Iraqi Journal for Computer Science and Mathematics
Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Multi-Strategy Fusion for Enhancing Localization in Wireless Sensor Networks (WSNs)

Mahdi Abed Salman[1*], Muhammed Abaid Mahdi [1]

[1] Department of Computer Science, Faculty of Science for Women (SCIW), University of Babylon, Babylon, Iraq

*Corresponding Author: Mahdi Abed Salman

DOI: https://doi.org/10.52866/ ijcsm.2024.05.01.021
Received August 2023; Accepted December 2023; Available online February 2024

ABSTRACT: Localization in wireless sensor networks (WSNs) plays a crucial role in various applications that rely on spatial information. This paper introduces the Multi-Strategy Fusion for Localization model, which integrates optimization techniques (ABO, DSA, EHO, and KNN) and neurocomputing techniques (BP, MTLSTM, BILSTM, and Autoencoder) to enhance localization accuracy in WSNs. The work is divided into three phases: data collection, model building, and implementation. The first and the last are carried out in the field, while the second is made in the laboratory. The three phases involve a few general steps. The (1) Data Collection Phase includes four steps: (a) Deploy three anchors at known locations, forming an equilateral triangle. (b) Each anchor starts broadcasting its location. (c) Using an ordinary sensor, the RSSI of each anchor is measured at every possible location where the signal of the three anchors can reach it. (d) Data is logged to a CSV file containing the measuring location and the RSSI of the three anchors and their locations. The Model Building Phase includes (a) preprocessing of the collected data, and (b) building a model based on optimization and optimization techniques. The Implementation phase includes five steps: (a) Convoy sensors to the target field. (b) Manually deploy anchors according to the distribution plan. (c) Randomly deploy ordinary sensors. (d) Each ordinary sensor starts in initialization mode. When receiving a signal from three anchors, a sensor computes its location and stores it for future use. When a sensor has its location, it turns into operational mode. (e) A sensor in the operational mode attaches the location with sensed data each time it sends it to the sink or neighbors according to routing protocols (routing is not considered in this study). ABO and DSA optimization techniques show similar performance, with lower Mean Squared Error (MSE) values compared to EHO and KNN. ABO and DSA also have similar Mean Absolute Error (MAE) values, indicating lesser average absolute errors. BP emerges as the top performer among the neurocomputing techniques, demonstrating better accuracy with lower MSE and MAE values compared to MTLSTM, BILSTM, and Autoencoder. Finally; The Multi-Strategy Fusion for Localization model offers an effective approach to enhance localization accuracy in wireless sensor networks. The paper focuses on addressing the correlation between wireless device positions and signal intensities to improve the localization process. The obtained results and provided justification emphasize the significance and value of the model in the field of localization in WSNs. The model represents a valuable contribution to the development of localization techniques and improving their accuracy to meet the needs of various applications. The model opens up opportunities for its utilization in diverse domains such as environmental monitoring, healthcare, smart cities, and disaster management, enhancing its practical applications and practical significance.

Keywords: Localization, RSSI, Wireless Sensor Networks, Optimization algorithms. Prediction Algorithms.

1. INTRODUCTION

During the last decade, Wireless Sensor Networks WSN have been increasingly highlighted through several research papers. WSN is composed of a number of sensors (called nodes) which typically are designed with limited capabilities[1][15]. The first and most important one is to have limited energy. Unfortunately, sensor consumes its energy in communication and processing task [2][16]. Nevertheless, WSNs have an important role in monitoring several environments. They also can be used in unreachable and dangerous fields. Some of applications that use WSN could need the coordinates of sensors.

Localization in wireless sensor networks refers to the process of determining the physical locations of individual sensor nodes within the network. It enables various applications and services that rely on knowing the spatial information of the nodes.

*Corresponding author: Mahdi.salman@uobabylon.edu.iq

Localization in WSNs can be classified into two main categories as explain in Figures 1 and 2: range-based and range-free localization. Range-based techniques utilize direct measurements of signal characteristics to estimate distances or angles. On the other hand, range-free techniques exploit indirect information, such as connectivity patterns or neighborhood relationships, to infer node positions. The localization models include multi types:

(A) TOA (Time of Arrival): TOA is a range-based localization technique where the time taken by a signal to travel from a source node to a destination node is measured. By knowing the speed of signal propagation, the distance between nodes can be estimated, allowing for localization.

(B) TDOA (Time Difference of Arrival): TDOA is another range-based localization technique that involves measuring the difference in arrival times of signals at multiple receiver nodes. By comparing the time differences, the relative distances between the source node and the receivers can be calculated, enabling localization.

(C) RSSI (Received Signal Strength Indicator): RSSI is a range-based or range-free localization technique that measures the power level or strength of the received signal at a sensor node. By analyzing the signal strength, either absolute distances or relative proximity between nodes can be estimated for localization purposes.

(D) AOA (Angle of Arrival): AOA is a range-based localization technique based on measuring the angle at which a signal arrives at a sensor node. By using multiple sensors or antennas, the angle of arrival can be determined, allowing for localization based on the angular information.

The choice of localization technique depends on various factors such as the availability of hardware capabilities, network deployment scenario, required accuracy, and energy constraints. Table 1 show compare among these techniques. While the idea of how achive these techniques shown in Figer 3.

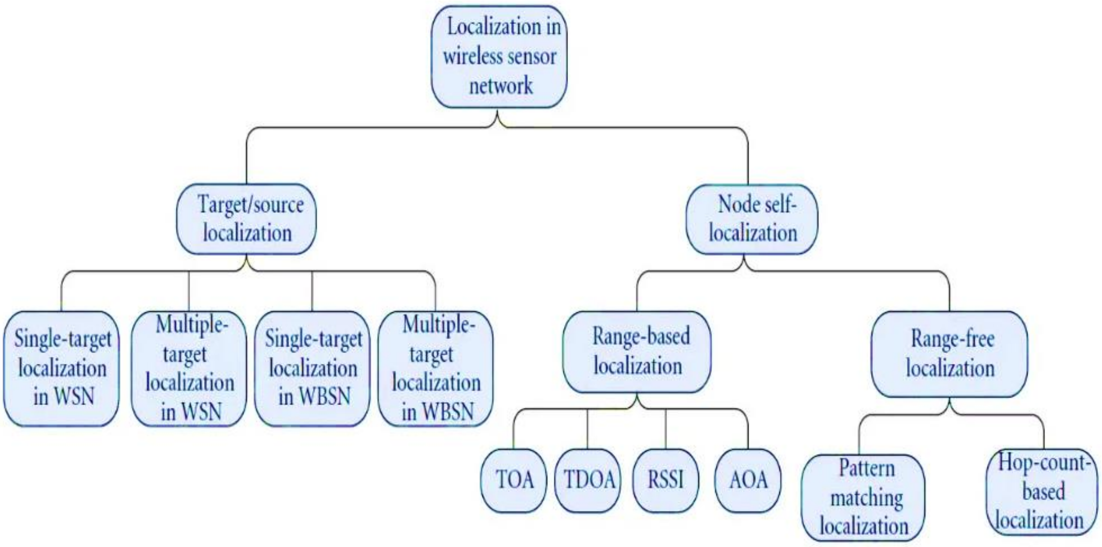


Figure (1): Localization in WSNs

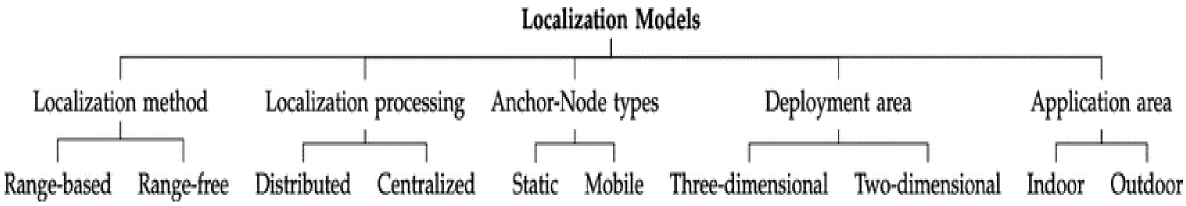


Figure (2): Localization Models

Due to the cost of using GPS unit embedded within each node, several algorithms have been suggested for estimating coordinates of sensors based on referencing coordinates and distance [13][14]. However, they vary in terms of accuracy and cost.

Multi-strategy fusion refers to the process of combining multiple localization strategies or techniques to enhance the accuracy and robustness of localization in wireless sensor networks. Table 2 show a high-level comparison of the different localization techniques commonly used in multi-strategy fusion

TABLE (1): Compression of Range based algorithm

Techniques	Accuracy	Cost	Energy Efficiency	Size of Hardware
TOA	Medium	High	Less	Large
TDOA	High	Low	High	Large
AOA	Low	High	Medium	Large
RSSI	Medium	Low	High	Small

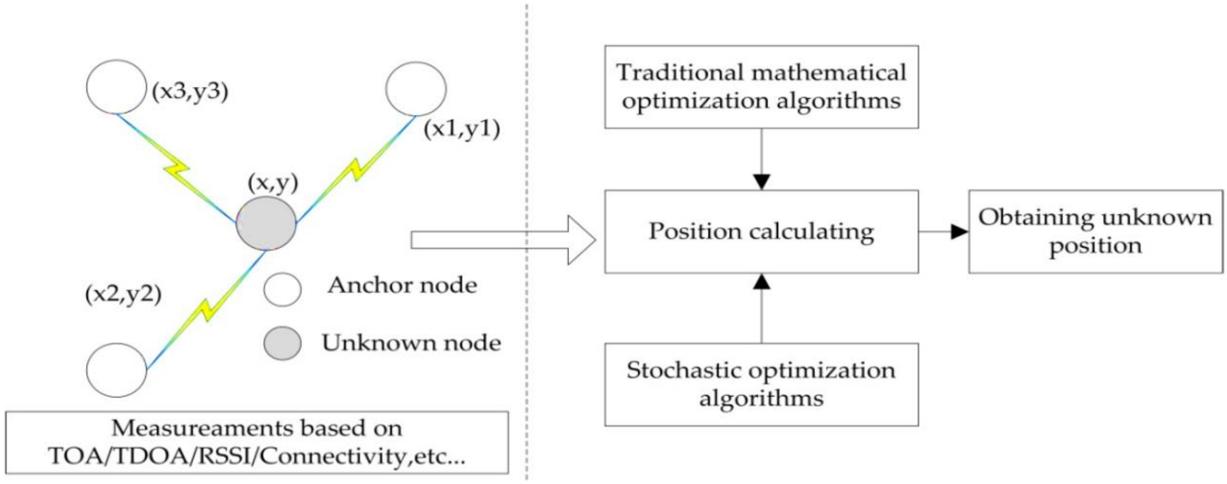


Figure (3): General Processing Localization Models working

TABLE (2) a high-level comparison of the different localization techniques used in multi-strategy fusion

Technique	Description	Advantages	Limitations
Range-Based Techniques	Utilize direct measurements (TOA, TDOA, AOA) to estimate distances or angles between nodes.	Accurate in line-of-sight scenarios.	Susceptible to signal attenuation and multipath effects.
Range-Free Techniques	Exploit indirect information (RSSI, connectivity) to infer node positions.	Simple to implement and energy-efficient.	Less accurate than range-based techniques.
Geometric Approaches	Use geometric calculations (trilateration, triangulation) to estimate node locations.	Accurate when distance or angle measurements are reliable.	Sensitive to measurement errors and noise.
Statistical Approaches	Incorporate probabilistic models (Bayesian inference, Kalman filtering) for fusion.	Consider uncertainties and improve accuracy.	Computationally intensive and require good models.
Machine Learning-Based Fusion	Employ machine learning models (neural networks, support vector machines) for fusion.	Adapt to complex relationships and optimize fusion process.	Require training data and computational resources.

Technique	Description	Advantages	Limitations
Dempster-Shafer Theory	Fuse information using evidence theory, combining multiple sources with degrees of belief.	Handle uncertainty and conflict in data.	Require careful assignment of belief values and thresholds.
Cooperative Localization	Collaborative approach where nodes exchange information to estimate positions together.	Improve accuracy through collaboration and information sharing.	Increased communication overhead and synchronization challenges.

This research presents a new approach that combines the advantages of analysis to main three optimization techniques with three advanced ensemble learning techniques of Multi-Strategy Fusion

2. RELATED WORKS

Several researchers have paid attention to localization in WSN due to its importance. We review the most studies related:

[3] presents a localization and deployment model for wireless sensor networks (WSNs) that utilizes an arithmetic optimization algorithm. The objective is to improve the accuracy of node localization in WSNs. The proposed model aims to optimize the deployment of sensor nodes and enhance the overall localization accuracy. The authors introduce an arithmetic optimization algorithm as the foundation of their localization and deployment model. However, the specific details of the algorithm, such as its formulation or approach, are not provided in the reference. It is likely that the algorithm involves mathematical optimization techniques to optimize the deployment of sensor nodes for improved localization accuracy. The main benefit of the proposed model is its focus on enhancing the localization accuracy of sensor nodes in wireless sensor networks. By utilizing an arithmetic optimization algorithm, the model aims to optimize the deployment of sensor nodes, which can potentially lead to improved localization performance.

[4] addresses the problem of target localization in wireless sensor networks (WSNs) using received signal strength (RSS) measurements and convex relaxation techniques. The authors propose a method that estimates the location of a target based on the RSS values collected from multiple sensor nodes in the network. The main benefit of the proposed method is its utilization of received signal strength (RSS) measurements for target localization in wireless sensor networks. RSS-based localization techniques have been widely studied and can provide a cost-effective solution for estimating the location of targets in WSNs.

[5] presents a location discovery scheme for wireless sensor networks (WSNs) based on directionality. The authors aim to address the challenge of accurately determining the location of sensor nodes in a WSN using the concept of directionality. it can be assumed that the scheme involves utilizing the directional properties of sensor nodes to estimate their locations. This may involve techniques such as angle of arrival (AOA), beamforming, or other methods that exploit the directional characteristics of wireless signals.

[6] propose a mobile anchor-assisted localization algorithm for wireless sensor networks (WSNs) based on regular hexagons. The algorithm utilizes the movement of mobile anchors and regular hexagons to estimate the location of sensor nodes in the network. By leveraging the information provided by the mobile anchors, the algorithm aims to improve the localization accuracy. The authors present simulation results and compare their proposed algorithm with other localization methods, demonstrating enhanced localization performance.

[7] introduce a trilateral centroid localization and modification algorithm for wireless sensor networks (WSNs). The algorithm estimates the location of sensor nodes based on trilateration and centroid calculation techniques. Trilateration involves using distance measurements from at least three anchor nodes to determine the position of a target node. based on RSSI. Estimating the unknown locations depend on three steps: (1) determining the distance between unknown nodes and three anchors. (2) estimating the location of unknown nodes. (3) calculate the error to correct the location.

[8] present a simulation and analysis of an RSSI-based trilateration algorithm for localization in Contiki-OS, an operating system for the Internet of Things (IoT). The algorithm utilizes RSSI measurements, which represent the received signal strength indicator, to estimate the location of sensor nodes. The authors provide simulation results, compare the algorithm's performance with other localization methods, and analyze its accuracy in terms of localization. authors used trilateration with RSSI using strength of signal to estimate the unknown locations.

[9] focuses on localization in wireless sensor networks (WSNs) using a modified trilateration technique based on fuzzy optimization. The authors aim to improve the accuracy of trilateration by incorporating fuzzy optimization techniques. The algorithm has two phases: (1) using trilateration to estimate the locations. (2) using fuzzy weighted approach in centroid technique. This algorithm achieved good accuracy and cost (minimum number of anchors).

[10] introduces an improved centroid DV hop-based algorithm for localization in wireless sensor networks (WSNs). The authors aim to enhance the accuracy and efficiency of localization by refining the existing centroid DV hop algorithm. Three algorithms are suggested in [10] for localizing the position of unknown nodes. DV hop algorithm achieves good results in term of comparing with range free algorithm, however low accuracy comparing with range based algorithm. 2D Hyperbolic Algorithm is used to estimate the location of the node and the last algorithm is centroid algorithm to estimate the unknown locations using the nearest anchors.

[11] Present method include two stages: Stage 1: Local Modified Strong Tracking Filtering Estimator: (a) Designing a modified strong tracking filtering estimator in each cluster head of the WSNs. (b) Compensating for uncertainties in system modeling and noise covariances using a fading factor. (c) Obtaining local estimates for maneuvering target tracking. Stage 2: Multi-Rate Fusion Estimator: (a) Designing a multi-rate fusion estimator to generate a fused estimate with higher precision. (b) Considering energy efficiency and tracking accuracy in the fusion process. Main contribuitions of it: (a) Introduction of a multi-rate fusion strategy and a hierarchical two-stage fusion structure for maneuvering target tracking in WSNs. (b) Design of a local modified strong tracking filtering estimator in the first stage to obtain local estimates with improved robustness.(c) Development of a multi-rate fusion estimator in the second stage to generate a fused estimate with higher precision. (d) Presentation of a designed E-puck robot tracking platform for experimental validation. Table (3) shown compare among the privouse works based on methodology used; main results and the main befits.

TABLE (3)**: Compare among the previous works**

Paper	Author(s)	Year	Proposed Method	Main Results	Main Benefit
[3]	S. J. Bhat, and S. KV	2022	Localization and deployment model using arithmetic optimization algorithm	Improved localization accuracy	Enhanced localization accuracy through optimization
[4]	W. Ding, Q. Zhong, Y. Wang, C. Guan, and B. Fang	2022	Target Localization based on RSS and Convex Relaxation	RSSI	Utilization of RSS and convex relaxation for localization
[6]	G. Han, C. Zhang, J. Lloret, L. Shu, and J. J. Rodrigues	2014	Mobile anchor-assisted localization algorithm based on regular hexagons	Improved localization accuracy compared to other methods	Utilization of mobile anchors and regular hexagons for localization
[7]	L. Yujun, and C. Meng	2015	Trilateral centroid localization and modification algorithm for wireless sensor network	RSSI	Utilization of trilateration and centroid calculation for localization
[8]	R. Valli, A. Sundhar, V. Vignesh, and S. Kotari	2016	RSSI-based trilateration algorithm for localization in Contiki-OS	Improved localization accuracy compared to other methods	Utilization of RSSI measurements for localization
[9]	R. Y. Jyoti, and N. Singh	2013	Localization using modified trilateration based on fuzzy optimization	RSSI	Utilization of fuzzy optimization for improved trilateration
[10]	H. Agarwal, A. Dwivedi, and A. Kaur	2017	Improved centroid DV hop based algorithm	enhance the accuracy and efficiency of localization by refining the existing centroid DV hop algorithm	Estimate the unknown locations using the nearest anchors
[11]	X. Yang, W. -A. Zhang, L. Yu, K. Xing	2016	Multi-Rate Distributed Fusion Estimation for Sensor Network-Based Target Tracking	Satisfactory estimation precision with reduced sampling and estimation rates	Energy-efficiency and tracking accuracy in target tracking

3. ADVANTAGEOUS TECHNIQUES

This section shown the techniques will used in this paper

3.1Optimization Techniques

Optimization aims to find the best solution among numerous possibilities in a problem space using various techniques, classify into four main types (optimization of single object function, optimization of multi object functions, single object function with constrictions and optimization of multi object functions with constrictions). A visual representation of these optimization techniques is shown in Figure 4.

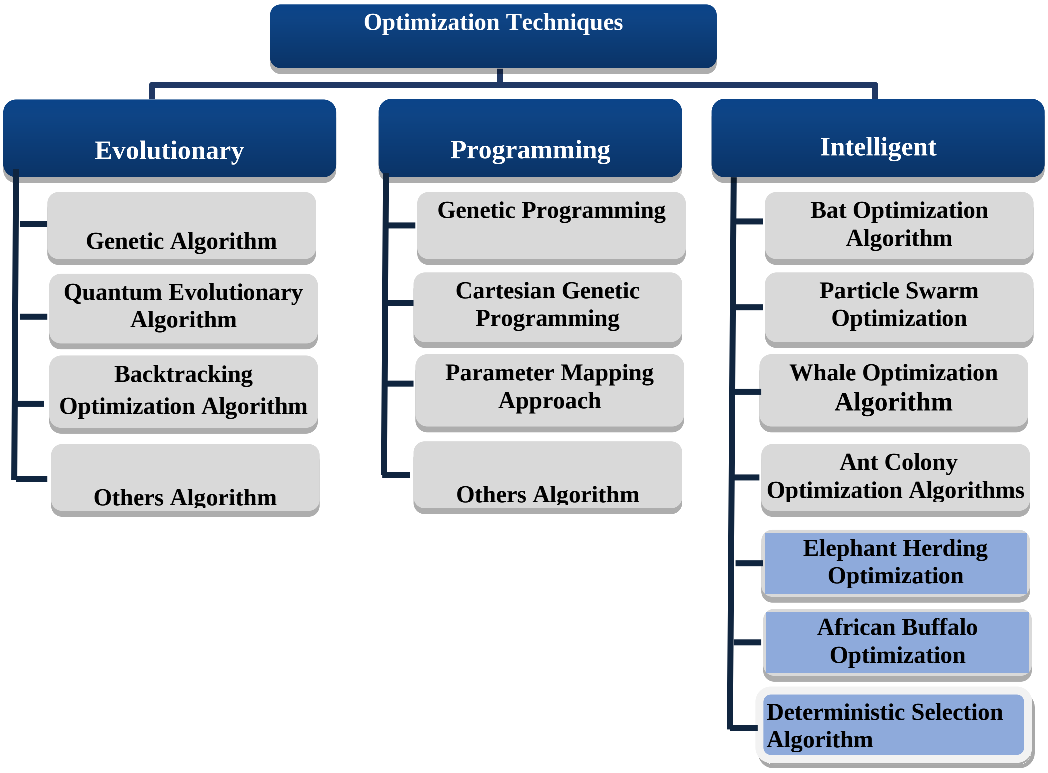


Figure (4): Optimization Techniques

3.1.1 Elephant Herding Optimization (EHO)

EHO is a population-based metaheuristic optimization technique that simulates elephant herding behavior. The goal of EHO is to find a solution to various optimization tasks by determining the best and worst solutions based on two operators: the clan operator and the separating operator. The clan operator updates the position of individuals in the clan based on their relationship with the leader, while the separating operator simulates male elephant behavior by representing the worst position in each iteration. Algorithm 1 shows the details of the EHO algorithm. ([17][18]. In the EHO algorithm, the number of clans is represented by cl, new individuals are denoting as NX and the scale factor and random factor are represented by α and β, respectively

---

Algorithm 1: EHO

Input: N, D, Max_x, Min_x

Output: *Global* $\boldsymbol{X_{best}}$

//Initialization

1: ***For** each i in N*

2: | *Compute the fitness* $f(X_i)$*of each* $\boldsymbol{X_i}$

3: ***End For***

4: *Divided N in to set of cl*
5: ***While** not meeting the stop conditions do*
6: *rank all* $f(X_i)$
7: ***For** each cl in N* *//clan operator*
8: ***For** each i in cl*
9: $NX_{cl,i} = X_{cl,i} + \alpha * (X_{best,cl} - X_{cl,i}) * R1$
10: ***IF*** $X_{cl,i} = X_{best,cl}$
11: $X_{middle,cl,d} = \frac{1}{n_{cl}} * \sum_{k=1}^{n_{cl}} X_{cl,k,d}$
12: $NX_{cl,i} = \beta * X_{mi,cl}$
13: ***End IF***
14: ***End For***
15: ***End For***
16: ***For** each cl in N* *// separating operator*
17: $X_{worst,cl} = Min_x + (Max_x - Min_x + 1) * R_2$
18: ***End For***
// Evaluate individuals based on its position
19: ***End while***
20: ***End EHO***

3.1.2 African Buffalo Optimization Algorithm (ABO)

ABO is a population-based metaheuristic technique inspired by African Buffaloes alarm and alert calls during their activities of foraging and defending. Algorithm 2. shows the details of the ABO algorithm. [19][20] In the ABO algorithm, learning factors are represented by LP_1 and LP_2, while moves of exploration and exploitation are represented by mw_i and m_i, respectively. Additionally, the time unit is denoted by λ.

Algorithm 2: ABO

Input: *N, D,* LP_1 *,* LP_2
Output: *Global* $\boldsymbol{X_{best}}$
//Initialization of the population
1: ***For** each i in N*
2: *Compute the fitness* $f(X_i)$ *of each* $\boldsymbol{X_i}$
3: ***End For***
4: ***While** not meeting the stop conditions do*
5: ***For** each i in N* *// update* X_i *exploitation*
6: $m_{i+1} = m_i + LP_1 * R_1\,(f(X_{best,i}) - mw_i) + LP_2 * R_1(X_{best,i} - mw_i)$
7: $m_{i+1} = \frac{mw_i + m_i}{\lambda}$
8: ***IF*** X_{be} *updating*
Go to 13
9: ***Else***
10: *Go to 2*
11: ***End IF***
12: ***End For***
13: ***End while***
14: ***End EHO***

3.1.3 Deterministic Selection Optimization Algorithm (DSA)

DSA is a method used in various optimization problems to select the most promising solutions from a set of candidate solutions. In the DSO, each candidate solution is evaluated based on a fitness or objective function that quantifies its quality or performance. The algorithm aims to identify the best solutions that are likely to lead to better results in subsequent iterations. The algorithm follows these steps: (A) Each candidate solution is evaluated using the fitness function, which calculates a numerical value representing the quality of the solution. (B) The solutions are ranked based on their fitness values, with the highest-ranked solutions considered the most promising. (C) The top-ranked solutions are selected for further processing or reproduction, while lower-ranked solutions may be discarded or subjected to evolutionary operators such as mutation or crossover. (D) The selected solutions are used to generate new candidate solutions for the next iteration. This can involve processes like crossover, where elements of multiple solutions are combined to create new offspring solutions, or mutation, which introduces small random changes to the selected solutions. The Deterministic Selection algorithm ensures that the best-performing solutions have a higher chance of being preserved and used in generating new solutions. By iteratively applying this process, the algorithm attempts to converge towards an optimal or near-optimal solution for the given optimization problem.

Algorithm 3: DSA

```
Input: L (list), k (integer)
Output: Result (selected element)
IF L has 10 or fewer elements then
    Sort L
    Return the element in the kth position
End if
Partition L into subsets S[i] of five elements each (n/5 subsets total)
for i = 1 to n/5 do
    x[i] = select(S[i], 3)
End for
M = select({x[i]}, n/10)
Partition L into L1 < M, L2 = M, L3 > M
if k <= length(L1) then
    Return select(L1, k)
else if k > length(L1) + length(L2) then
    Return select(L3, k - length(L1) - length(L2))
else
    Return M
End DSA
```

Where; the algorithm takes a list L of n element and an integer k as input and returns the selected element based on the kth position. The algorithm first checks if the list L has 10 or fewer elements. If so, it sorts the list and returns the element in the kth position. If the list has more than 10 elements, it proceeds to partition L into subsets S[i], each containing five elements. It then recursively calls itself with each subset S[i] to select the third smallest element x[i] from each subset. Next, the algorithm selects the median element M from the array of x[i] values by recursively calling itself with the array of x[i] and selecting the element at the n/10 position. The list L is then partitioned into three parts: L1 containing elements less than M, L2 containing elements equal to M, and L3 containing elements greater than M. Based on the value of k, the algorithm either recursively calls itself with L1 if k is less than or equal to the length of L1, or with L3 if k is greater than the combined length of L1 and L2. Otherwise, it returns M if k falls within the range of L2.

3.2 Prediction Techniques

Prediction is finding the event will occurs in the future based on set of current facts. The results of prediction are true if the predictor build from the facts otherwise become virtual. Figure 5 provides more information on some prediction techniques.

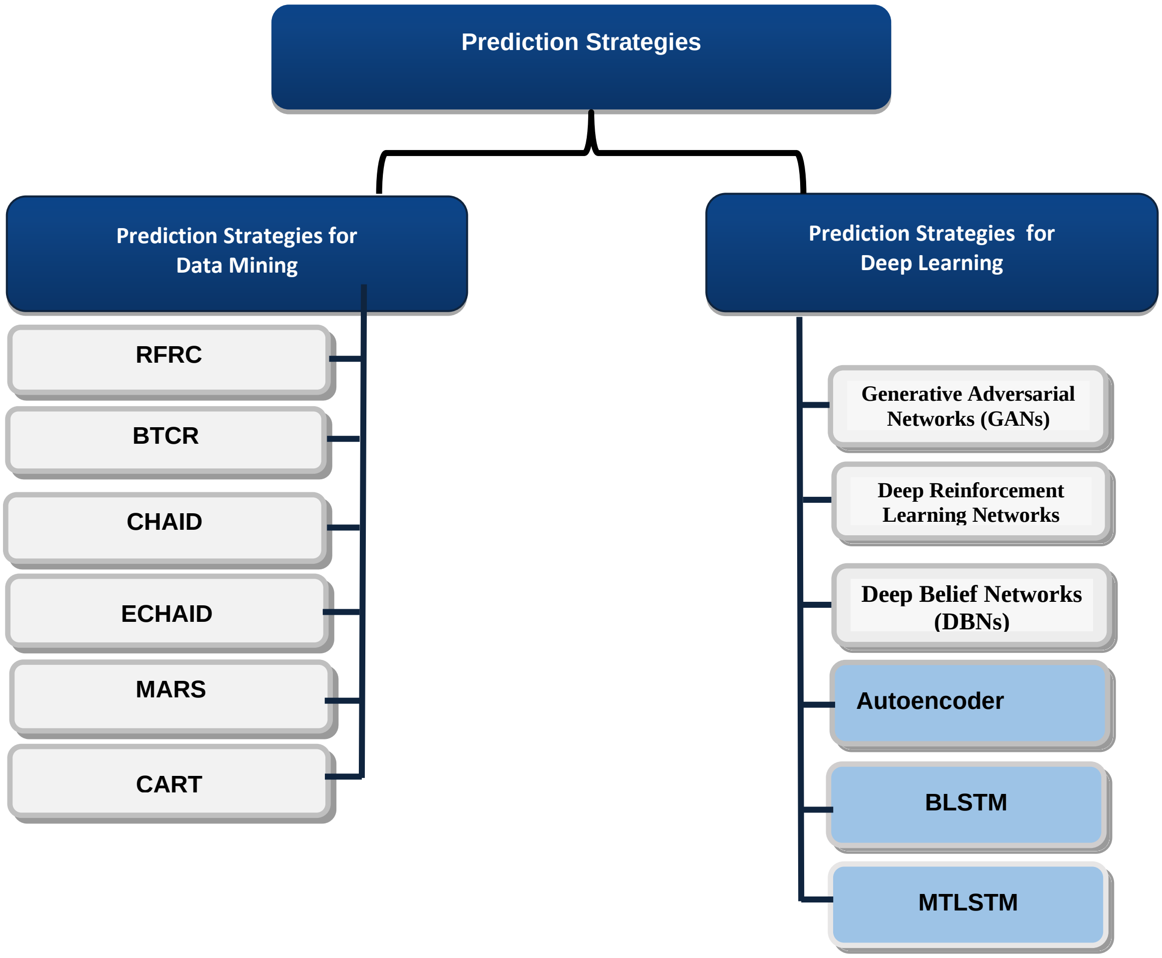


Figure (5): **Some of Prediction Strategies**

3.2.1 Autoencoder Neural Network Algorithm

Autoencoders (AE) are a type of artificial neural network that aims to learn a compressed representation of input data through a bottleneck layer, known as the latent space, and then reconstruct the output from this compressed representation. They can compress input data into a lower-dimensional representation and then reconstruct the output from this compressed representation. Main used of AE.

- feature extraction in unsupervised machine learning tasks.
- They can be learned automatically from data examples.
- Autoencoders are data-specific and may not work well with data that is significantly different from the data they were trained on.
- The decompressed outputs may be degraded compared to the original inputs.

The main steps to training an autoencoder algorithm include defining the architecture of the network, initializing the weights and biases, feeding input data into the encoder to obtain a compressed representation, reconstructing the original input data through the decoder, computing the loss function to measure the difference between the reconstructed output and the input data, and using an optimization algorithm to update the weights and biases to minimize the loss function. The algorithm learns to compress input data into a lower-dimensional representation while preserving important features of the data [24][25]

Algorithm 4: AE

Input: *Dataset*
Output: *Predicted of target*
// Initialization the Parameters

```
While not meeting the max _epoch do
    For each batch in max _iterations
        For each x(t) in the batch
            Z(t) = ((x(t), w(t)) + b(t))
            Xnew(t) = (Wi(t) * (Z(t)) + bi(t))
            f(t) = [(Wi(t)((x(t), w(t)) + b(t)) + bi(t))- x(t)]^2
        End for
    End for
End while
End AE
```

$$Z(t) = \left(\left(x(t), w(t)\right) + b(t)\right)$$
$$Xnew(t) = \left(Wi(t) * \left(Z(t)\right) + bi(t)\right)$$
$$f(t) = [\left(Wi(t)\left(\left(x(t), w(t)\right) + b(t)\right) + bi(t)\right) - x(t)]^2$$

3.2.2 Bidirectional LSTM Algorithm (BLSTM)

BLSTM shown in algorithm 5 is an extension of traditional LSTM that improves model performance on various problems. BLSTM processes input in two dimensions [21][22]

3.2.3 Multiplicative LSTM Algorithm (MTLSTM)

MLSTM appear in algorithm 6 is a hybrid network that combines the multiplicative RNN (MRNN) and LSTM's gating framework. By using connections from the intermediate state in MRNN to each gate in LSTM, it increases flexibility [23][24]

Algorithm 6: MTLSTM

Input: *Dataset*
Output: *Predicted of target*

```
//Control net complexity and input the data blocks to the allocated layers
While not meeting the max # epoch do
    For each batch in max_ iterations
        For each x(t )in the batch
            M(t) = Wm.x(t) ⊙ Wm.h(t − 1)
            h'(t) = Wh.x(t) + Wh.M(t)        //M(t): intermediate state to each gate unit; h'(t):hidden state
            i(t) = σ(Wi .x(t) + Wi.M(t))
            fg(t) = σ(Wf .x(t) + Wo.M(t))
            o(t) = σ(Wo .x(t) + Wo.M(t))
        End for
    End for
End while
End MTLSTM
```

$$M(t) = Wm.x(t) \odot Wm.h(t-1)$$
$$h'(t) = Wh.x(t) + Wh.M(t)$$
$$i(t) = \sigma\left(Wi\,.x(t) + Wi.M(t)\right)$$
$$fg(t) = \sigma\left(Wf\,.x(t) + Wo.M(t)\right)$$
$$o(t) = \sigma\left(Wo\,.x(t) + Wo.M(t)\right)$$

The size of M(t) in MTLSTM is selected to be equal to h(t), resulting in MTLSTM models being 1.25 times larger than equivalent LSTM models.

Algorithm 5: BLSTM

Input: *Dataset*

Output: *Predicted of target*

// Initialization the net Parameter

//Control net complexity and input the data blocks to the allocated layers

While *not meeting max _epoch do*

  For *each batch in max_iterations*

    For *each x(t) in the batch*

      // ***forward layer***

      $c(t) = \tanh(Wc(x(t), y(t-1)) + bc)$

      $i(t) = \sigma(Wi(x(t), y(t-1)) + bi)$

      Compute fg(t)

      cs(t)=fg(t) *c(t-1) *(i(t)*cs(t))

      $o(t) = \sigma(Wo(x(t), y(t-1)) + bo)$

      $y'(t) = o(t) * \tanh(c(t))$

      // ***Backward Layer***

      $c(t) = \tanh(Wc(x(t), y(t-1)) + bc)$

      $i(t) = \sigma(Wi(x(t), y(t-1)) + bi)$

      Compute Fg(t)

      cs(t)=fg(t) *c(t-1) *(i(t)*cs(t))

      $Co(t) = \sigma(Wo(x(t), y(t-1)) + bo)$

      $y''(t) = o(t) * \tanh(c(t))$

      // ***Activation layer***

      $y(t) = f(y'(t), y''(t))$

    End for

  End for

End while

End BLSTM

Table (4) Compare among Three Prediction Techniques

Algorithm	Advantages	Disadvantages
Autoencoder	▪ can learn complex patterns from raw data. ▪ achieve high accuracy on benchmark datasets. ▪ highly scalable for big data applications. ▪ adaptable to a wide range of applications.	▪ require large amounts of data to train effectively. ▪ computationally intensive and require powerful hardware. ▪ difficult to interpret and explain their outputs. ▪ prone to overfitting.
BSTM	▪ Solve problem sprinted into fixed sizes ▪ More accuracy compares with LSTM	▪ Used two LSTM cell. ▪ Not suitable for multi-application related of speech ▪ Increase the computation
MTLSTM	▪ The MLSTM characterizes by its being more meaningful for autoregressive density estimation. ▪ Flexible. ▪ For every input have different recurrent transition functions	Outperformed only in language modelling and related problems only

4. MULTI-STRATEGY FUSION FOR LOCALIZATION

The proposed model used to estimate the position of a sensor node according to signal strength and positions of three anchors shown in figure 6 while the complete algorithm of that model explain with all details in algorithm 7 . The Multi-Strategy Fusion for Localization model is composed of a multistage each one responsible for a specific task, these tasks integrate with each other's to achieve the main goal of it (i.e., determined the location of Sensor Nodes):

- ***Stage #1:*** **Distribution Anchors Nodes and Scan the Region** include distribution Anchors Nodes in Distribution Anchors Nodes known location of specific region; Then Scan this region to compute the RSSI for each Anchor; in addition; add these RSSI as new feature for each anchor (i.e., generated new dataset have nine features (X, Y and RSSI) for three anchors).
- **Stage #2: Finding the list of best Candidates** through two steps; (a) Compute the Fitness function as min(error), (b) Sort individual and select 100 best Candidates then pass the results into stage #3.
- **Stage #3: Build Multi Predictor based on optimization for finding the location of Sensor Nodes, the input of this stage is** best Candidates pulse new dataset generated through stage #1; after that apply in parallel four optimization techniques called (KNN, EHO, ABO and DSA) Then pass the results of each technique into the stage of evaluation measures.
- **Stage #4: Build Multi Predictor based on Deep Learning for finding the location of Sensor Nodes:** the input of this stage is the training dataset generated through split the new dataset result from stage #1 into two parts training and testing part; **after that** apply in parallel three deep learning techniques called (autoencoder, BILSTM, and MTLSTM) Then pass the results of each technique into the stage of evaluation measures.
- **Stage #5: Compute Evaluation Measures:** this stage includes two steps the first is compute the average error of each prediction or optimization technique after compute the distance between the predict target (coordinates of Sensor Nodes **(***Predict_ SN(x,y)) and actual target for testing dataset (Test_ SN(x,y) while the second step is compute the MSE and MAE among the optimization technique to determined the best technique also among the prediction techniques.*

The proposed method is based on the hypothesis that there is a strong correlation between the position of any wireless device (sensor) and the signal intensity of nearby devices. This presumption states that after training the model utilizing signal strength indicators of various neighbor anchor nodes relative to various positions of the sensor node, we can forecast the position of the sensor node.

5. RESULTS OF MULTI-STRATEGY FUSION FOR LOCALIZATION MODEL

This section presents the implementation of suggest algorithm developed to address the problems. It also explains the main parameters utilized in each algorithm. Additionally, it provides evidence for the obtained results and interprets them in a manner that is accessible to a wide range of readers. Starting by the processes of data collection phase of the model.The dataset consists of 1148 samples and nine features with two targets as explain in Table 5. The main attributes to be considered as inputs in the dataset are the positions of the mobile anchors and their related RSSI measured relatively to each sensor node. While the position of the given sensor is the target. Hence, the input vector is represented by :{{(X1, Y1), RSSI1}, {(X2, Y2), RSSI2}, {(X3, Y3), $RSSI_3$}}, where:(xi,yi) and RSSI are the position coordinates and signal strength indicator of anchor i respectively.

Our model is of type multi-output, roughly speaking, xk and yk the estimated coordinates of the unknown sensor k. k :1..K. K is the number of dataset items is equal to the total number of deployed sensors. Hence, for three mobile anchors the length of pattern vector becomes eleven features (nine inputs and two targets).

The data is collected as grid of sensors. Distributed inside the triangle that is formed by the three anchors. Four different positions for mobile anchors are considered. Then, RSSI are collected at each grid point to complete the data vector.

TABLE (5)**: Samples of Dataset**

Features									**Targets**	
Anchor Node (AN #1)			**Anchor Node (AN #2)**			**Anchor Node (AN #3)**			**Sensor Node (SN)**	
X1	**Y1**	**RSSI1**	**X2**	**Y2**	**RSSI2**	**X3**	**Y3**	**RSSI3**	**X**	**Y**
12.15	13.08	5.65	17.05	1.79	0	0	6.97	15.66	4.14	3.85
13.88	5.71	11.66	17.05	1.79	0	0	6.97	15.66	5.9	10.06
6.13	11.51	13.18	0	0	6.97	15.66	17.05	1.79	6.41	9.56
4.09	13.13	14.88	0	0	17.05	1.79	6.97	15.66	3.94	1.1
9.67	13.6	7.68	6.97	15.66	0	0	17.05	1.79	11.56	7.16
14.67	12.53	4.69	6.97	15.66	17.05	1.79	0	0	5.75	11.13
8.42	12.03	9.65	0	0	17.05	1.79	6.97	15.66	5.82	6.09
8.16	14.69	8.99	17.05	1.79	6.97	15.66	0	0	8.93	1.1
14.35	4.12	13.29	0	0	17.05	1.79	6.97	15.66	3.76	1.67
10.34	10.69	8.77	0	0	17.05	1.79	6.97	15.66	8.11	6.97
13.59	10.39	6.98	6.97	15.66	0	0	17.05	1.79	6.67	2.08
8.79	8.69	13.15	17.05	1.79	0	0	6.97	15.66	10.61	7.77
8.81	13	8.73	17.05	1.79	0	0	6.97	15.66	10.49	7.67
10.79	9.95	9.04	17.05	1.79	6.97	15.66	0	0	7.41	6.64
13.6	8.41	8.92	6.97	15.66	17.05	1.79	0	0	8.14	2.11
8.81	8.41	14.03	0	0	17.05	1.79	6.97	15.66	11.23	8.41
11.52	14.67	5.66	17.05	1.79	0	0	6.97	15.66	5.56	1.06
9.34	8.15	13.16	6.97	15.66	0	0	17.05	1.79	10.28	8.22

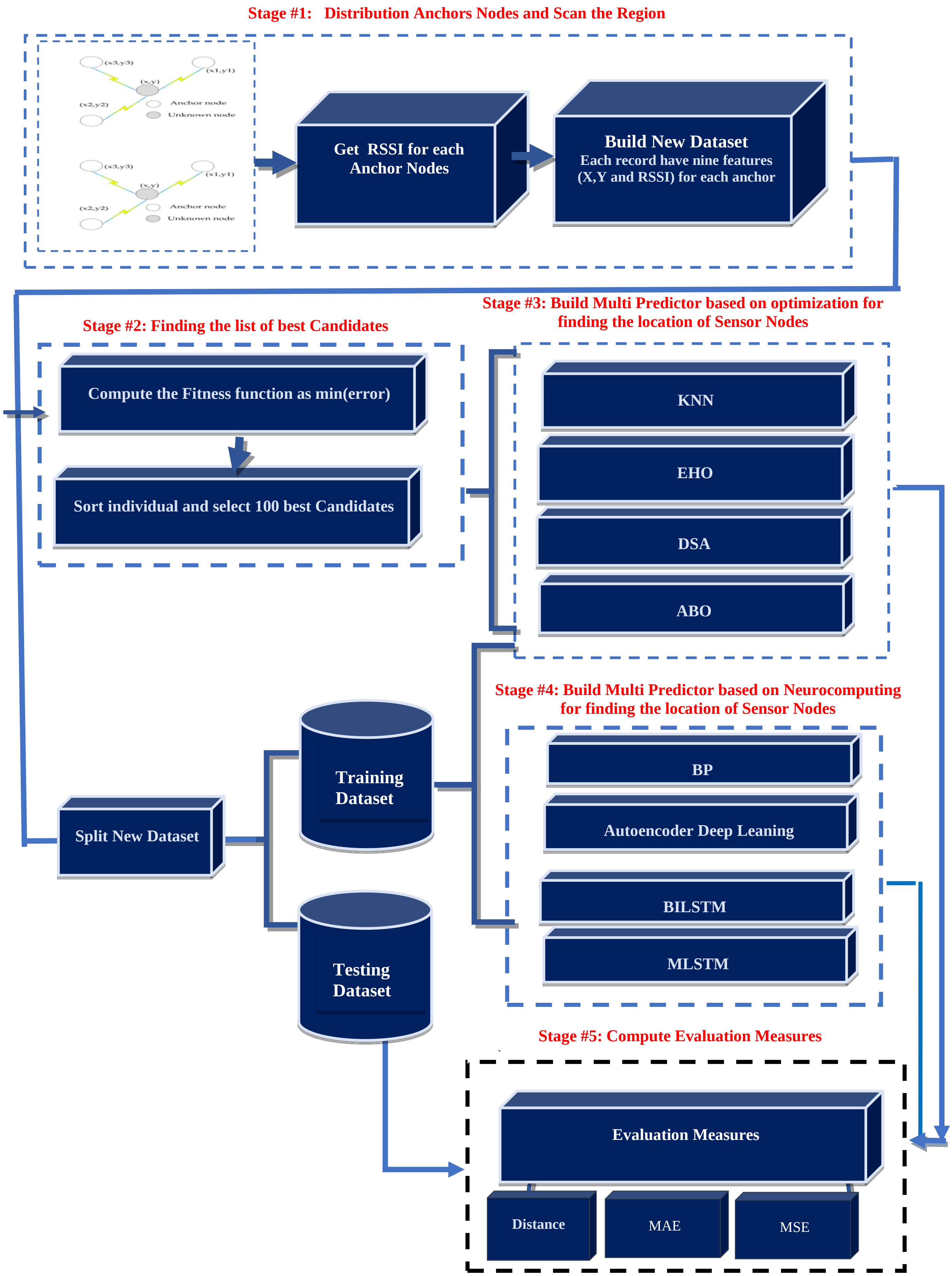


Figure (6): Model of Multi-Strategy Fusion for Localization

Algorithm: Enhancing Localization in Wireless Sensor Networks

Input: *Location of three deployed anchors*

Output: *Predicted the location of sensors*

// Setting the environment (area, objective function and RSSI)
Let NDataset=[]

1: ***Scan*** *geographical area A contain set of N unknown location of Sensor Node; with set of M =3 known location of Anchor Nodes.*

2: ***Set*** *objective function as the min (error)*

3: ***Get*** *RSSI for each anchor and add it as new features assotated with anchor' location*

4: ***Append*** *nine features (Coordinates pulse RSSI for three anchors)to NDataset*

// Find the list of best Candidates

5: ***Gen =0***

6: ***For*** *each reading-row in NDataset* *// NDataset : New dataset*

7: ***For*** *each F in reading-row*

8: ***Generation*** *initial population contain 1000 individual (I-pop)*

9: ***Compute*** *weight of each individual through go to (2)*

10: ***End for***

11: ***Sort*** *individual based on wights*

12: ***Save*** *the best possible 100 candidate solution in list called best_ candidates*

End for

// Build Multi Predictor based on optimization for finding the location of Sensor Nodes

13: ***For*** *each I in I_POP* *//I: individual, I-POP: Initial population*

14: ***For*** *each C in List of best_ candidates* *// C: List of best_ candidates*

15: ***For*** *each F in reading-row* *// F: features*

16: ***Call KNN*** ***Go to step 31*** *// KNN: K-Nearest Neighbor(KNN) Algorithm*

17: ***Call ABO*** ***Go*** *to step 31* *// ABO: African Buffalo Optimization Algorithm*

18: ***Call DSA*** ***Go*** *to step 31* *// DSA: Deterministic Selection Optimization Algorithm*

19: ***Call EHO*** ***Go*** *to step 31* *// EHO: Elephant Herding Optimization Algorithm (EHO)*

20: ***End for***

21: ***End For***

End For

// Build Multi Predictors based on Neurocomputing for finding the location of Sensor Nodes

22: ***Split NDataset into parts called training part and testing part***

23: ***For*** *each training part*

24: ***For*** *each F in* ***NDataset***

25: **Call** ***Autoencoder*** ***Go*** *to step 31*

26: **Call** ***BiLSTM*** ***Go*** *to step 31*

27: **Call** ***MLSTM*** ***Go*** *to step 31*

28: **Call BP** ***Go*** *to step 31*

29: ***End for***

30: ***End for***

31: *Test the conditions for stopping*

32: ***IF*** *(generates error <Emax) or (G<G_{max})* *//Emax: Maximum Error; G_{max}: Max number of generation*

33: *Go to step 38*

34: *Else*

35: *Go to(step 13 for Optimization models) or (step 23 for prediction model)*

36: ***End IF***

37: ***End for***

38: ***End for***

// Performance Evaluation.

39: ***For*** *each r in NDataset*

40: ***For*** *each F in reading-row*

41: ***Call*** MSE (*Test_SN(x,y) and Predict_ SN(x,y))*

42: ***Call*** MAE (*Test_SN(x,y) and Predict_ SN(x,y))*

43: ***Call*** *Distance (Test_ SN(x,y) and Predict_ SN(x,y))*

44: ***End for***

45: ***End for***

46: ***End Enhancing Localization in Wireless Sensor Networks***

5.1 Results of Finding the Location of Sensor Nodes based on Optimization Techniques

The main parameters used in each optimization algorithms:

A. EHO algorithm: num_iterations=100, num_eles=10(i.e., The number of elephants/herds in the algorithm in this work chose it from the best 100 candidates), exploration_rate=range (0.5 , 1.0)(i.e., A higher exploration rate promotes broader exploration of the solution space, while a lower rate emphasizes exploitation of local optima), herding_strength= range (1.0 , 10.0)(i.e., A higher value indicates a stronger influence of the leading elephant on the other elephants, encouraging them to follow a similar direction. A lower value allows for more independent movement of elephants, which can help explore diverse regions of the search space. In this work used =8.0), crossover_probability=0.9 while, mutation_probability=0.1.

B. ***ABO algorithm:*** num_iterations=100, num_buffalos=10, lb=0 (i.e., lower_bounder) and ub=20 (upper_bounder), distance_metric: euclidean_function, ABO-Operations: updating the solution and fitness.

C. DSA : num_sublist= (size of dataset/5); buffer=range(1, num_sublist), start_index= 0/ actual starting index, end_index= the length of the list / the actual ending index of the list, pivot_strategy:this parameter determines the strategy for selecting the pivot element during the partitioning step. recursive_limit:this parameter specifies the maximum recursion depth allowed during the recursive calls.

D. KNN algorithm: n_neighbors is Number of neighbors select from the best 100 candidates. Weights: Weight function used in prediction.in this work all points in each neighborhood are weighted equally; p: Power parameter for the Minkowski distance metric. When p=1, it is equivalent to using the Manhattan distance. When p=2, it is equivalent to using the Euclidean distance. Metric: Distance metric used between points is Euclidean. The results of prediction of each algorithm shown in Table 6 While table 7 shown the error of each corrodents pulse the distance between the actual and predicted location of sensor nodes ***(SN(x,y)).***

TABLE (6): RESULTS OF EACH Optimization Techniques

ABO				DSA				EHO				KNN			
Actual X	**Predicted X**	**Actual Y**	**Predicted Y**	**Actual X**	**Predicted X**	**Actual Y**	**Predicted Y**	**Actual X**	**Predicted X**	**Actual Y**	**Predicted Y**	**Actual X**	**Predicted X**	**Actual Y**	**Predicted Y**
4.14	4.44	3.85	0.40	4.14	2.10	3.85	5.28	4.14	10.32	3.85	9.17	2.48	3.07	9.96	1.78
5.9	4.49	10.06	4.77	5.9	5.03	10.06	14.13	5.9	6.79	10.06	8.94	1.26	1.95	11.92	2.69
6.41	2.18	9.56	8.92	6.41	7.20	9.56	10.26	6.41	11.24	9.56	0.22	12.4	2.15	3.22	2.02
3.94	4.00	1.1	2.77	3.94	3.37	1.1	2.82	3.94	4.53	1.1	11.94	5.36	2.80	5.98	3.14
11.56	13.16	7.16	5.91	11.56	11.74	7.16	5.17	11.56	7.97	7.16	10.50	6.78	1.65	5.99	1.57
5.75	3.42	11.13	8.78	5.75	10.84	11.13	9.13	5.75	16.67	11.13	4.65	6.71	1.92	3.64	1.80
5.82	2.83	6.09	5.68	5.82	7.86	6.09	7.00	5.82	14.38	6.09	8.99	9.16	1.02	4.65	4.68
8.93	9.23	1.1	2.24	8.93	8.53	1.1	1.41	8.93	7.64	1.1	2.08	10.49	3.54	7.67	1.99
3.76	0.67	1.67	2.63	3.76	7.51	1.67	0.16	3.76	6.05	1.67	0.72	10.51	1.55	1.42	2.75
8.11	6.75	6.97	3.68	8.11	7.62	6.97	8.55	8.11	6.29	6.97	2.69	5.74	3.16	5.32	3.30
6.67	6.66	2.08	6.88	6.67	7.28	2.08	3.08	6.67	0.18	2.08	4.55	8.7	2.31	6.01	2.88
10.61	13.30	7.77	2.37	10.61	7.58	7.77	7.80	10.61	8.50	7.77	2.80	9.25	4.07	8.23	0.41
10.49	9.33	7.67	6.22	10.49	13.02	7.67	10.12	10.49	9.30	7.67	7.28	10.85	3.35	10.35	2.00
7.41	8.92	6.64	6.21	7.41	12.29	6.64	8.41	7.41	13.47	6.64	7.72	10.16	3.60	6.7	3.91
8.14	10.03	2.11	1.53	8.14	8.27	2.11	7.94	8.14	8.57	2.11	5.74	14.34	0.08	2.42	5.98
11.23	14.82	8.41	6.29	11.23	12.75	8.41	8.55	11.23	10.78	8.41	8.89	4.01	4.04	10.77	0.91
5.56	5.80	1.06	5.65	5.56	6.41	1.06	1.83	5.56	2.49	1.06	2.35	8.74	2.16	11.04	2.08
10.28	8.11	8.22	12.35	10.28	9.40	8.22	9.32	10.28	2.84	8.22	15.19	11.38	1.05	11.83	4.71
5.86	3.67	3.41	5.19	5.86	4.66	3.41	4.85	5.86	13.28	3.41	7.77	12.5	1.06	12.92	4.86

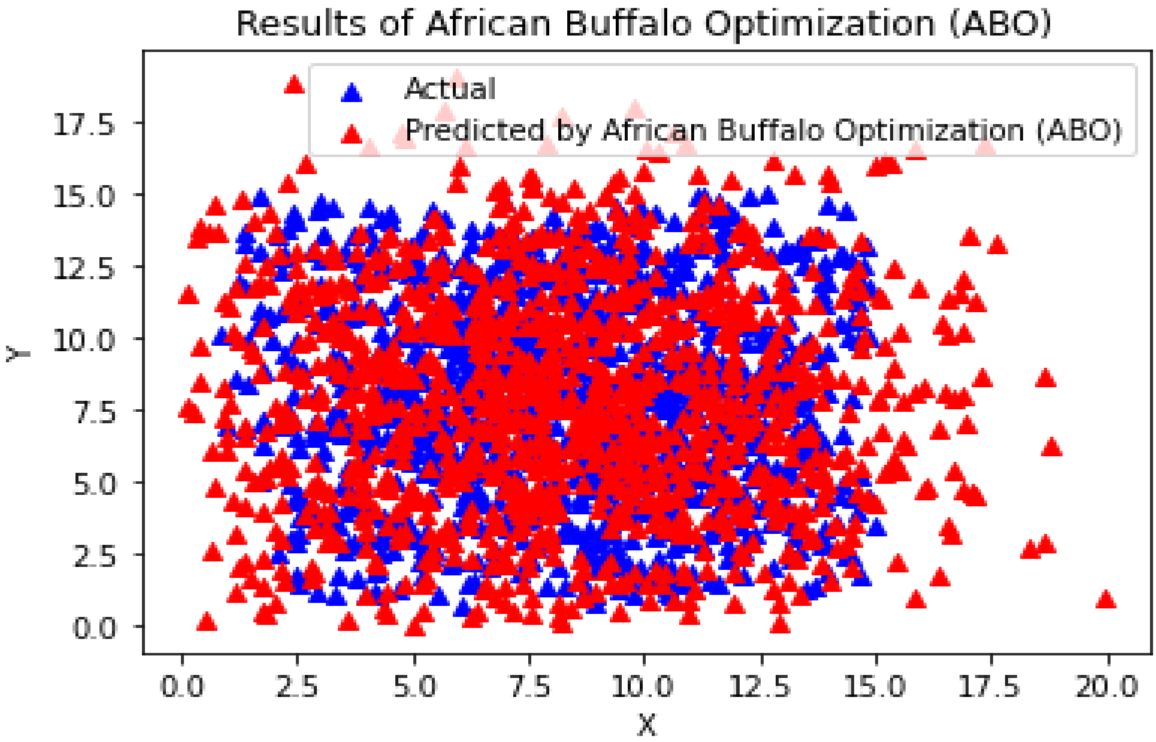
The results of the Model of Multi-Strategy Fusion for Localization related to each optimization technique shown in Figure 7 that show distribution of original and predict location of all senor nodes.

While Figure 8 shown the distribution of distance based on Euclidean (i.e., The relationship between the distance and the frequency). In Figure 8 show the high frequency of sample lay in left side and the max distance not reach to 10 also; in DSA most high frequency of sample lay in left side and the max distance not reach to 10 and it appear as more suitable from ABO while distance reach to 15 in EHO while the distance that represent the different between the actual and predict values of SN based on Euclidean reach to 17.5.

Compare among the performance of the four optimization technique show in Figure 9 and Table 8 that shown DSA consider as pragmatic tool to determine the best (Pvoit) location for each ***(SN(x,y)).*** While; Figure 10 show the analysis of optimization techniques based on distance.

TABLE (7): Error and Distance for each Optimization Techniques

ABO			DSA			EHO			KNN		
Error of ABO_X	Error of ABO of Y	Distance	Error of DSA_X	Error of DSA_Y	Distance	Error of EHO- X	Error of EHO- Y	Distance	Error of KNN- X	Error of KNN- Y	Distance
0.30	6.21	3.46	2.04	1.43	2.49	6.18	5.32	8.15	0.59	8.18	8.20
1.41	0.50	5.47	0.87	4.07	4.16	0.89	1.12	1.42	0.69	9.23	9.25
4.23	8.46	4.27	0.79	0.70	1.05	4.83	9.34	10.51	10.25	1.20	10.32
0.06	6.06	1.67	0.57	1.72	1.81	0.59	10.84	10.85	2.56	2.84	3.82
1.60	3.97	2.03	0.18	1.99	1.99	3.59	3.34	4.90	5.13	4.42	6.76
2.33	5.04	3.31	5.09	2.00	5.47	10.92	6.48	12.69	4.79	1.84	5.13
2.99	4.99	3.01	2.04	0.91	2.23	8.56	2.90	9.04	8.14	0.03	8.13
0.30	0.57	1.18	0.40	0.31	0.50	1.29	0.98	1.62	6.95	5.68	8.97
3.09	5.30	3.23	3.75	1.51	4.03	2.29	0.95	2.48	8.96	1.33	9.05
1.36	4.89	3.56	0.49	1.58	1.65	1.82	4.28	4.64	2.58	2.02	3.27
0.01	5.69	4.80	0.61	1.00	1.17	6.49	2.47	6.94	6.39	3.13	7.11
2.69	0.10	6.03	3.03	0.03	3.02	2.11	4.97	5.39	5.18	7.82	9.37
1.16	1.03	1.85	2.53	2.45	3.52	1.19	0.39	1.24	7.50	8.35	11.21
1.51	4.53	1.57	4.88	1.77	5.19	6.06	1.08	6.15	6.56	2.79	7.125



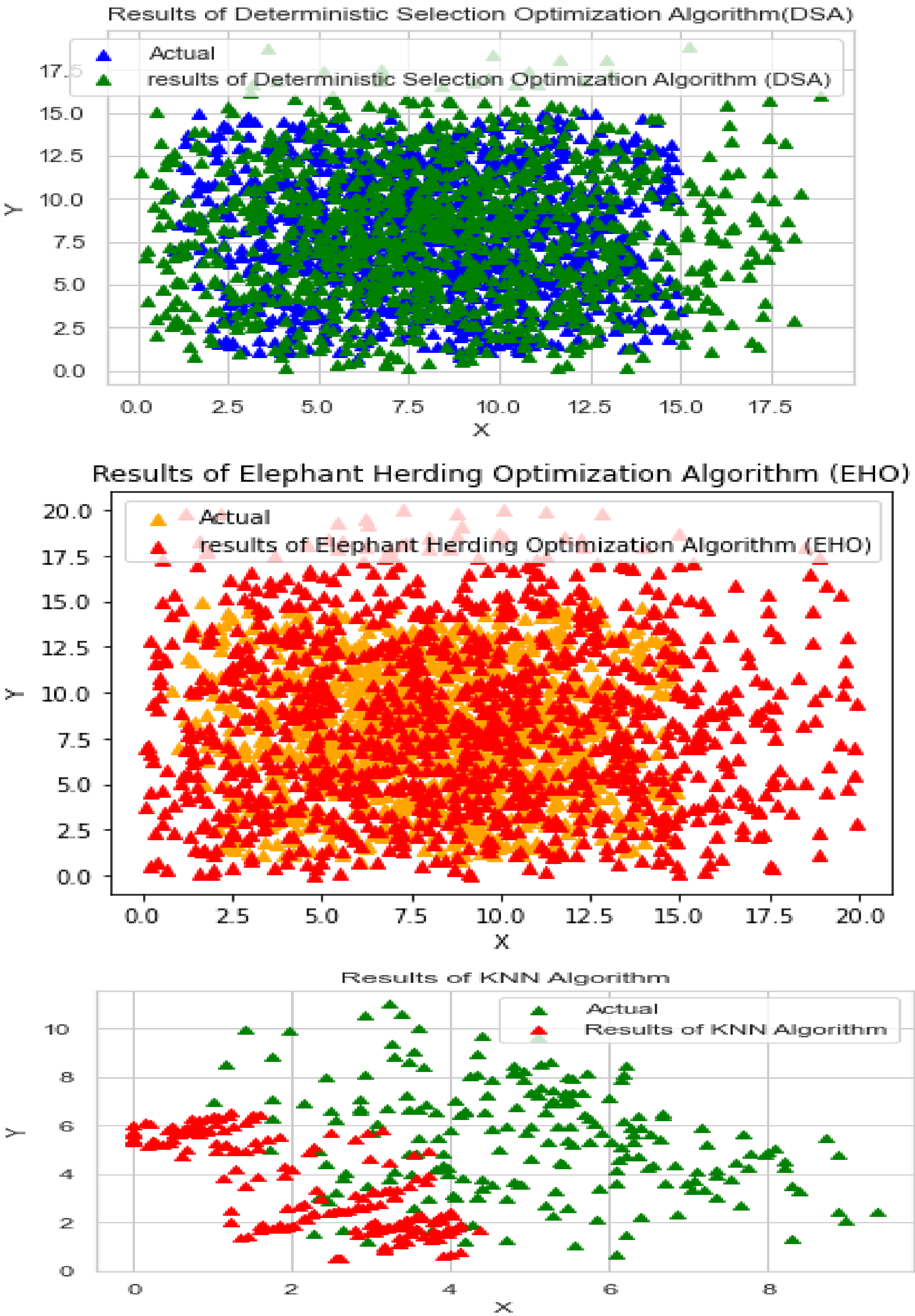


Figure (7): Distribution of Original and Predict Location of all Senor Nodes discovery through three anchors.

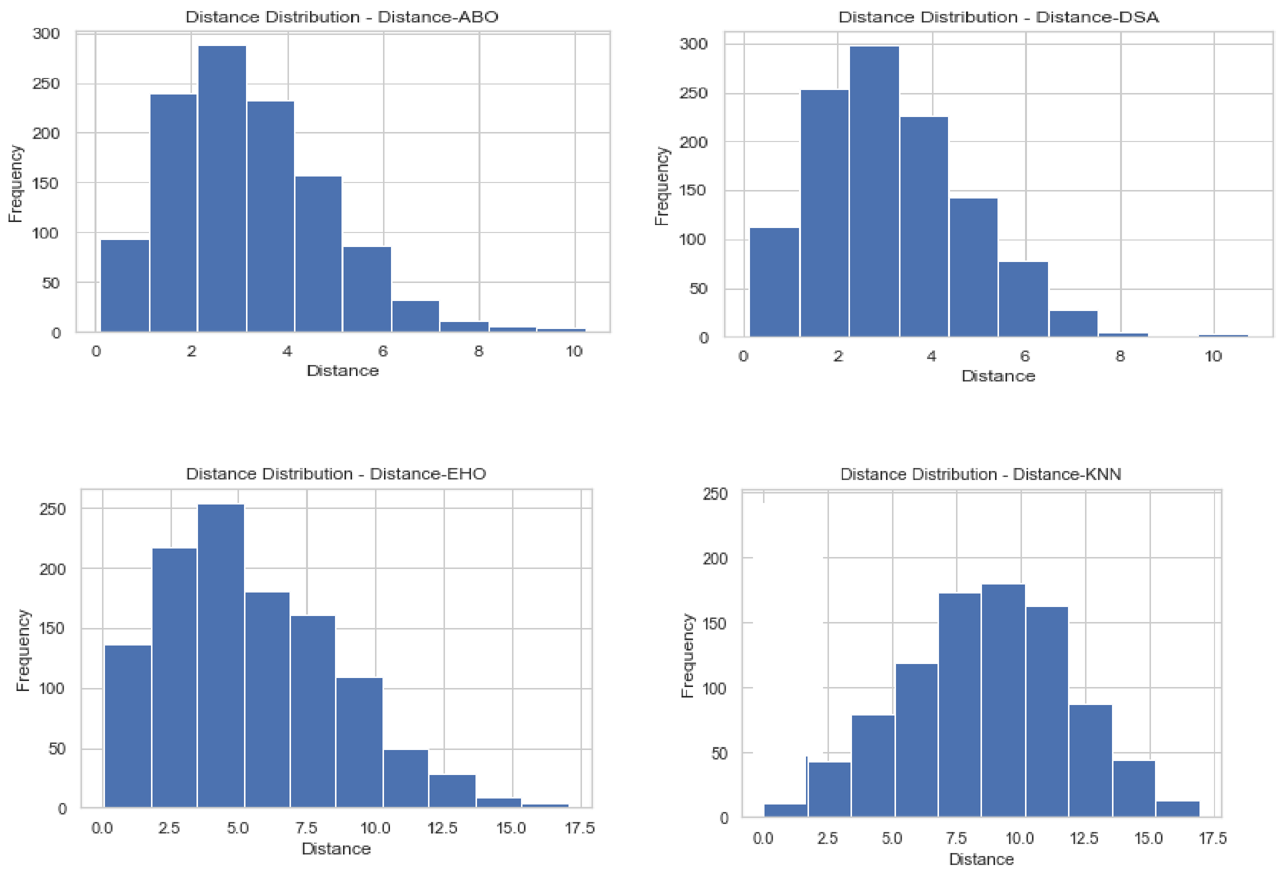


Figure (8): Distribution of distance for each optimization Techniques

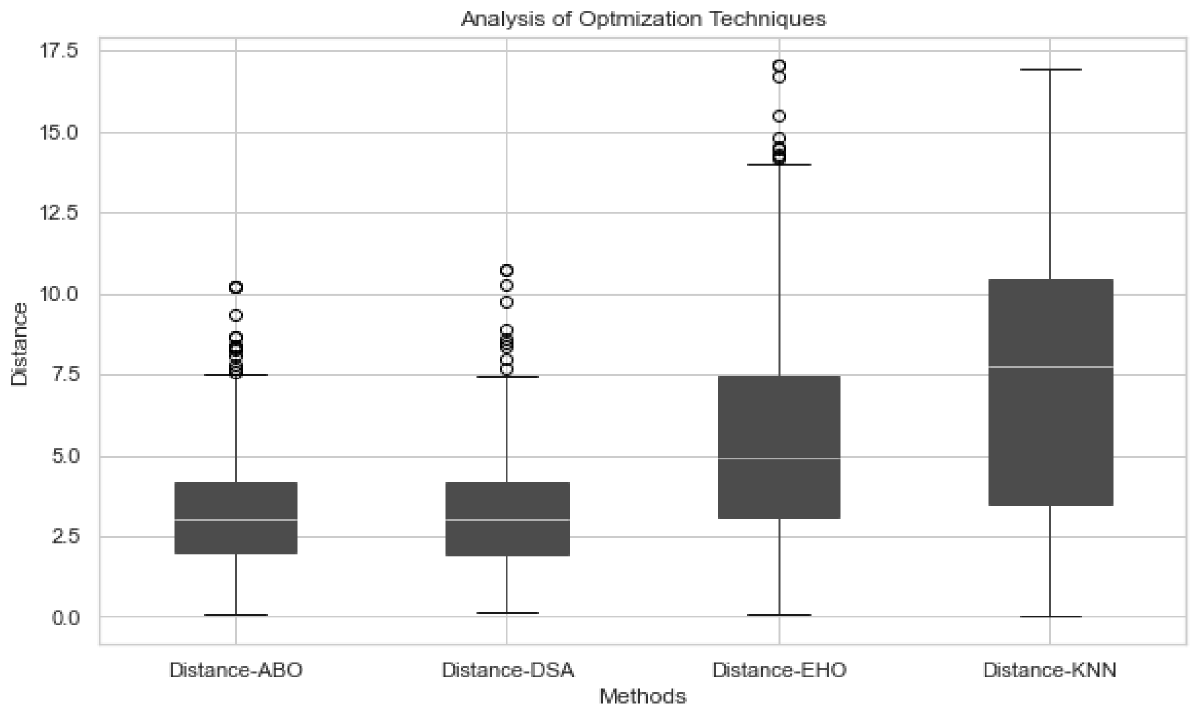


Figure (9) Compare among the performance of the four-optimization technique

TABLE (8): *Error and Average Distance of Optimization Techniques*

ABO			DSA			EHO			KNN		
MSE	MAE	Average Distance	MSE	MAE	Distance	MSE	MAE	Distance	MSE	MAE	Distance
3.18	4.04	3.175836	3.172	4.04	3.169883	9.83	6.91	5.430286	10.83	11.19	8.669502

Table 8 provide compare the performance of optimization techniques can summarization these results as follow: ABO and DSA show similar performance, with lower MSE values (3.18 and 3.172) compared to EHO and KNN. Add to that; ABO and DSA also have similar MAE values (4.04) indicating lesser average absolute errors. The distance metric for ABO and DSA is around 3.17, suggesting a moderate dissimilarity between predicted and actual values. EHO has a higher MSE (9.83) and MAE (6.91) compared to ABO and DSA, indicating less accuracy. In adding; EHO's distance value is 5.43, suggesting a larger dissimilarity between predicted and actual values. KNN has the highest MSE (10.83) and MAE (11.19), indicating the highest level of error among the four algorithms. Also; KNN's distance value is 8.67, indicating a significant dissimilarity between predicted and actual values. ABO and DSA outperform EHO and KNN in terms of accuracy and error metrics. While the best technique is DSA.

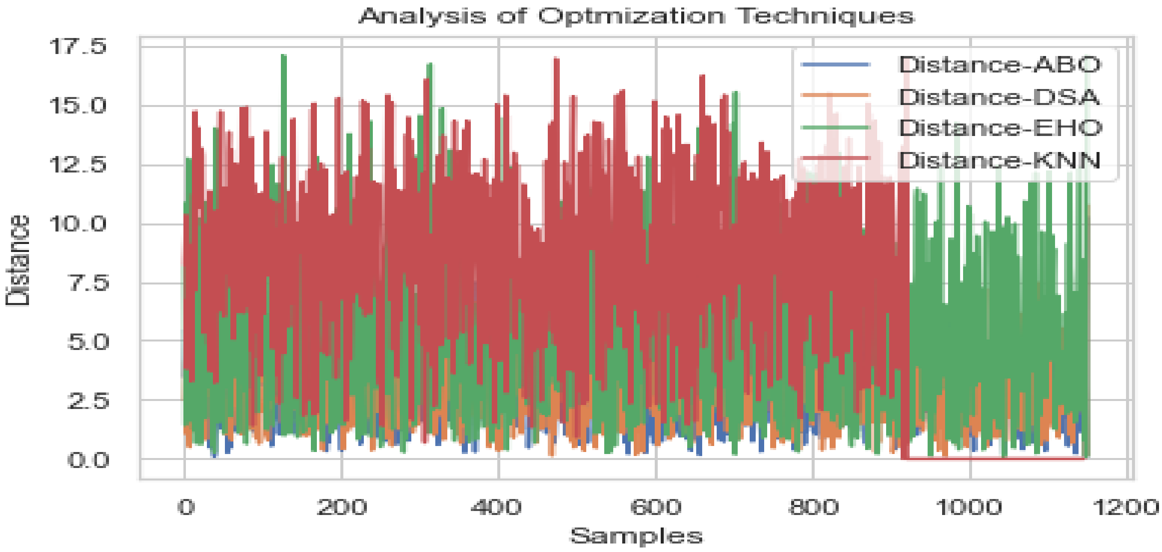


Figure (10) Analysis of Optimization Techniques based on Distance

5.2 Results of Finding the Location of Sensor Nodes based on Neurocomputing Techniques

The main parameters used in each Neurocomputing algorithms:

A. MTLSTM algorithm: activation function is ReLU, Number of LSTM Units=64, Dropout Rate=0.3, Learning_Rate=0.001, Batch_Size=32, optimizer: Adam, and epochs = 10

B. BILSTM algorithm: Number of LSTM Units=64, Dropout Rate=0.3, Learning_Rate=0.001, Batch_Size=32, optimizer: Adam, and epochs = 10.

C. Autoencoder: Number of Hidden Units is 9 units , Optimizer: used is Adam. Loss Function: is mean squared error (MSE), Batch Size: 32, Epochs: 50, learning rate= 0.001,

D. BP algorithm: Max_iterations=100, Learning_Rate=0.01, Momentum=0.9, Batch-Size=rang (32-256), Activation Functions: for hidden layers is ReLU (Rectified Linear Unit) while for the output layer is sigmoid. Number of hidden layer=2.

The results of prediction of each neurocomputing algorithm shown in Table 9 While table 10 shown the error of each corrodents pulse the distance between the actual and predicted location of sensor nodes ***(SN(x,y)).***

TABLE (9): RESULTS OF Each Neurocomputing Techniques

MLSTM				BILSTM				Autoencoder				BP			
Actual X	Predicted X	Actual Y	Predicted Y	Actual X	Predicted X	Actual Y	Predicted Y	Actual X	Predicted X	Actual Y	Predicted Y	Actual X	Predicted X	Actual Y	Predicted Y
10.64	7.13	8.91	5.40	10.64	7.10	8.91	4.95	12.01	0.61	8.28	0.68	12.01	8.29	8.28	6.81
13.35	7.47	2.61	5.33	13.35	7.26	2.61	5.04	9.70	0.54	8.72	0.58	9.70	7.62	8.72	8.39
4.14	7.12	1.87	5.65	4.14	7.26	1.87	5.01	5.18	0.60	13.68	0.72	5.18	10.87	13.68	8.56
8.76	6.93	7.22	5.50	8.76	7.08	7.22	4.92	2.14	0.49	10.88	0.54	2.14	6.52	10.88	8.08
12.54	7.25	6.86	5.34	12.54	7.14	6.86	5.05	7.38	0.63	7.77	0.69	7.38	8.69	7.77	6.86
11.82	7.20	2.33	5.46	11.82	7.17	2.33	5.00	10.62	0.56	7.23	0.63	10.62	8.35	7.23	8.36
8.97	6.95	3.12	5.53	8.97	7.08	3.12	4.93	7.70	0.58	2.78	0.67	7.70	9.26	2.78	9.03
8.97	7.21	2.96	5.21	8.97	7.03	2.96	4.87	9.15	0.54	3.47	0.59	9.15	8.06	3.47	7.36
5.73	7.14	8.17	5.26	5.73	7.02	8.17	4.96	6.00	0.56	7.50	0.69	6.00	8.31	7.50	7.43
3.46	7.08	6.65	5.62	3.46	7.22	6.65	5.03	2.09	0.46	2.57	0.50	2.09	7.51	2.57	8.48
8.99	7.24	8.66	5.22	8.99	7.07	8.66	4.88	6.21	0.61	10.44	0.68	6.21	10.24	10.44	7.68
7.34	7.07	10.47	5.56	7.34	7.16	10.47	5.00	8.78	0.62	7.38	0.68	8.78	8.46	7.38	6.12
9.78	7.17	1.10	5.61	9.78	7.28	1.10	5.06	7.54	0.54	13.45	0.63	7.54	8.15	13.45	9.26
4.39	7.16	3.76	5.40	4.39	7.13	3.76	4.97	4.21	0.63	7.65	0.69	4.21	9.07	7.65	7.18
13.18	7.15	3.54	5.61	13.18	7.25	3.54	5.01	4.65	0.50	7.23	0.53	4.65	8.02	7.23	8.31
11.57	7.28	4.99	5.19	11.57	7.06	4.99	4.93	6.15	0.49	9.29	0.51	6.15	7.66	9.29	8.32
10.16	7.11	1.61	5.59	10.16	7.20	1.61	5.03	11.68	0.52	9.16	0.58	11.68	7.71	9.16	7.13
5.78	7.30	10.74	5.38	5.78	7.18	10.74	5.04	3.99	0.61	8.93	0.76	3.99	9.80	8.93	8.32

The results of the Model of Multi-Strategy Fusion for Localization related to each Neurocomputing technique and the distribution of distance based on Euclidean (i.e., The relationship between the distance and the frequency) shown in Figuer 11. In that Figure the three techniques (BP, MTLSTM and BILSTM) the frequency of samples lay in left side and the max distance not reach to 10 also Back probagation appear as more suitable from other while distance reach to 18 in autoencoder, in general the distance represent the different between the actual and predict values of ***(SN(x,y)).*** based on Euclidean.

Compare among the performance of the four Neurocomputing technique show in Figure 12 and Table 11 that shown Bp consider as good tool to determine the location for each ***(SN(x,y)).*** While; Figure 103 show the analysis of neurocomputing techniques based on distance.

TABLE (10): Error and Distance for each Optimization Techniques

MLSTM			BILSTM			Autoencoder			BP		
Error of MTLSTM_ X	Error of MTLSTM of Y	Distance	Error of BILSTM_ X	Error of BILSTM_ Y	Distance	Error of Auto- X	Error of Auto-Y	Distance	Error of BP- X	Error of BP- Y	Distance
-3.51	-3.51	4.96	-3.54	-3.54	5.31	-11.40	-7.60	13.70	-3.72	-1.47	4.00
-5.88	2.72	6.48	-6.09	-6.09	6.56	-9.16	-8.14	12.25	-2.08	-0.33	2.10
2.98	3.78	4.81	3.12	3.12	4.43	-4.58	-12.96	13.75	5.69	-5.12	7.66
-1.83	-1.72	2.51	-1.68	-1.68	2.85	-1.65	-10.34	10.47	4.38	-2.80	5.20
-5.29	-1.52	5.50	-5.40	-5.40	5.70	-6.75	-7.08	9.79	1.31	-0.91	1.59
-4.62	3.13	5.58	-4.65	-4.65	5.36	-10.06	-6.60	12.03	-2.27	1.13	2.53
-2.02	2.41	3.14	-1.89	-1.89	2.62	-7.12	-2.11	7.42	1.56	6.25	6.44

-1.76	2.25	2.86	-1.94	-1.94	2.72	-8.61	-2.88	9.08	-1.09	3.89	4.04
1.41	-2.91	3.23	1.29	1.29	3.46	-5.44	-6.81	8.72	2.31	-0.07	2.31
3.62	-1.03	3.77	3.76	3.76	4.09	-1.63	-2.07	2.64	5.42	5.91	8.02
-1.75	-3.44	3.86	-1.92	-1.92	4.24	-5.60	-9.76	11.25	4.03	-2.76	4.89
-0.27	-4.91	4.91	-0.18	-0.18	5.48	-8.16	-6.70	10.56	-0.32	-1.26	1.30
-2.61	4.51	5.21	-2.50	-2.50	4.69	-7.00	-12.82	14.61	0.61	-4.19	4.23
2.77	1.64	3.22	2.74	2.74	2.99	-3.58	-6.96	7.83	4.86	-0.47	4.88
-6.03	2.07	6.38	-5.93	-5.93	6.11	-4.15	-6.70	7.88	3.37	1.08	3.54
-4.29	0.20	4.30	-4.51	-4.51	4.51	-5.66	-8.78	10.44	1.51	-0.97	1.80
-3.05	3.98	5.01	-2.96	-2.96	4.52	-11.16	-8.58	14.07	-3.97	-2.03	4.46
1.52	-5.36	5.57	1.40	1.40	5.87	-3.38	-8.17	8.84	5.81	-0.61	5.84
-2.04	-5.52	5.89	-2.22	-2.22	6.29	-8.88	-2.30	9.17	-0.90	4.84	4.92
0.80	-6.55	6.59	0.94	0.94	7.25	-9.83	-7.66	12.46	-1.22	1.61	2.03
-1.71	-0.93	1.95	-1.93	-1.93	2.26	-7.88	-8.32	11.46	0.32	-1.25	1.29
0.98	-3.12	3.27	0.95	0.95	3.69	-6.22	-9.25	11.15	2.64	-2.75	3.82
-1.98	-0.33	2.01	-1.83	-1.83	2.05	-4.57	-1.64	4.86	3.12	4.38	5.38

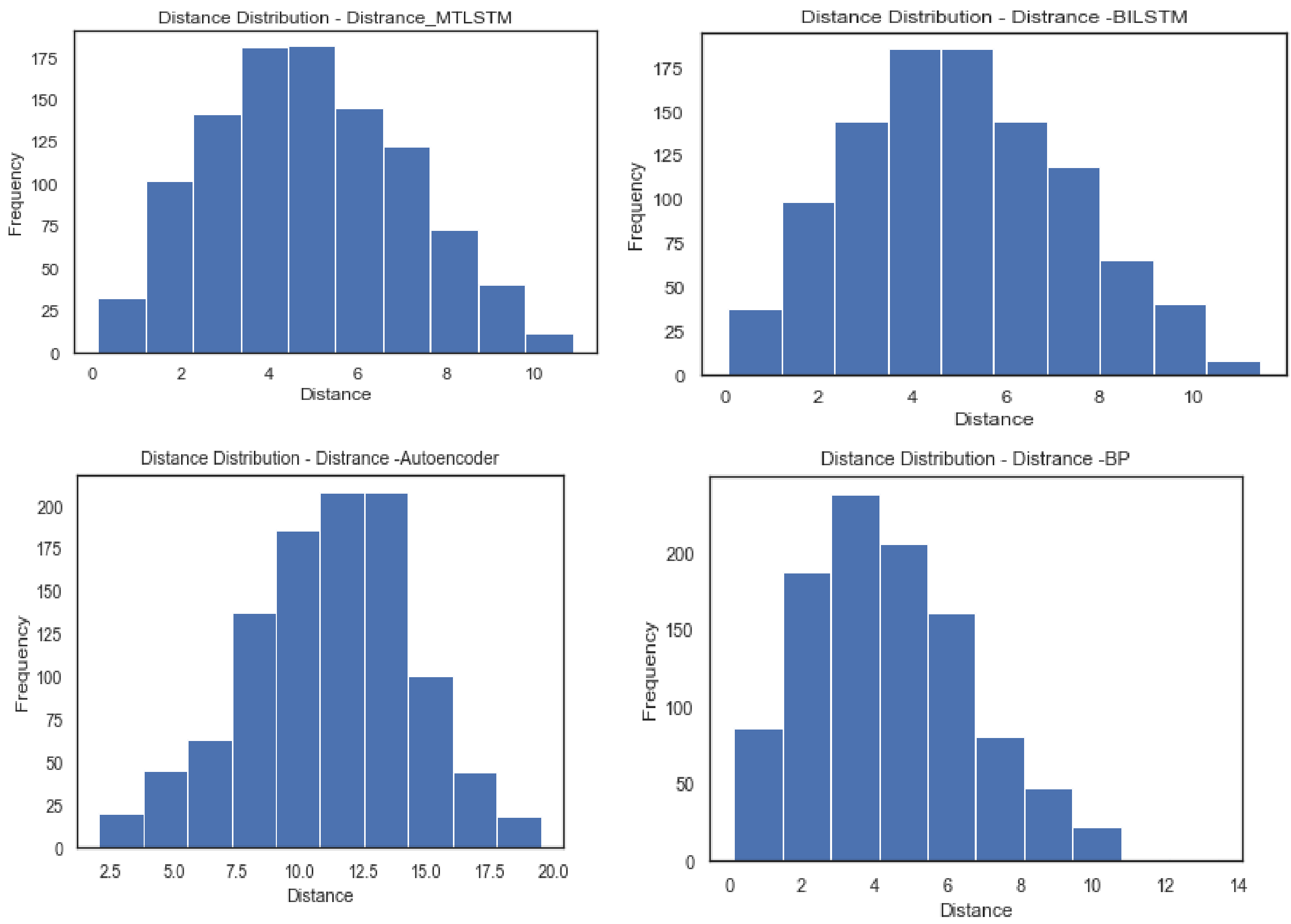


Figure (11): Distribution of Distance for each Neurocomputing Techniques

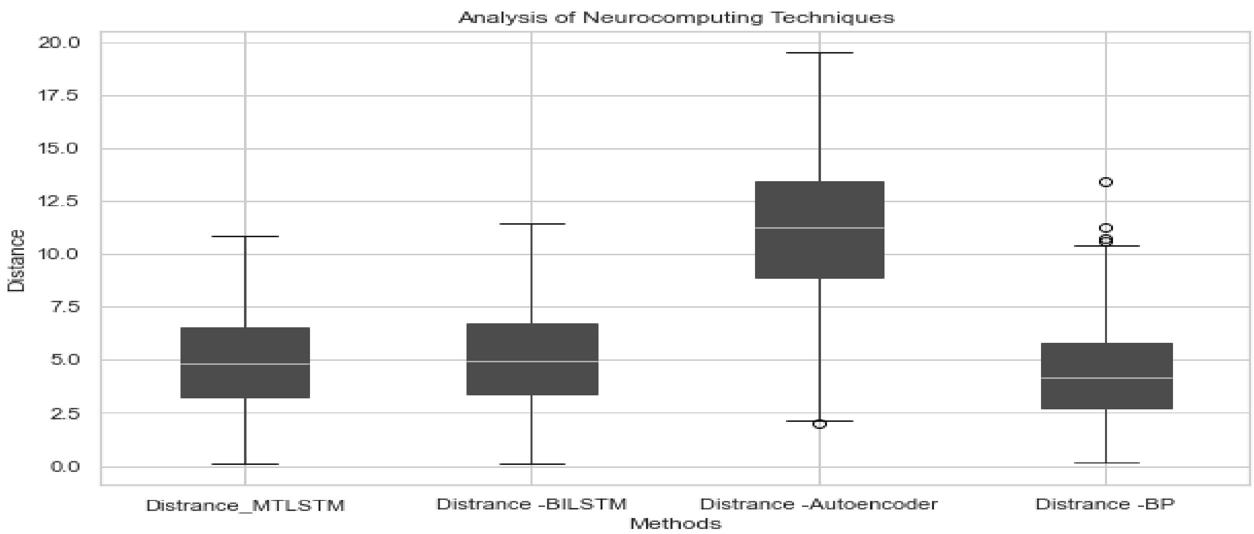


Figure (12) Compare among the performance of the four-Neurocomputing techniques

TABLE (11): *Error and Average Distance of Neurocomputing Techniques*

MTLSTM			BILSTM			Autoencoder			BP		
MSE	MAE	Average Distance	MSE	MAE	Distance	MSE	MAE	Distance	MSE	MAE	Distance
0.64	2.57	4.87	0.75	0.75	5.04	7.61	7.27531	11.07	0.41	0.27	4.39

Table 11 provide compare the performance of neurocomputing techniques can summarization these results as follow: MTLSTM and BP achieved the lowest (MSE and MAE) values, indicating better prediction accuracy compared to BILSTM and Autoencoder. Autoencoder exhibited the highest MSE and MAE, indicating larger prediction errors compared to other techniques. Average distance measurements showed that Autoencoder had the largest differences between predicted and actual values. MTLSTM and BILSTM had slightly higher average distances compared to BP but were still lower than Autoencoder. As a results, BP emerged as the top performers, demonstrating better accuracy and smaller errors compared to MTLSTM, BILSTM and Autoencoder.

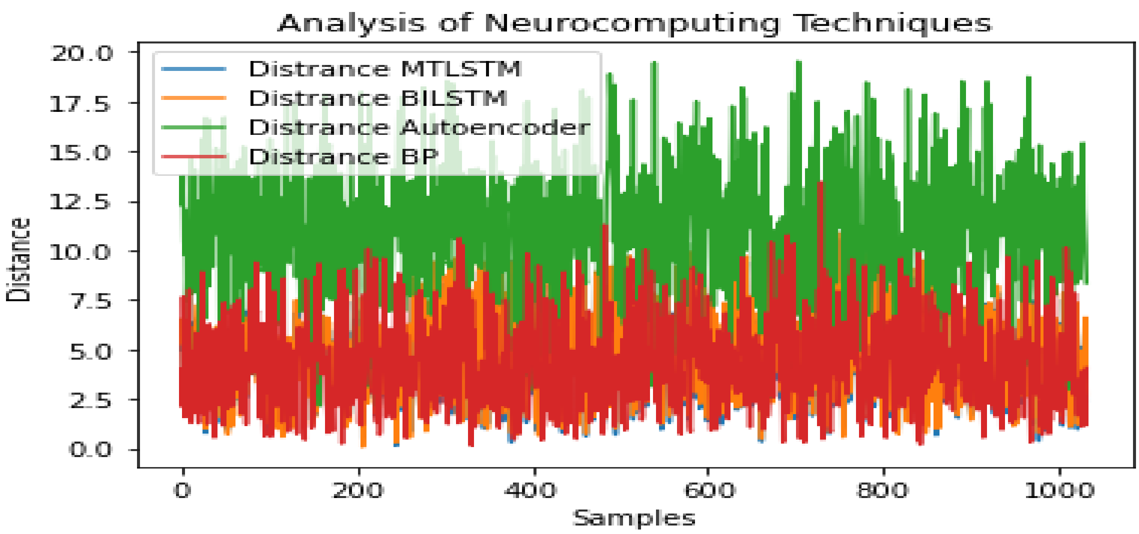


Figure (13) Analysis of Optimization Techniques based on Distance

6. CONCLUSION

Localization in wireless sensor networks (WSNs) is crucial for various applications that rely on spatial information. It can be classified into range-based and range-free techniques, each utilizing different measurements and information to estimate node positions. This paper present the Model called Multi-Strategy Fusion for Localization integrates optimization techniques (ABO, DSA, EHO, and KNN) and neurocomputing techniques (BP, MTLSTM, BILSTM, and Autoencoder) to improve localization accuracy in WSNs.

This model integrates different stages and techniques to determine the location of sensor nodes accurately. Through distributing anchor nodes, scanning the region, and get RSSI values. Then finds the best candidates based on a fitness function and selects the top 100 candidates for next processing step. After that, utilizes optimization techniques (KNN, EHO, ABO, and DSA) to predict the location of sensor nodes, with ABO and DSA showing similar and better performance compared to EHO and KNN. On other hand employs neurocomputing techniques (autoencoder, BILSTM, and MTLSTM and BP) for prediction, with BP (showcasing the best accuracy and smaller errors compared to the other techniques.

The experiments results shown; ABO and DSA optimization techniques show similar performance, with lower Mean Squared Error (MSE) values compared to EHO and KNN. ABO and DSA also have similar Mean Absolute Error (MAE) values, indicating lesser average absolute errors. BP emerges as the top performer among the neurocomputing techniques, demonstrating better accuracy with lower MSE and MAE values compared to MTLSTM, BILSTM, and Autoencoder.

As results; the Multi-Strategy Fusion for Localization model is important and considered a pragmatic tool for future use to: (a) Improved Localization Accuracy: it provides improved localization accuracy compared to using a single technique. By integrating multiple localization techniques, the model can leverage the strengths of each technique and compensate for their limitations. (b) Adaptability to Different Scenarios: The model's ability to integrate range-based and range-free localization techniques makes it highly adaptable to different scenarios. This adaptability enhances the model's practicality and usability in real-world applications.

The Multi-Strategy Fusion for Localization model can be applied in various applications:(a) Environmental Monitoring such as tracking air quality, temperature variations, or pollutant levels, through accurate localization of sensor nodes is crucial. (b) Healthcare and Elderly Care: the model can be applied to accurately locate medical devices, wearable sensors, or patients themselves. This enables remote monitoring, tracking patient movements, and providing timely medical assistance. (c) Smart Cities and Infrastructure: accurate localization of sensors and devices is necessary for various services like smart parking, waste management, or energy optimization. (d) Disaster Response and Emergency Management: During disaster response and emergency management situations, accurate localization of sensor nodes and first responders is critical. The model can be utilized to precisely determine the location of deployed sensors, emergency personnel, or affected individuals, enabling effective coordination and response efforts.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

None

REFERENCES

[1] K. M. Saifuddin, and G. D. Devanagavi, "DV-Hop Propagation Based Localization in Wireless Sensor Networks." pp. 1-7.

[2] S. Balachandran Nair Premakumari, P. Mohan, and K. Subramanian, “An Enhanced Localization Approach for Energy Conservation in Wireless Sensor Network with Q Deep Learning Algorithm,” Symmetry, vol. 14, no. 12, pp. 2515, 2022.

[3] S. J. Bhat, and S. KV, “A localization and deployment model for wireless sensor networks using arithmetic optimization algorithm,” Peer-to-Peer Networking and Applications, vol. 15, no. 3, pp. 1473-1485, 2022.

[4] W. Ding, Q. Zhong, Y. Wang, C. Guan, and B. Fang, “Target Localization in Wireless Sensor Networks Based on Received Signal Strength and Convex Relaxation,” Sensors, vol. 22, no. 3, pp. 733, 2022.

[5] A. Nasipuri, and K. Li, "A directionality based location discovery scheme for wireless sensor networks." pp. 105-111.

[6] G. Han, C. Zhang, J. Lloret, L. Shu, and J. J. Rodrigues, “A mobile anchor assisted localization algorithm based on regular hexagon in wireless sensor networks,” The Scientific World Journal, vol. 2014, 2014.

[7] L. Yujun, and C. Meng, “A trilateral centroid localization and modification algorithm for wireless sensor network,” Lecture Notes in Electrical Engineering, vol. 355, pp. 96-97, 2015.
[8] R. Valli, A. Sundhar, V. Vignesh, and S. Kotari, “Simulation and Analysis of Rssi Based Trilateration Algorithm for Localization in Contiki-Os,” ICTACT Journal on Communication Technology, vol. 7, no. 3, 2016.
[9] R. Y. Jyoti, and N. Singh, “Localization in WSN using modified trilateration based on fuzzy optimization,” International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 7, 2013.
[10] H. Agarwal, A. Dwivedi, and A. Kaur, "An improved centroid DV hop based algorithm." pp. 72-75.
[11] X. Yang, W. -A. Zhang, L. Yu and K. Xing, "Multi-Rate Distributed Fusion Estimation for Sensor Network-Based Target Tracking," in IEEE Sensors Journal, vol. 16, no. 5, pp. 1233-1242, March1, 2016, doi: 10.1109/JSEN.2015.2497464.
[12] L. Karim, N. Nasser, and T. El Salti, "RELMA: a range free localization approach using mobile anchor node for wireless sensor networks." pp. 1-5.
[13] Kaur Amanpreet, Padam Kumar and Govind P. Gupta, "A weighted centroid localization algorithm for randomly deployed wireless sensor networks", Journal of King Saud University-Computer and Information Sciences 31, no. 1, pp. 82-91, 2019.
[14] Saber Amri, Fekher Khelifi, Abbas Bradai, Abderrezak Rachedi, Med Lassaad Kaddachi and Mohamed Atri, "A new fuzzy logic based node localization mechanism for wireless sensor networks", Future Generation Computer Systems, vol. 93, pp. 799-813, 2019.
[15] Hua Wang, Yingyou Wen, Yao Lu, Dazhe Zhao and Chunlin Ji, "Secure localization algorithms in wireless sensor networks: a review", In Advances in Computer Communication and Computational Sciences, pp. 543-553, 2019.
[16] Singh Santar Pal and Subhash Chander Sharma, "Implementation of a PSO based improved localization algorithm for wireless sensor networks", IETE Journal of Research 65, no. 4, pp. 502-514, 2019.
[17] Y. Duan, C. Liu, S. Li, X. Guo, and C. Yang, "Gradient-based elephant herding optimization for cluster analysis," Applied Intelligence, vol. 52, pp. 11606-11637, 2022.
[18] M. A. Elhosseini, R. A. El Sehiemy, Y. I. Rashwan, and X. Z. Gao, "On the performance improvement of elephant herding optimization algorithm," Knowledge-Based Systems, vol. 166, pp. 58-70, 2019.
[19] P. Singh, N. K. Meena, A. Slowik, and S. K. Bishnoi, "Modified African buffalo optimization for strategic integration of battery energy storage in distribution networks," IEEE Access, vol. 8, pp. 14289-14301, 2020.
[20] C. P. Igiri, Y. Singh, and D. Bhargava, "An improved African buffalo optimization algorithm using chaotic map and chaotic-levy flight," International Journal of Engineering & Technology, vol. 7, no. 4, pp. 4570-4576, 2018.
[21] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural networks, vol. 18, no. 5-6, pp. 602-610, 2005. DOI: 10.1016/j.neunet.2005.06.042.
[22] H. Fei and F. Tan, "Bidirectional grid long short-term memory (BiGridLSTM): A method to address context-sensitivity and vanishing gradient," Algorithms, vol. 11, no. 11, pp. 172, 2018. DOI: 10.3390/a11110172.
[23] Y. Ma, Z. Zhang, D. Li, and M. Tang, "Deflated reputation using multiplicative long short-term memory neural networks," Future Generation Computer Systems, vol. 118, pp. 198-207, 2021. DOI: 10.1016/j.future.2020.12.029.
[24] B. Krause, L. Lu, I. Murray, and S. Renals, "Multiplicative LSTM for sequence modelling,"
[25] Shivakumar Kagi, Basavaraj S. Mathapati, Localization in wireless sensor network using machine learning optimal trained deep neural network by parametric analysis, Measurement: Sensors, Volume 24, 2022, 100427, https://doi.org/10.1016/j.measen.2022.100427.
[26] S. Gao, G. Wang, and X. Ma, "Deep Autoencoder Neural Networks for Gene Expression Analysis," Frontiers in Genetics, vol. 8, p. 166, 2017. [Online]. Available: https://doi.org/10.3389/fgene.2017.00166.
[27] Z. Wan, Y. Mahajan, B. W. Kang, T. J. Moore, and J. H. Cho, "A Survey on Centrality Metrics and Their Network Resilience Analysis," IEEE Access, vol. 9, pp. 104773-104819, 2021. doi: 10.1109/ACCESS.2021.3094196.
[28] A. Landherr, B. Friedl, and J. Heidemann, "A Critical Review of Centrality Measures in Social Networks," Business & Information Systems Engineering, vol. 2, no. 6, pp. 371-385, 2010. doi: 10.1007/s12599-010-0127-3.
[29] A. Saxena and S. Iyengar, "Centrality Measures in Complex Networks: A Survey," arXiv preprint arXiv:2011.07190.
[30] K. Das, S. Samanta, and M. Pal, "Study on centrality measures in social networks: a survey," Social Network Analysis and Mining, vol. 8, no. 1, pp. 1-23, 2018. doi: 10.1007/s13278-018-0493-2.
[31] Z. Wan, Y. Mahajan, B. W. Kang, T. J. Moore, and J. H. Cho, "A Survey on Centrality Metrics and Their Network Resilience Analysis," IEEE Access, vol. 9, pp. 104773-104819, 2021. doi: 10.1109/ACCESS.2021.3094196.

[32] B. Cai, L. Zeng, Y. Wang, H. Li, and Y. Hu, "Community detection method based on node density, degree centrality, and k-means clustering in complex network," Entropy, vol. 21, no. 12, p. 1145, 2019. doi: 10.3390/e21121145.