two types of data are used in this article: the signal strength and the pair-wise distance between the nodes. When the nodes are physically close to each other, the vectors of location data are similar to each other. An optimal kernel function is utilized which is a weighted combination of basis kernel function. To measure the similarity between all sensor nodes, the authors used this kernel function, and further, used semi-supervised Laplacian regularized least squares to establish the relationship among the signal or measured distance space and the physical location space. The locations of non-anchor nodes can be estimated with this relationship.

In [22], a range-free localization algorithm was proposed by Lee et al., based on the multi-dimensional support vector regression. This algorithm is developed as a multi-dimensional regression problem and uses a new MSVR training method to solve regression problem. Finally, the conducted simulations show that the suggested method is efficient in isometric and anisometric networks.

Feng *et al.* [23] utilized hierarchical support vector machines to address the localization problem in WSN. Firstly, H-SVM presents an efficient localization process in a distributed way due to a hierarchical structure. Secondly, H-SVM can estimate the locations of nodes based on only the simple data, i.e., the number of hops, needless of any specific hardware. Finally, the mean and variance of estimation error in H-SVM are seldom considered in the previously mentioned works in [15]. Moreover, this algorithm reduces the training complexity in SVM from $O(n^3)$ to $O(n^2)$.

In [24], Yun *et al.* presented an intelligent approach which uses the received signal strength from anchor nodes. In this research, two schemes are presented. In the first scheme, the localization problem is considered as a unique problem in which the proximity of a sensor node to any anchor node is computed. In other words, the edge weight in each anchor node is used for the calculation of the location of sensor nodes. Using a fuzzy logic system, the edge weights are modeled and optimized by means of a genetic algorithm. In the second scheme, the localization problem is viewed as a single problem and estimates the position of sensor nodes which are mapped from anchor nodes signals and by means of a neural network algorithm. In this article, the best neural network is used based on the number of connected anchors.

Velimirovic *et al.* [25] presented a localization method to solve the problem of uncertainty in received signal strength and the localization error. It is based on the fuzzy set-based called range-Free Fuzzy Ring Overlapping (FRORF). The FRORF scheme isolates a region of the localization space, where the sensor node most likely resides using beacon signals broadcasted by anchors. This method is placed into the range-free and area-based localization methods group.

3. NETWORK AND ATTACK MODEL

Our main focus in this work is on the multi-hop range-free localization algorithms. In the multi-hop approaches, both anchors and sensor nodes are involved in the positioning process. In these types of algorithms, location data are broadcasted by the anchor nodes and these data

are transferred hop-by-hop via the sensor nodes. A localization method is utilized after receiving these data and the position coordinates of a sensor node can be calculated with the assistance of the locations of anchors and hop counts to each anchor nodes. Here, we model a node compromise attack and explain its effect on the three range-free localization algorithms: DV-hop [4], LSVM [16] and NN [17]. What is meant here by the attack is the node compromise attack in which the attacker can take control of the stored data in the sensors or anchors and reuses these nodes in the network after changing the node functionalities. A number of assumptions has been made. It is assumed that all the sensor nodes and the anchor nodes have identical communication ranges. In our assumptions, these nodes are static and no changes occur in the network topology. Further, no hypothesis is made about the security of data stored in the anchor and sensor nodes. In other words, an attacker can obtain the stored data in a sensor node or an anchor node by compromising them. We will consider beacon compromise attack [26] and sensor node compromise attack [26].

3.1. Beacon Compromise Attack

In anchor node compromise attacks, the attacker sends the false location data to the network after compromising a beacon node. These false location data are used by the localization approaches and as the result the wrong location to be estimated. For example, the anchor node A that is located in the geographical position (x_a, y_a) broadcasts the false location data (x'_a, y'_a) to the network as shown in Figure 1. This can result in false location estimation by other non-anchor nodes. Here, we will explain the effect of this attack on the DV-hop [4], LSVM [16] and NN [17] localization algorithms.

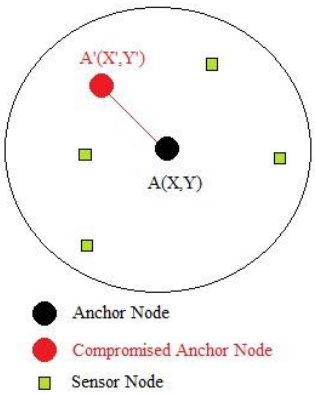

FIGURE 1. Example of Beacon Compromise attack

In the all three considered methods, each anchor node i broadcasts a Hello message containing the position of the anchor and a hop-count field which is initially set to zero. Each receiver

node S keeps the lowest hop-count value to anchor node i. Then, the receiver node S rebroadcasts the Hello message with a hop-count field updated by the increase of hop-count value by one. The receiver node S considers the subsequent received Hello messages of that anchor node i with the greater hop-count field as old data and discards these messages.

In the second stage of the DV-hop algorithm, each anchor node i estimates its hop size using Equation (1).

$$hopsize_i = \frac{\sum_{j\neq i}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j\neq i} h(A_i, A_j)} \tag{1}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the anchor nodes i and j, and $h(A_i, A_j)$ is the hop counts among two anchor nodes i and j, respectively. Then, each anchor node i broadcasts its hop size into the network. After receiving the hop size of anchor i by a sensor node, the sensor node multiplies the hop size by hop counts and estimates the distance to the anchor node i. Subsequently, these estimated distances are used by multilateration technique for calculating the location of non-anchor nodes. If the malicious anchor node k, which has the real geographical coordinates (x_k, y_k) and is a member of the set of m compromised anchors $K = \{k_1, k_2, \dots, k_m\}$, sends the false coordinate data $(x'_k, y'_k) = ((x_k + z), (y_k + w))$. The variable $|z| > 0$ is the difference between real coordinates and advertised false coordinates on the x-axis and the variable $|w| > 0$ is the difference between real coordinates and advertised false coordinates on the y-axis. Equation (1) changes as follows.

$$hopsize_i = \frac{\sum_{j\neq i, j\neq k\in K}\sqrt{(x_i - x_j)^2 + (y_i + y_j)^2} + \sum_{k\in K}\sqrt{(x_i - x'_k)^2 + (y_i - y'_k)^2}}{\sum_{j\neq i} h(A_i, A_j)} \tag{2}$$

$$= \frac{\sum_{j\neq i, j\neq k\in K}\sqrt{(x_i - x_j)^2 + (y_i + y_j)^2}}{\sum_{j\neq i} h(A_i, A_j)} + \frac{\sum_{k\in K}\sqrt{(x_i^2 + x'^2_k - 2x_i x'_k + y_i^2 + y'^2_k - 2y_i y'_k)}}{\sum_{j\neq i} h(A_i, A_j)} \tag{3}$$

$$= \frac{\sum_{j\neq i, j\neq k\in K}\sqrt{(x_i - x_j)^2 + (y_i + y_j)^2}}{\sum_{j\neq i} h(A_i, A_j)} + \frac{\sum_{k\in K}\sqrt{(x_i^2 + x_k^2 + z^2 + 2x_k z - 2x_i x_k - 2x_i z + y_i^2 + y_k^2 + w^2 + 2y_k w - 2y_i y_k - 2y_i w}}{\sum_{j\neq i} h(A_i, A_j)} \tag{4}$$

$$= \frac{\sum_{j\neq i, j\neq k\in K}\sqrt{(x_i - x_j)^2 + (y_i + y_j)^2}}{\sum_{j\neq i} h(A_i, A_j)} + \frac{\sum_{k\in K}\sqrt{(x_i - x_k)^2 + z^2 - 2z(x_i - x_k) + (y_i - y_k)^2 + w^2 + 2w(y_i - y_k)}}{\sum_{j\neq i} h(A_i, A_j)} \tag{5}$$

Therefore, the anchor node compromise attack causes the change in the hop size calculated by the anchor node i using Equation (5). With the hop size change, the attacker can affect multilateration and cause the sensor nodes with an unknown location to estimate the false location.

In the LSVM [16], the false advertised coordinate of a malicious anchor node can be used in the training phase of the support vector machines. This can lead to a misclassification in the localization phase. In the NN algorithm [17], both collected location data and inter-beacon hop-count distances of anchors are used to train neural network. In this algorithm, the collected data for training is as follows:

$$\begin{bmatrix} h(A_1,A_1) & h(A_1,A_2) & \dots & h(A_1,A_k) & : & C_1^x & C_1^y \\ h(A_2,A_1) & h(A_2,A_2) & \dots & h(A_2,A_k) & : & C_2^x & C_2^y \\ \vdots & \vdots & \dots & \dots & \dots & \vdots & \vdots \\ h(A_i,A_1) & h(A_i,A_2) & \dots & h(A_i,A_k) & : & C_i^x & C_i^y \\ \vdots & \vdots & \dots & \dots & \dots & \vdots & \vdots \\ h(A_k,A_1) & h(A_k,A_2) & \dots & h(A_k,A_k) & : & C_k^x & C_k^y \end{bmatrix} \quad (6)$$

Here, $h(A_i, A_j)$ is the hop-count of the shortest path between the two anchor nodes i and j and, C_i^x and C_i^y is the location class corresponding to the anchor node i on x and y axes, respectively. To determine the location class of an anchor node i, all of the deployment area is divided into (N_x, N_y) small virtual cells. The assignment of an anchor node i with the coordinate (x_i, y_i) to a class c in x-dimension or y-dimension is done by the arithmetic relation $(c-1)(\frac{D_x}{N_x}) < x_i \leq c(\frac{D_x}{N_x})$ or $(c-1)(\frac{D_y}{N_y}) < x_i \leq c(\frac{D_y}{N_y})$.

The attacker causes a change in the location class of learning samples by a launching beacon compromise attack in the NN algorithm [17]. In a broad sense, changes in learning data can lead the neural network algorithm to be falsely learned. For instance, the anchor node i with the real geographical coordinate (x_i, y_i) is assigned to the class $C_i^x = \left\lfloor \frac{x_i * N_x}{D_x} \right\rfloor$ and $C_i^y = \left\lfloor \frac{y_i * N_y}{D_y} \right\rfloor$ on x and y dimensions, respectively. Advertising false coordinates $(x'_i, y'_i) = ((x_i + z), (y_i + w))$, where $|z| > 0$ and $|w| > 0$ by a malicious anchor node, the malicious anchor node i is assigned to a different location class as follows:

$$C_{i'}^x = \left\lfloor \frac{(x_i + z) * N_x}{D_x} \right\rfloor = \left\lfloor \frac{x_i N_x + z N_x}{D_x} \right\rfloor = \left\lfloor \frac{x_i N_x}{D_x} + \frac{z N_x}{D_x} \right\rfloor \Rightarrow C_i^x + \left\lfloor \frac{z N_x}{D_x} \right\rfloor \leq C_{i'}^x \leq C_i^x + \left\lfloor \frac{z N_x}{D_x} \right\rfloor + 1 \quad (7)$$

$$C_{i'}^y = \left\lfloor \frac{(y_i + w) * N_y}{D_y} \right\rfloor = \left\lfloor \frac{y_i N_y + w N_y}{D_y} \right\rfloor = \left\lfloor \frac{y_i N_y}{D_y} + \frac{w N_y}{D_y} \right\rfloor \Rightarrow C_i^y + \left\lfloor \frac{w N_y}{D_y} \right\rfloor \leq C_{i'}^y \leq C_i^y + \left\lfloor \frac{w N_y}{D_y} \right\rfloor + 1 \quad (8)$$

A false location class of a malicious anchor can lead to the misclassifications of the location classes of sensors in the localization phase.

3.2. Sensor Node Compromise Attack

In sensor node compromise attacks, an attacker sensor node may change the data in the Hello packets which an anchor has broadcasted into the network. For instance, the compromised sensor node may attempt to modify the position coordinates of an anchor node when the malicious sensor node receives and rebroadcasts the Hello message of that anchor. Or the malicious sensor node may reduce the hop count distances of the received Hello packet, and then broadcasts it into the network. Thus, the attacker sensor node can produce fake location coordinates or bogus hop count distances of some anchor nodes and disturb the hop size calculation process in some other anchors. It is necessary to mention that producing counterfeit location coordinates by malicious sensor nodes does not have any effect on the localization process in the LSVM and NN algorithms. It is because each anchor node individually sends its position coordinate to a head anchor node, and head anchor node use the position coordinates in the training phase of classification tools.

In the first case, an attacker sensor node can change the sent location coordinates of an anchor as shown in Figure 2. In this situation, the anchor nodes that receive these data use false position coordinates in the DV-hop localization algorithm and compute a wrong hop size, leading to an incorrect computation in localization estimation of sensor nodes. In this case, some of the anchor nodes that received correct position coordinates use Equation (1) to determine their own hop size. In the case of fake location coordinates received by an anchor node i, it uses Equation (5), leading to incorrect hop size estimation.

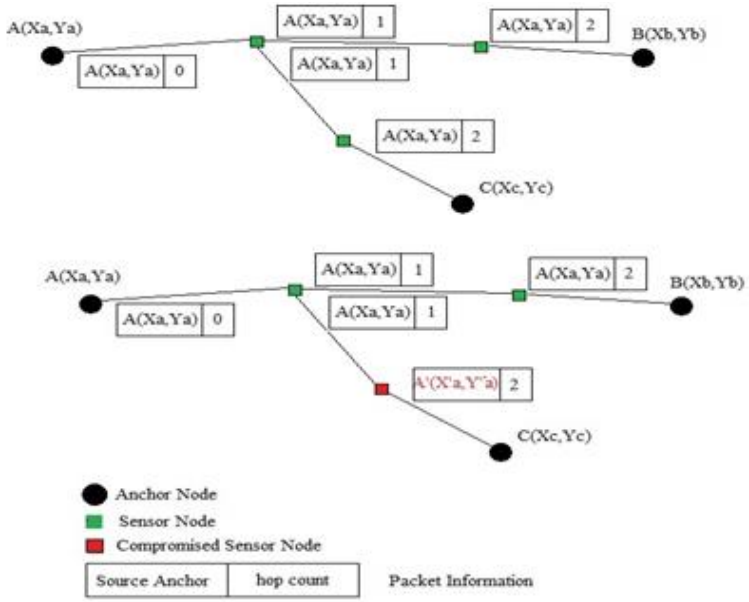

FIGURE 2. Node Compromise attack by the attacker sensor node with the manipulation of the anchor node's location

In the second scenario of a sensor node compromise attack, an attacker sensor node attempts to change the hop count distances of some anchors. In this type of attack, the attacker sensor node tries to make a disturbance in the localization process by decreasing or increasing the hop count distances. In other words, the attacker sensor node manipulates the hop count field of the received Hello packet and rebroadcasts this packet to the next neighbor nodes. In the case of a hop count increase, it is possible that an alternative path is selected as the shortest path by the subsequent nodes (includes sensors and anchor nodes) to the anchor node whose hop count field of its Hello message is changed. However, this article just focuses on the decrease of the hop count fields. An attacker sensor node reduces the hop count field of a Hello message of and anchor node *A* as shown in Figure 3.

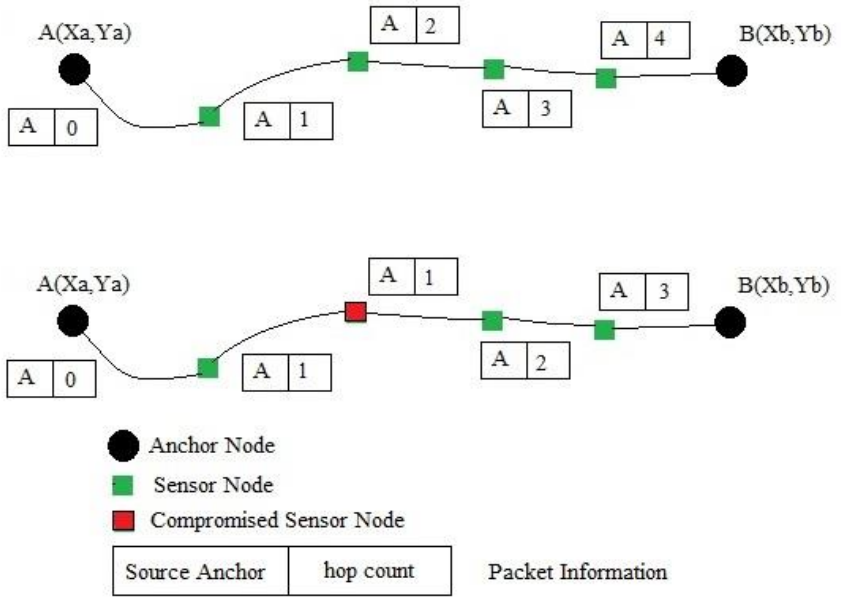

FIGURE 3. Node Compromise attack by changing the hop count

Suppose the anchor node i receives the changed hop count distances from m anchor nodes $K = \{k_1, k_2, \dots, k_m\}$ in a manipulated form of $h'(A_i, A_{k_1}), h'(A_i, A_{k_2}), \dots, h'(A_i, A_{k_m})$ where $h(A_i, A_{k_m}) \neq h'(A_i, A_{k_m})$. Therefore, the anchor node i measures its hop size using Equation (9) in the DV-hop algorithm.

$$hopsize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i, j \neq k \in K} h(A_i, A_j) + \sum_{k \in K} h'(A_i, A_k)} \tag{9}$$

In LSVM and NN algorithms, the hop size between the anchor nodes i and k is changed to $h'(A_i, A_k)$ and is used in the learning phase in these methods.

4. SIMULATION

The MATLAB software is used for the simulating the localization algorithms [4, 16, 17]. It is assumed that the deployment area is a square, which means that $D_x = D_y$ and consequently $N_x = N_y$. The result of each simulation run is shown based on the average of the position errors of all sensor nodes, called as localization error. The neural network algorithm is implemented by the "tansigmoidal" activation function for hidden neurons and the linear activation function for output neurons. The tansigmoidal activation function $f(x)=\frac{1-e^{-x}}{1+e^{-x}}$ converts a real-valued input number to a range between -1 and 1. In other words, large

negative numbers and large positive numbers become -1 and 1, respectively. This function introduces non-linearity into the network when it is utilized in the input and hidden layers [27, 28]. The non-linearity makes multilayer networks more powerful. The output of the linear activation function $f(x) = x$ is equal to the input value. This function is suitable for continuous-valued targets [27, 28]. Later, the simulation parameters and the evaluation criteria shall be explained.

In these simulations, both node compromise attacks are studied. In the sensor node compromise attack, the sensor node attempts to change hop count distances and in the anchor node compromise attack, the compromised anchor node broadcasts the false location coordinates in the network. The amount of position error is calculated based on different parameters and the obtained results are analyzed. The position error of a node is the difference between its estimated point and its real geographical point. The less the difference is, the more the localization accuracy will be.

In the default setting, 256 nodes (including sensor nodes and anchor nodes) are used in the network and the deployment environment is 50*50 meters. The communication range is 7 meters and this communication range is equal for all the sensor and anchor nodes. Also, the network assumed to have the noise-free condition and the sent packet is received by all the nodes within communication range. The initial parameters for simulation are as shown in Table 2.

Each simulation run is repeated 50 times. The sensor and anchor nodes are randomly scattered in the specified area for each repetition. The acquired results are the average of runs.

TABLE 2. Initial Parameters of Simulation

Size	Parameters
50*50 square meter	Deployment Environment
256	The number of all sensor nodes(together with anchor nodes)
64	The number of anchor nodes
7m	Communication Range of nodes
0	The number of compromised sensor nodes
0	The number of compromised anchor nodes
50	Number of runs for each simulation

4.1. Simulation Results

In this section, the effect of different parameters on the localization error of the above algorithms is shown.

4.1.1. Different Number of Sensor Nodes & Anchor Node Compromise Attack

Figure 4 shows the impact of the anchor node compromise attack on the localization methods. To evaluate this parameter, the number of the compromised anchor nodes is considered as 20 and the overall number of nodes is set to 192, 256, 320 and 380 (the number of sensor nodes is considered as 128, 192, 256 and 320, respectively).

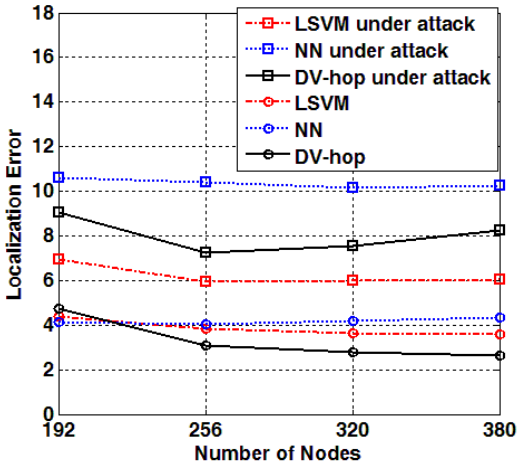

FIGURE 4. Effect of different numbers of sensor nodes & anchor node compromise attacks on localization accuracy

The simulation results show that with the increase in the node density in a normal condition, the accuracy of LSVM and DV-hop algorithms slightly increases and the average localization error partially decreases. Under the same conditions, no changes in the neural network algorithm occur. When an attack occurs and a malicious anchor node sends fake coordinates over the network, different behavior is observed. In the NN algorithm, it is seen that with the rise in the number of sensors, the localization error slightly decreases. However, in the SVM algorithm, a stable behavior is observed except when the number of nodes is 192. This behavior shows that the increase in the sensors has no effect on localization accuracy in the anchor node compromise attack. In the DV-hop algorithm, no distinct behavior is observed. These results show that the localization accuracy of the DV-hop algorithm is highly dependent on the number of the fake location coordinates that is broadcasted in the network.

4.1.2. Different Number of Sensor Nodes & Anchor Node Compromise Attack

In Figure 5, the effect of node compromise attacks on the localization methods is depicted. In the conducted simulation, 192, 256, 320 and 384 sensor nodes are employed, and the number of the compromised sensor nodes is assumed to be fixed as 30 sensor nodes. As demonstrated in Figure 5, the DV-hop algorithm has weaker performance compared to the LSVM and NN algorithms. Besides, the average localization error is increased by six times in the DV-hop

algorithm under the effect of an anchor node attack. As can be seen from Figure 5, this difference is decreased by five times with the rise in the population of the sensor nodes in the network. An analogous behavior can be seen in LSVM and NN algorithms. In other words, with the increase in the number of sensor nodes in the network, the average of the localization error is decreased. Under the sensor node compromise attack, these two algorithms have better performance rather than the DV-hop algorithm. However, the number of sensor nodes in the network has more influence in the NN algorithm.

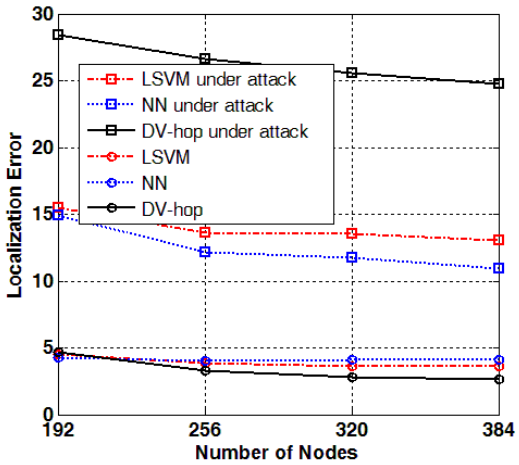

FIGURE 5. The effect of different number of nodes & sensor node compromise attack on localization accuracy

4.1.3. Different Number of Anchor Nodes & Anchor Node Compromise Attack

In this part, the effect of the number of anchor nodes on the localization algorithm is explained. The number of anchor nodes is considered to be 40, 52, 64, 77 and 88. The number of compromised anchor node is taken to be fixed at 20 anchor nodes. As depicted in Figure 6, it is observed that the increase in the number of anchor node has an insignificant impact on the DV-hop algorithm in a normal condition without any malicious anchor node. On the contrary, the LSVM and NN algorithms have a better performance when the number of anchor node increases.

In the case of an attack, it is expected that the average localization error in localization algorithm decreases with the increase in the number of anchors as is presented in Figure 6. As shown in the figure, the LSVM algorithm has a better accuracy compared to the NN and DV-hop algorithms. The effect of such attack noticeably decreases in the LSVM algorithm with the increase in the number of anchor nodes. In other words, when the number of the anchor

nodes increases, the anchor node compromise attack is neutralized to a great extent. This effect is lower in DV-hop and NN algorithms.

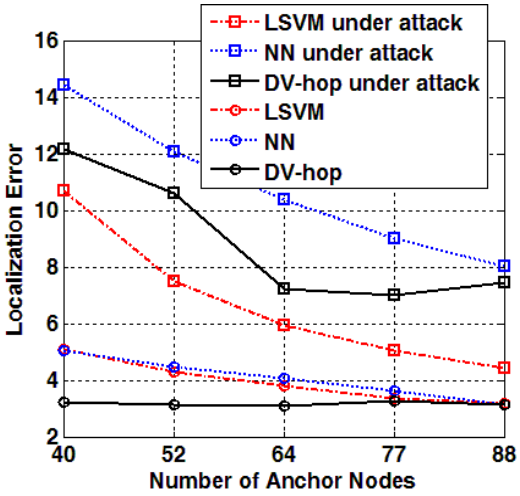

FIGURE 6. The effect of different number of the compromised anchor nodes anchor node compromise attack on the localization algorithms

4.1.4. Different Number of Compromised Sensor Nodes & Sensor Node Compromise Attack

In this part, the effect of the number of the compromised sensor nodes on the localization algorithms is assessed. As shown in Figure 7, the increase in the number of compromised sensor nodes leads to a rise in the average localization error.

As can be seen, when a sensor node compromised attack occurs, the average of localization error increases in the DV-hop algorithm compared to the two other algorithms. However, in the DV-hop algorithm, with the growth in the number of the compromised sensor nodes, the average localization error increases with a steeper slope compared to other two algorithms. That is to say, the DV-hop performance is weaker compared to the two other algorithms when compromised sensor node attack occurs.

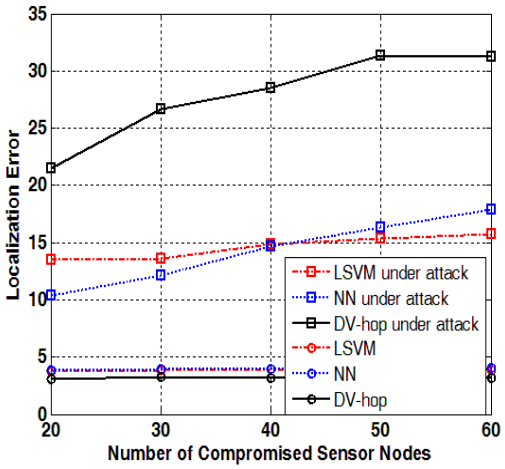

FIGURE 7. The effect of different number of the compromised sensor nodes & sensor node compromise attack on localization accuracy

It is observed that the NN algorithm performs better compared to LSVM based on the average of localization error when an attacker compromise 20 sensor nodes. However, the average of localization error in the NN algorithm increases with the growth in the number of the compromised sensor nodes. If the number of the compromised sensor nodes exceeds 40, its performance is degraded in contrast to the LSVM algorithm. Using the LSVM algorithm, a more stable behavior is observed on condition that a rise in the number of the compromised sensor nodes occurs. With the increase in the number of the compromised sensor nodes, the localization accuracy slightly reduces in the LSVM algorithm.

4.1.5. Communication Range

The communication range has a direct and undeniable impact on the hop-count distances. Therefore, two nodes can exchange messages with each other through a lower hop count if the nodes have a longer communication range. We try to point out the impact of communication range on the localization accuracy in the normal conditions and in the case of an attack. Figure 9 presents the average localization error for a variety of communication range values. From this figure, we can observe that positioning errors of the localization algorithms start to

decline as the larger communication ranges are used. With a communication range larger than 7 meters, the impact of this parameter on the localization accuracy is reduced. We observe a similar behavior for the LSVM and NN algorithms in the case of an attack and no attack. However, in the DV-hop method, we can see that the rate of localization error increases more steeply than the other examined methods with a communication range larger than 7 meters.

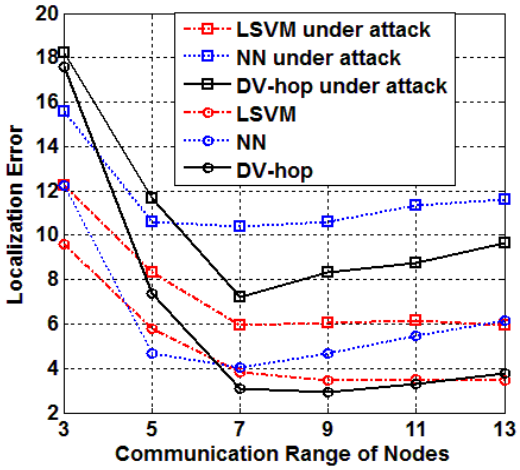

FIGURE 8. Communication range in case of attack and no attack in a network with 256 nodes (64 anchors & 192 sensors)

The results of simulations simulated in section 4.1 are summarized in a tabulated format as follows:

TABLE 3. Summary results of the impact of various parameters on the localization error of positioning approaches in the presence of node compromise attack

	DV-hop	**LSVM**	**NN**
Increase in number of sensor nodes under the anchor node compromise attack	No effect	No effect	Slightly decreases
Increase in the number of sensor nodes under the sensor nodes compromise attack	Increases	Decrease	Decrease
Increase in the number of anchor nodes under the anchor nodes compromise attack	Decrease	Decrease	Decrease
Increase in the number of compromised sensor nodes under the sensor nodes compromise attack	Increases with a steeper slope than the two other	Slight increase	Increases

4.1.6. Border Problem

In this section, an effort is made to analyze the border problem when an anchor node compromise attack occurs. The issue is that, as any sensor nodes get closer to the boundary of the deployment environment, it is likely to have more errors at the time of location estimation. To explain the effect of attacks on the border problem in positioning methods, the effect of the number of sensor nodes on this issue is studied. For this purpose, with the use of initial parameters in Table 1, the number of nodes is considered as 256, 320 and 384 nodes in 3 different simulation runs respectively. The number of the compromised anchor nodes is assumed to be 20 anchor nodes. The deployment environment is divided into circular sectors to assess the border problem. The center of these circular sectors is the midpoint of the deployment field. The average of the localization error of all the sensor nodes located in a sector is considered as the localization error of that sector. In this simulation, the center of the sectors is coordinated (25, 25) and the distance between each sector is considered 4 meters. Figure 9 shows the divisions of these sectors.

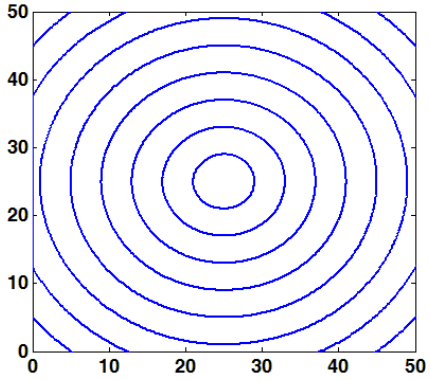

FIGURE 9. The division of deployment environment (50*50) into 4-meter sectors

In Table 4, the average numbers of nodes (including sensor nodes and anchor nodes) in each sector are presented in 3 simulation runs. As it is shown, due to small space of last sectors, the number of nodes declines in these sectors. The results are shown in Figure 10, Figure 11, and Figure 12 for a network with 256, 320 and 384 nodes respectively. Only in the last sector of the first simulation run with 256 nodes, a big jump is observed in comparison with the other sectors as it is shown in Figure 11. This is due to the existence of only a small number of nodes in this sector. In other sectors, a similar pattern is observed for all three simulations that are run. In other words, as we get farther from the center of deployment field, the localization error increases. This pattern becomes more obvious at the time of an attack. Besides, it can be inferred that an increase in the number of nodes has little impact on the border problem by

comparing the obtained results. On the other hand, it can be seen that LSVM has better performance compared to other two algorithms, while in a normal condition without any attacks DV-hop has less localization error.

TABLE 4. The average number of nodes in each sector in 3 simulations with 256, 320 and 384 nodes (including 64 anchors) respectively

Number of nodes / Sectors	nodes=256	nodes=320	nodes=384
Sector 1	2.26	4.42	6.12
Sector 2	10.44	14.26	17.84
Sector 3	18.92	23.90	32.44
Sector 4	25.34	35.06	43.66
Sector 5	33.88	45.12	56.58
Sector 6	41.34	55.08	69.10
Sector 7	34.44	44.30	55.68
Sector 8	14.04	20.62	24.48
Sector 9	2.34	4.24	5.10

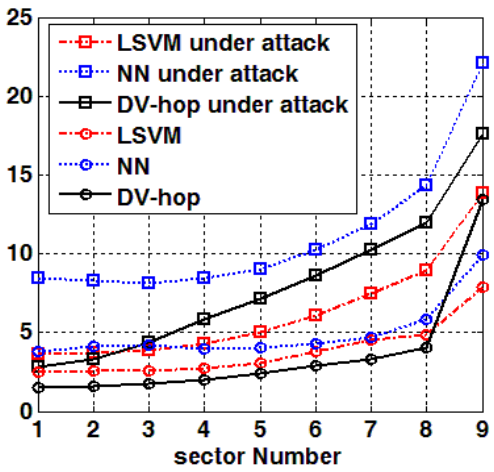

FIGURE 10. Border problem in case of attack and no attack in three runs with 256 nodes

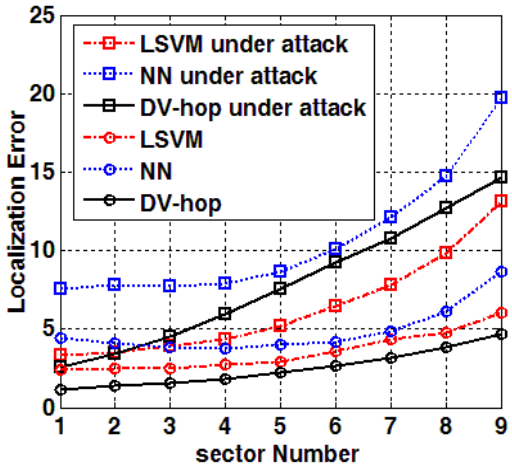

FIGURE 11. Border problem in case of attack and no attack in three runs with 320 nodes

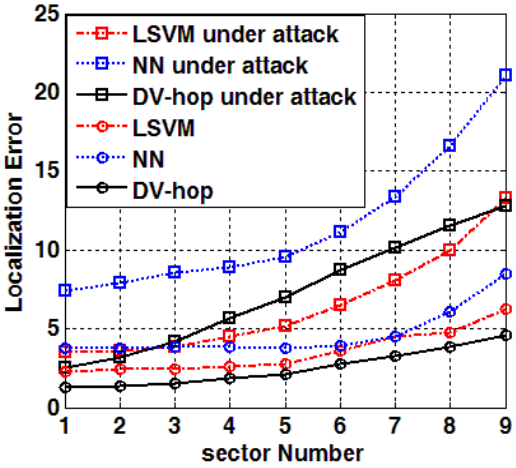

FIGURE 12. Border problem in case of attack and no attack in three runs with 384 nodes

For a further analysis of the border problem, the effect of the number of anchor nodes on this issue is studied. For this purpose, 64, 76 and 88 anchor nodes out of 256 available nodes are considered in 3 different simulation runs respectively. The average number of sensor nodes in each sector is depicted in Table 5. Due to a small area of the first and last sectors, the average number of the sensor node is limited. The results of these simulations are depicted in Figure 13, Figure 14 and Figure 15. In the occurrence of an attack, the increase in the number of anchor nodes affects the localization error in outer sectors, resulting in a decrease of the localization error in these sectors when the number of anchor nodes increases. This issue is more obvious in the LSVM algorithm. On the other hand, if the number of anchor nodes is 88 and no attack occurs, the LSVM, NN, and DV-hop algorithms have a similar average of localization error as shown in Figure 15. However, the DV-hop algorithm has better performance in terms of the average of localization error in inner sectors.

TABLE 5. The average number of sensor nodes in each sector in 3 simulations with 64, 76 and 88 anchor nodes respectively

Number of anchors / Sectors	Anchor nodes=64	Anchor nodes=76	Anchor nodes=88
Sector 1	2.26	2.6	2.58
Sector 2	10.44	8.94	8.64
Sector 3	18.92	17	16.44
Sector 4	25.34	24.60	22.22
Sector 5	33.88	31.80	29.60
Sector 6	41.34	37.72	36.80
Sector 7	34.44	32.72	27.96
Sector 8	14.04	13.28	12.06
Sector 9	2.34	2.34	2.70

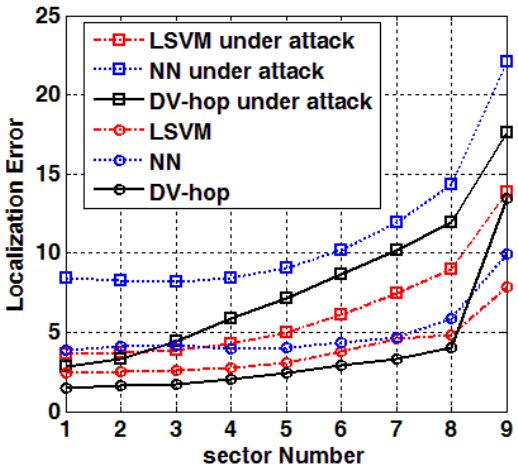

FIGURE 13. Border problem in case of attack and no attack in three runs with 64 anchors

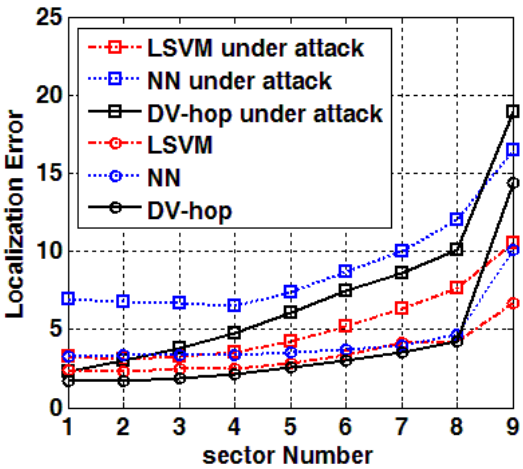

FIGURE 14. Border problem in case of attack and no attack in three runs with 76 anchors

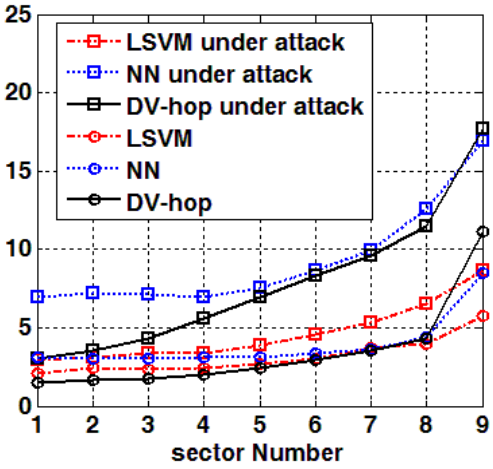

FIGURE 15. Border problem in case of attack and no attack in three runs with 88 anchors

In summary, by comparing the two simulations conducted in this section, it can be stated that the effect of the number of anchor nodes for reducing the localization error in the border problem is more than the effect of increasing the number of sensor nodes that are deployed in the network. A summary of the obtained results is shown in Table 6.

TABLE 6. Summary results of the impact of the anchor node compromise attack on the border problem in the localization algorithms

	DV-hop	**LSVM**	**NN**
Increasing the number of sensor nodes	Reduction in localization error in borderline areas-less average of localization error in inner areas compared to the other two algorithms.	No change in localization error in borderline area-less average of localization error in outer sectors compared to the other two algorithms	No change in the average of localization error in the borderline area.
Increasing the number of anchor nodes	Reduction in localization error in borderline areas-less average of localization error in inner areas compared to the two other algorithms	Reduction in localization error in borderline areas-less average of localization error in outer areas compared to the two other algorithms	Reduction in the average of localization error in borderline areas.

5. CONCLUSION

This paper has made an attempt to explore the effect of sensor node compromise attacks and anchor node compromise attacks on the multi-hop range-free localization algorithms. The performance of the three algorithms LSVM, NN, and DV-hop are compared. Through various

scenarios, simulation runs have been conducted in order to verify the results. First, the effects of various attacks are modeled and described. The sensor node compromise attack and the anchor node compromise attack are specifically modeled to achieve this objective.

The outcome of several simulations for these attacks is carried out and analyzed. The results show that when the anchor node compromise attack occurs, the LSVM algorithm is more accurate compared to the other two algorithms in terms of localization error, and DV-hop is weaker when encountering these attacks. In simpler terms, the attackers can have a more destructive effect on the localization procedure in the DV-hop algorithm rather than two other algorithms. A similar effect is observed in the two other algorithms as well, although it can be said that the two algorithms LSVM and NN have better performance than the DV-hop in the occurrence of a sensor node compromise attack. Considering the occurrence of sensor node compromise attacks, the average localization error in these two algorithms, i.e. LSVM and NN, is similar to each other. On the other hand, the LSVM algorithm acts better than the two other algorithms in the border problem issue when an anchor node compromise attack occurs. However, the performance of the DV-hop algorithm is better in terms of the average of the localization error if no attacks occur.

References

[1] C. Y. Chong, S. Kumar, *Sensor networks: Evolution, opportunities, and challenges*. Proceedings of the IEEE, **91**(8), (2003), 1247-1256.

[2] C. Savarese, J. Rabay, Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks, *General Track of the annual conference on USENIX Annual Technical Conference*, (2002), 317-327.

[3] A. Savvides, C. C. Han, M. Srivastava, Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors, *7th annual international conference on Mobile computing and networking*, (2001), 166-179.

[4] D. Niculescu, B. Nath, Ad Hoc Positioning System (APS), *Global Telecommunications Conference*, (2001), 2926-2931.

[5] S. Capkun, M. Cagalj, M. Srivastava, Securing Localization with Hidden and Mobile Base Stations, *25th IEEE Conference on Computer Communications*, (2006), 1-10.

[6] S. Capkun, K. Rasmussen, M. Cagalj, M. Srivastava, *Secure Location Verification with Hidden and Mobile Base Stations*, IEEE Transactions on Mobile Computing, **7**(4), (2008), 470-483.

[7] T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. Abdelzaher, Range-Free Localization Schemes for Large Scale Sensor Networks, *9th Annual International Conference on Mobile Computing and Networking*, (2003), 81-95.

[8] L. Lazos, R. Poovendran, SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks, *3rd ACM workshop on Wireless security*, (2004), 21-30.

[9] Z. Li, W. Trappe, Y. Zhang, B. Nath, Robust Statistical Methods for Securing Wireless Localization in Sensor Networks, *4th international symposium on Information processing in sensor networks*, (2005), 1-8.

[10] Y. Zhang, W. Liu, Y. Fang, D. Wu, *Secure Localization and Authentication in Ultra-Wideband Sensor Networks*, IEEE Journal on Selected Areas in Communications, **24**(4), (2006), 829-835.

[11] Y. Zhang, W. Liu, W. Lou, Y. Fang, *Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks*, IEEE Journal on Selected Areas in Communications, **24**(2), (2006), 247-260.

[12] R. Garg, A. Varna, M. Wu, *An Efficient Gradient Descent Approach to Secure Localization in Resource Constrained Wireless Sensor Networks*, IEEE Transactions on Information Forensics and Security, **7**(2), (2012), 717-730.

[13] Y. Zeng, J. Cao, J. Hong, S. Zhang, L. Xie, *Secure Localization and Location Verification in Wireless Sensor Networks: A Survey*, The Journal of Supercomputing, **64**(3), (2010), 685-701.

[14] G. Han, J. Jiang, L. Shu, M. Guizani, S. Nishio, *A Two-Step Secure Localization for Wireless Sensor Networks*, The Computer Journal, **56**(10), (2012), 1154-1166.

[15] H. Chen, W. Lou, Z. Wang, J. Wu, Z. Wang, A. Xia, *Securing DV-Hop Localization Against Wormhole Attacks in Wireless Sensor Networks*, Pervasive and Mobile Computing, **16**(1), (2015), 22-35.

[16] D. Tran, T. Nguyen, *Localization in Wireless Sensor Networks Based on Support Vector Machines*, IEEE Transactions on Parallel and Distributed Systems, **19**(7), (2008), 981-994.

[17] A. Chatterjee, *A Fletcher-Reeves Conjugate Gradient Neural-Network-Based Localization Algorithm for Wireless Sensor Networks*, IEEE Transactions on Vehicular Technology, **59**(2), (2010), 823-830.

[18] C. Wang, J. Chen, Y. Sun, *Sensor Network Localization Using Kernel Spectral Regression*, Wireless Communications and Mobile Computing, **10**(8), (2009), 1045-1054.

[19] V. Chaurasiya, N. Jain, G. Nandi, *A Novel Distance Estimation Approach for 3D Localization in Wireless Sensor Network Using Multi Dimensional Scaling*, Information Fusion, **15**(1), (2014), 5-18.

[20] S. Afzal, H. Beigy, *A Localization Algorithm for Large Scale Mobile Wireless Sensor Networks: A Learning Approach*, The Journal of Supercomputing, **69**(1), (2014), 98-120.

[21] J. Chen, C. Wang, Y. Sun, X. Shen, *Semi-supervised Laplacian Regularized Least Squares Algorithm for Localization in Wireless Sensor Networks*, Computer Networks, **55**(10), (2011), 2481-2491.

[22] J. Lee, B. Choi, E. Kim, *Novel Range-Free Localization Based on Multidimensional Support Vector Regression Trained in the Primal Space*, IEEE Transactions on Neural Networks and Learning Systems, **24**(7), (2013), 1099-1113.

[23] V. S. Feng, T. C. Wang, S. Y. Chang, H. P. Ma, Location Estimation in Indoor Wireless Networks by Hierarchical Support Vector Machines with Fast Learning Algorithm, *International Conference on System Science and Engineering*, (2010), 321-326.

[24] S. Yun, J. Lee, W. Chung, E. Kim, S. Kim, *A Soft Computing Approach to Localization in Wireless Sensor Networks*, Expert Systems with Applications, **36**(4), (2009), 7552-7561.

[25] A. Velimirovic, G. Djordjevic, M. Velimirovic, M. Jovanovic, *Fuzzy Ring-Overlapping Range-Free (FRORF) Localization Method for Wireless Sensor Networks*, Computer Communications, **35**(13), (2012), 1590-1600.

[26] A. Boukerche, H. Oliveira, E. Nakamura , A. Loureiro, *Secure Localization Algorithms for Wireless Sensor Networks*, IEEE Communications Magazine, **46**(4), (2008), 96-101.

[27] M. M. Hamed, M.G Khalafallah, E.A. Hassanien, *Prediction of Wastewater Treatment Plant Performance using Artificial Neural Networks*, Environmental Modelling & Software, **19**(10), (2004), 919-928.

[28] W. S. Sarle, *Neural Network FAQ, Part 1 of 7: Introduction*, Periodic Posting to the Usenet Newsgroup Comp. Ai. Neural-Nets. (2002), URL: ftp://ftp.sas.com/pub/neural/FAQ.html.