Multi-Resolution Mixer Network for Localization of Multiple Sensors from Cumulative Power Measurements

1st Md Mahmuddun Nabi Murad
Electrical Engineering Department
University of South Florida
Tampa, USA
mmurad@usf.edu

2nd Daniel Kwabena Yirenya-Tawiah
Electrical Engineering Department
Oregon State University
Corvallis, USA
yirenyad@oregonstate.edu

3rd Thomas Weller
Electrical Engineering Department
Oregon State University
Corvallis, USA
tom.weller@oregonstate.edu

4th Yasin Yilmaz
Electrical Engineering Department
University of South Florida
Tampa, USA
yasiny@usf.edu

***Abstract*—Wireless sensor networks (WSNs) can consist of many inexpensive sensors that communicate using the same wireless channel. In some applications, localization of these sensors can be as crucial as collecting their monitoring data. As the number of sensors increases, the complexity of processing data in such large networks grows significantly. In general, localization methods in WSNs typically rely on data containing sensor-specific information. However, the problem becomes more challenging when data contains no sensor-specific information. To address this issue, we propose a mixer-based deep neural network to estimate sensor positions using the received cumulative signal strength that is devoid of explicit sensor-specific information. Our approach employs wavelet decomposition to extract information from the input time series, combined with patching, embedding, and mixer techniques for position estimation. We compare the performance of our model with a nonlinear Kalman filter-based state estimation method. Extensive evaluations using data generated from our simulator demonstrate that our method consistently outperforms the unscented Kalman filter (UKF) method across all scenarios.**

***Index Terms*—wireless sensor network, sensor localization, sensor tracking, mixer network, unscented Kalman filter**

I. Introduction

A wireless sensor network (WSN) typically consists of multiple low-cost sensors positioned over an area to monitor the physical properties of the environment. Wildlife tracking [1], soil moisture monitoring [2], landslide monitoring [3], and military applications [4] are some of the important application domains for WSN. In many applications, we need the sensors' accurate positions along with their monitoring data. Therefore, localization techniques are an essential research topic for WSNs.

Various types of data, such as time difference of arrival (TDOA), time of arrival (TOA), angle of arrival (AOA), global positioning system (GPS) information, and received signal strength (RSS), are used to estimate the positions of the sensors in WSNs.

One of the straightforward solutions to localize the sensors is integrating the global positioning system (GPS) within the sensors. However, integrating GPS with all sensors is not viable for a cost-effective system. Recent advancements in machine learning have facilitated the development of numerous approaches for estimating sensor positions in WSNs. A machine learning method to refine the RSS for accurately estimating the sensor's position is proposed in [5]. In [6], the authors employ RSS to estimate the positions of sensors. However, these methods require extensive communication overhead, as all sensors interact with one another. In [7], another machine learning approach is introduced that trains a model using RSS fingerprint data for the entire experimental region. However, as the number of sensors increases, fingerprint-based methods become computationally intensive, rendering them impractical for real-world applications. Estimating the position of the underwater mobile sensor using the time difference of the arrival signal is proposed in [8]. In [9], the angle of the arrival signal is utilized to estimate the positions of the sensors. All of these methods require identifying and processing data from each sensor, making such WSNs both communication- and computation-intensive.

In this work, we propose a machine learning-based localization technique from total signal strength for cost-efficient WSNs in which sensors communicate in the same wireless channel anonymously. We designate sensors with known locations as interrogators and those with unknown locations as nodes. Our model estimates the positions of multiple nodes by analyzing the received cumulative signal strength at each interrogator. It does not require individual node recognition by

This work was funded by National Institute of Food and Agriculture under the award number 2023-67019-38829.

each interrogator, which allows for reduced power usage and the incorporation of more affordable sensors.

Our contributions can be summarized as follows:

- We propose a deep neural network architecture to estimate the nodes' current positions, leveraging the history of the received cumulative signal strength at each interrogator. Our model incorporates wavelet decomposition, patching, embedding, and mixing techniques to extract information efficiently.
- We introduce a simulator for low-cost wireless sensor localization to evaluate the performance of our model with respect to a popular nonlinear Kalman filter approach.

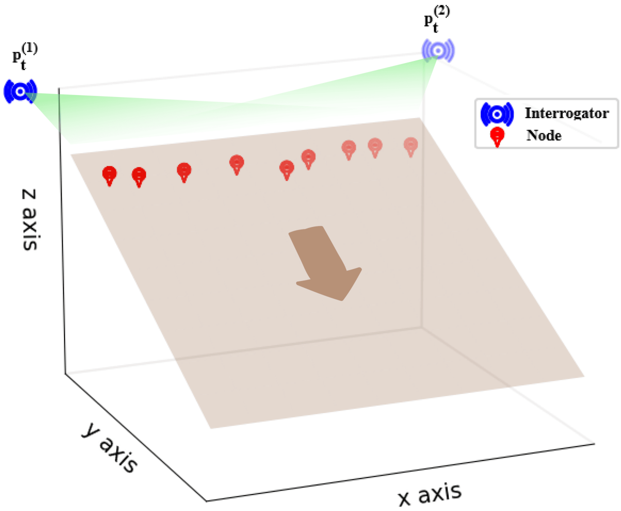

Fig. 1: A wireless network system with two interrogators and nine nodes. The positions of the interrogators are fixed and known, whereas the positions of the nodes are unknown, and they are moving downward across an inclined surface. $p_t^{(1)}$ and $p_t^{(2)}$ are the cumulative received power at time step t by interrogator-1 and interrogator-2, respectively. Our goal is to estimate the positions of the nodes.

II. Problem Statement

In various cost-efficient WSN applications, such as landslide monitoring and agricultural applications, estimating the sensor positions from limited communication data is a challenging task.

We consider a WSN in which a number of anchor nodes with known positions, denoted as interrogators in this work, receive signal from a known number of mobile sensors (nodes) in the same wireless channel (frequency band). Considering a landslide application, we use an inclined surface along which nodes move, as shown in Fig. 1. While the initial node positions can be known, their locations over time need to be estimated from the cumulative signals received by the interrogators. As the signals from different nodes interfere with each other in the same channel, making it infeasible to estimate the node positions from a single instance of the received cumulative signal, we consider the history of received cumulative signals, forming a time series at each interrogator.

Our goal is to estimate the positions of N nodes at each time step t, denoted as

$$\boldsymbol{S} = \{(x_t^{(1)}, y_t^{(1)}), (x_t^{(2)}, y_t^{(2)}), \dots, (x_t^{(N)}, y_t^{(N)})\} \in \mathbb{R}^{1\times 2N},$$

using the multivariate time series

$$\boldsymbol{P_L} = \{\boldsymbol{p}_{t-L+1}, \dots, \boldsymbol{p}_{t-1}, \boldsymbol{p}_t\} \in \mathbb{R}^{L\times M}$$

with a look-back window of length L. In this context, $\boldsymbol{p_t} = \{p_t^{(1)}, p_t^{(2)}, \dots, p_t^{(M)}\} \in \mathbb{R}^{1\times M}$ represents the cumulative power signals received by M interrogators at time t. As illustrated in Fig. 1, we assume the nodes move over time in an area with known geometry; hence, the z-axis location can be directly derived from the estimated x and y coordinates employing the known slope angle.

III. Proposed Method

We first normalize the received power series $\boldsymbol{P_L} \in \mathbb{R}^{L\times M}$ to $\underline{\boldsymbol{P_L}} \in \mathbb{R}^{L\times M}$ using the mean and standard deviation of the training dataset. To calculate the loss during model training, we then compare the estimated positions $\underline{\hat{\boldsymbol{S}}} \in \mathbb{R}^{1\times 2N}$ with the normalized ground truth positions $\underline{\boldsymbol{S}} \in \mathbb{R}^{1\times 2N}$. Our model integrates wavelet decomposition, patching, embedding, patch mixer, and embedding mixer to extract information from the history of the cumulative power series for estimating the current positions of the nodes. In the following subsection, we introduce the proposed model architecture[1].

A. Model Architecture

Following the method in [10], we utilize the multi-level discrete wavelet decomposition method to extract multi-resolution information from the input series in both time and frequency domains. The multi-level wavelet decomposition module outputs multiple detail coefficient series $\boldsymbol{X_{D_i}} \in \mathbb{R}^{M\times L_i}, i = 1, \dots, r$, and one approximation coefficient series $\boldsymbol{X_{A_r}} \in \mathbb{R}^{M\times L_r}$, where the detail coefficient series captures the detailed information and the approximation coefficient series captures the low-frequency information. L_i denotes the length of the coefficient series, and r is the number of decomposition levels. After the decomposition, we process each coefficient series utilizing individual resolution branches to ensure no information is lost. Each resolution branch combines a channel projection, patching and embedding, multiple mixers, and a head module.

With the channel projection module which consists of two linear layers with non-linear GELU activation, we project the coefficient series $\boldsymbol{X_{W_i}^T} \in \mathbb{R}^{L_i\times M}$ to $\underline{\boldsymbol{X_{W_i}^T}} \in \mathbb{R}^{L_i\times 2N}$. The transformation from M to $2N$ features is necessary as we perform the position estimation of N nodes with $2N$ features. By transposing $\underline{\boldsymbol{X_{W_i}^T}}$, we obtain $\underline{\boldsymbol{X_{W_i}}} \in \mathbb{R}^{2N\times L_i}$, which is subsequently passed through the patching and embedding modules [11]. The patching module captures the local information,

[1] The code is available at https://github.com/Secure-and-Intelligent-Systems-Lab/Localization-of-Sensors

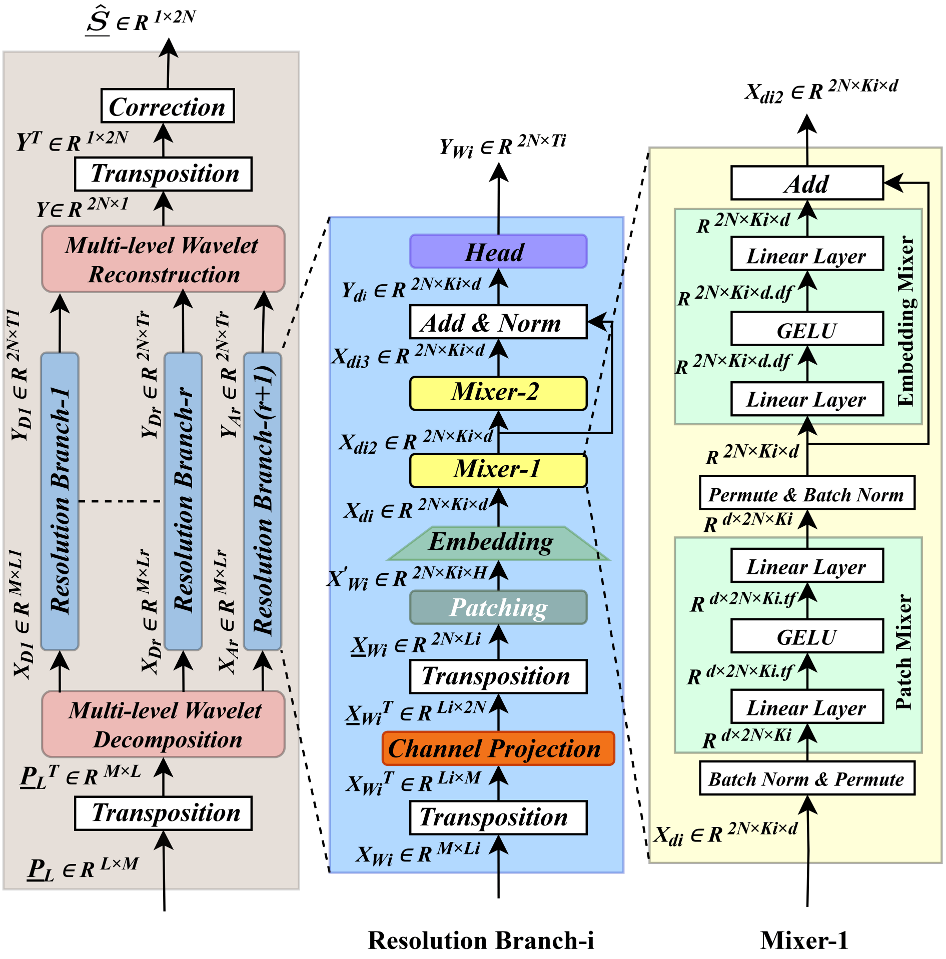

Fig. 2: Model estimates the positions of N nodes $\underline{\hat{\boldsymbol{S}}}$, using the cumulative power signal series $\underline{\boldsymbol{P_L}}$ of M interrogators. The terms $\boldsymbol{X_{W_i}}$ and $\boldsymbol{Y_{W_i}}$ refer to wavelet coefficient series (either approximation or detail).

and the embedding module encodes the data into d dimensions. Here, K_i and H represent the number of patches and the patch length, respectively.

To capture the global information from the local information of the patches, we further process the output of the embedding module $\boldsymbol{X_{d_i}} \in \mathbb{R}^{2N \times K_i \times d}$ employing multiple mixer modules. Each mixer module consists of a patch mixer and an embedding mixer. The patch mixer and embedding mixer are similar to the token mixer presented in [12], but they only differ from the token mixer in terms of mixing dimension. The patch mixer mixes the data in the patch dimension K_i by employing two linear layers and nonlinear GELU activation while the embedding mixer mixes the data in the embedding dimension d. In addition, we utilize expansion factor t_f in the patch mixer and d_f in the embedding mixer, which allows the model to capture the information in higher dimensions. Different from the patch mixer, the embedding mixer has a residual connection. In our model, we employ two mixer modules, where the second mixer module has a residual connection to facilitate the information flow efficiently. More mixer modules can be added if necessary. We add the output of the second mixer module and the residual connection before normalizing using 2D-batch normalization.

At the end of each resolution branch, we employ a head module consisting of a flatten layer and a linear layer. The flatten layer flattens the last two dimensions of $\boldsymbol{Y_{d_i}} \in \mathbb{R}^{2N \times\ K_i \times d}$, transforming the data from $\mathbb{R}^{2N \times K_i \times d}$ to $\mathbb{R}^{2N \times K_i . d}$. Subsequently, a linear layer projects the output of the flatten layer to $\boldsymbol{Y_{W_i}} \in \mathbb{R}^{2N \times T_i}$, where T_i is the sequence length of the predicted coefficient series $\boldsymbol{Y_{W_i}}$. To determine T_i, we pass an auxiliary series with the shape $\mathbb{R}^{2N \times 1}$ through the multi-level wavelet decomposition module at the initialization of the model. The sequence lengths of the output coefficient series are used as T_i for the corresponding resolution branch.

After obtaining the estimated approximation coefficient series $\boldsymbol{Y_{A_r}} \in \mathbb{R}^{2N \times T_r}$ and detail coefficient series $\boldsymbol{Y_{D_i}} \in \mathbb{R}^{2N \times T_i}, i = 1, \ldots, r$ from the resolution branches, we employ a multi-level wavelet reconstruction module to reconstruct the estimated positions $\boldsymbol{Y}$. After obtaining position estimates, we apply a correction module with the learnable weights

$K_1 \in \mathbb{R}^{2N}$ and bias $K_2 \in \mathbb{R}^{2N}$ using the formula: $\hat{\boldsymbol{S}} = (\boldsymbol{Y^T} \odot K_1 + K_2) \in \mathbb{R}^{1\times 2N}$, where $\odot$ denotes the element-wise multiplication.

IV. Experimental Setup

We consider an environment where nodes move downward along an inclined surface. Initially, node velocities are minimal but abruptly increase at some point in time and maintain high speeds until the end, similar to the situations observed in landslide monitoring systems. We define the inclined surface with x and y coordinates ranging from -10m to 10m and a 30-degree slope while the range of z coordinate depends on the slope angle. Since we set a constant value for the slope angle and will know it in practical applications, we only need to estimate the x, y coordinates of the nodes' positions. Using the estimated x, y coordinates and the slope angle, we can determine the z-coordinate of the nodes' positions. We simulate this environment using our simulator, explained in the next subsection, to assess our model's performance. The experimental setup comprises 9 nodes with initial x, y coordinates of (-8.5, 10.0), (-7.0, 10.0), (-4.5, 10.0), (-1.5, 10.0), (1.0, 9.5), (2.5, 9.3), (5.0, 9.8), (6.5, 9.8), and (8.5, 9.4), along with 2 stationary interrogators positioned at fixed x, y, z coordinates of (-12.0, 12.0, 10.0) and (12.0, 12.0, 10.0).

A. Simulator

The simulator of the system was derived from the electromagnetic properties of signals from both the interrogators and the nodes in our system. This simulator was generated using the MATLAB software. Our knowledge of these properties allowed us to determine the received power level at each interrogator based on the positions of the nodes and the input transmitted power.

In this free space simulator, we assign each interrogator and each node to a position in free space. We can also assign the sensors an orientation. With this, we can calculate the relative position of each node to the interrogator. This allows us to calculate how much gain the interrogating signal experiences relative to node position. We also take into account the polarization effect based on the initial orientation assigned to sensors. This allows us to include the influence of all polarization losses.

The simulator tracks the free space path loss to ensure we consider losses of the signal traveling to the node. The simulator considers the gain of the node sensors based on the interrogator position. It also considers the conversion loss due to the conversion of the first harmonic in the system to the second harmonic. The main reason for using a second harmonic system is to isolate the transmit signal from the receiving signal, thereby preventing adverse effects such as backscattering.

The free space path loss for the second harmonic is included in the simulation thereby providing us with enough information to determine the power level at the interrogators. To determine the influence of multiple nodes in the system, the simulator tracks the phase change of the signal from interrogator to node for the first harmonic and from node to interrogator for the second harmonic. Knowledge of both the phase and power allows the simulator to calculate the incident voltages.

The simulator performs a summation of these incident voltages before converting the total voltage back to power since we know the impedance of the interrogating antenna. The resulting power is the measurement that is comparable to the data we receive in practice.

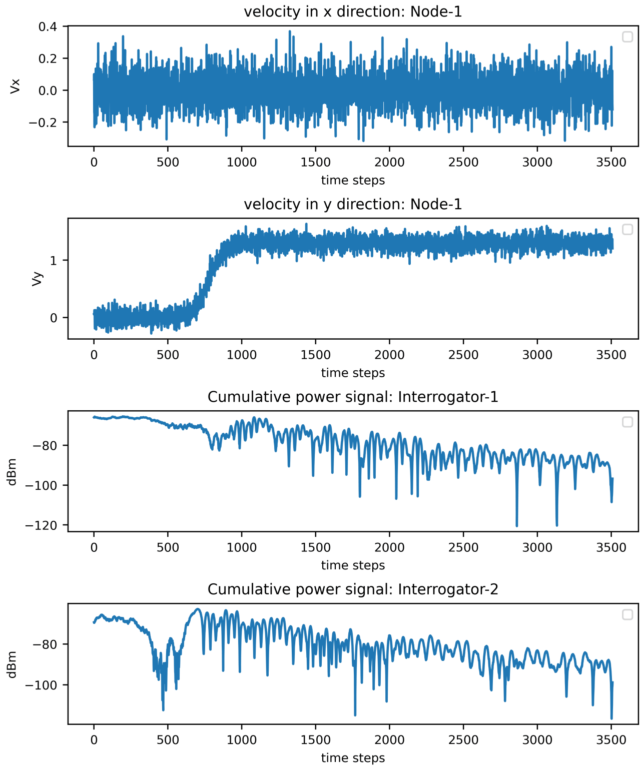

Fig. 3: Velocities of node-1, with u_1, u_2, u_3, and u_4 are set to 1, 1.5, 5, and 5.5, are shown in the first two sub-figures. The last two sub-figures illustrate the cumulative power signals received by the two interrogators.

B. Dataset

We utilize our simulator to generate the dataset based on a velocity model defined by the following equations:

$$V_x = \epsilon_x \tag{1}$$

$$V_y = \frac{a}{1 + e^{-c(t-b)}} + \epsilon_y \tag{2}$$

Here, $V_x \in \mathbb{R}^{1\times N}$ and $V_y \in \mathbb{R}^{1\times N}$ represent the velocities of N nodes in the x and y direction, respectively. The random variable $a \in \mathbb{R}^{1\times N}$, uniformly distributed between u_1 and u_2, corresponds to the magnitude of the velocities, while $b \in \mathbb{R}^{1\times N}$, uniformly distributed between u_3 and u_4, corresponds to the onset time of the velocity increase. The terms ϵ_x and ϵ_y represent the zero-mean Gaussian noise in the velocities, t denotes the time, and c is a constant parameter.

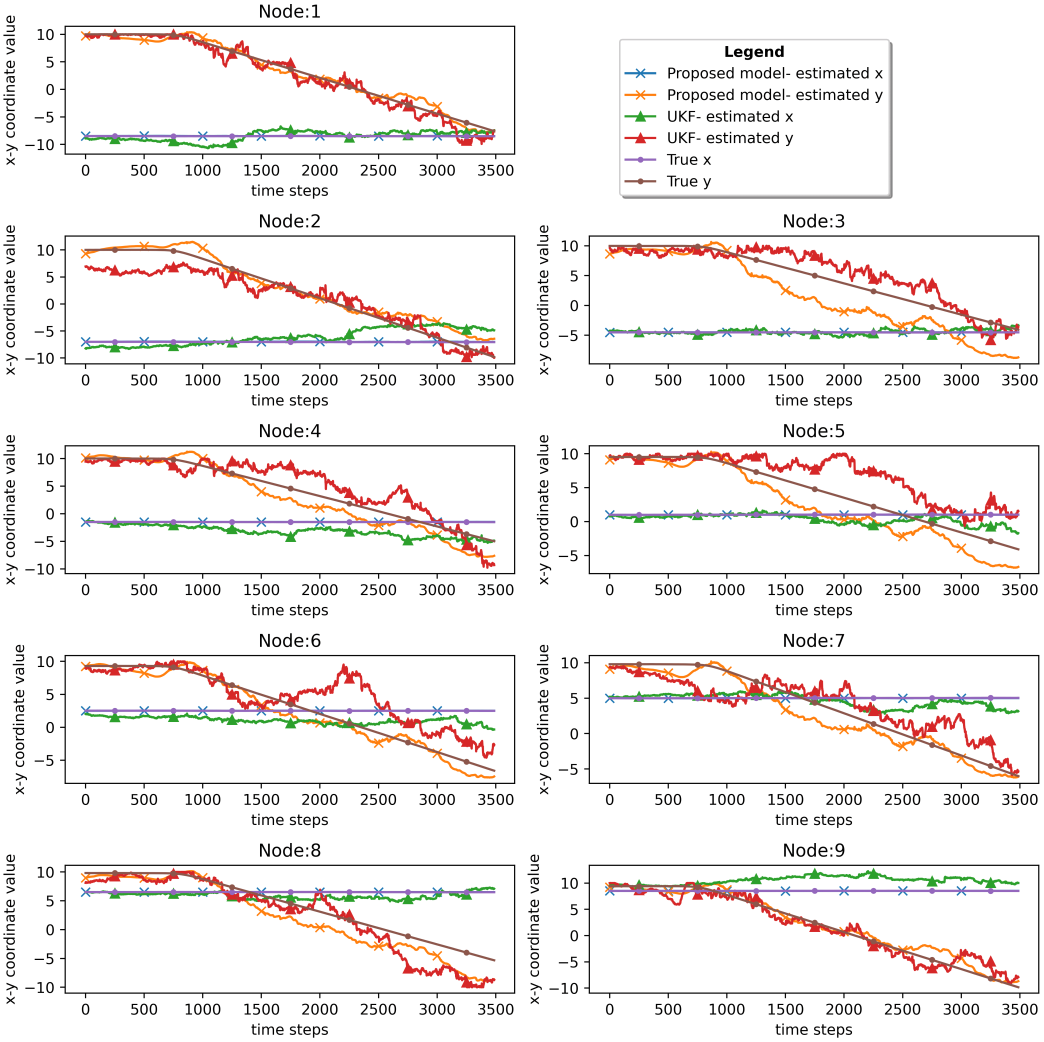

Fig. 4: Estimated nodes' positions of our proposed model and unscented Kalman filter with zero velocity assumption for the test dataset-2.

At the start of the node movements, we initialize a and b for all nodes individually and maintain the values throughout the movement. At each time step, with a time sampling interval dt, we sample the noise ϵ_x and ϵ_y from the Gaussian distribution and add to the velocities. Following the velocity equations, nodes are moved from their initial positions down the slope. We record the interrogators' cumulative power with the nodes' locations for each time step. Each sample in the dataset has 20 features representing the locations of 9 nodes by 18 features and the cumulative power of 2 interrogators by the remaining 2 features. Among all nodes, the velocity of node-1 and the cumulative received powers are presented in Fig. 3.

We run the simulator three times to create the training dataset while we generate the validation dataset from a single run. To evaluate the model's performance, we generate multiple test datasets, each corresponding to a separate simulator run. We use different u_1, u_2, u_3, and u_4 values when generating the training, validation, and testing datasets. The initial positions of the nodes and the interrogators' positions are consistent across all datasets since these positions will be known in practical applications. The dataset configurations are given in Table I.

TABLE I: Specifications of the datasets.

Dataset	Samples	u1	u2	u3	u4	c
Training	8887	1	1.5	1	2	4
Validation	2337	2	2.5	3	3.5	4
Testing-1	2234	2.5	3	4	4.5	4
Testing-2	3766	1	1.5	5	5.5	4
Testing-3	3159	1.5	2	5.5	6	4
Testing-4	2224	1.5	2	0.5	1	4

C. *Baseline*

As the relation between the nodes' position and the received power signal is nonlinear, we employ the unscented Kalman filter (UKF) state estimation method following the techniques presented in [13]. The UKF is well-suited for non-linear systems and provides a recursive state estimation. In our case, the state equation can be defined as,

$$\boldsymbol{S_t} = f(\boldsymbol{S_{t-1}}, \boldsymbol{v_{t-1}}) + \boldsymbol{w_{t-1}} \tag{3}$$

Here, $\boldsymbol{S_{t-1}}$ and $\boldsymbol{v_{t-1}}$ represent the positions and the velocity of the nodes at time step $(t-1)$, respectively. $\boldsymbol{w_{t-1}}$ denotes the process noise at time step $(t-1)$, while $f(.)$ is the linear state transition function. The measurement equation can be defined as,

$$\boldsymbol{p_t} = h(\boldsymbol{S_t}) + \boldsymbol{e_t} \tag{4}$$

Here, $\boldsymbol{p_t}$ represents the observed cumulative powers, and $\boldsymbol{e_t}$ represents the measurement noise at time step t. The non-linear measurement function $h(.)$ is implemented using our simulator.

Since we do not have access to the nodes' velocity, we estimate the average velocity for the nodes by leveraging the history of their estimated positions. However, in our analysis, we find that UKF performs well when we assume zero velocity for our problem.

D. Evaluation Metric

We assess our model's performance using the mean square error (MSE) metric, which compares the estimated and actual positions of the nodes in their original scale.

V. Results

We evaluate the performance of our proposed model in estimating the positions of nodes along an inclined surface by comparing it with the unscented Kalman filter-based estimation method. To ensure a fair comparison, we employ multiple test datasets, each featuring unique velocity patterns for the nodes. A look-back window of 256 past instances is used in all experiments to predict the next instance.

In Table II, we compare our model with the UKF method based on mean squared error (MSE) values. The results show that our proposed model outperforms the UKF model across all the test datasets. Since the UKF estimates node positions recursively, its estimation error gradually increases in our problem as the nodes move downward. Additionally, the UKF provides accurate estimation for only a few nodes, whereas our model achieves good estimations for almost all nodes. Fig. 4 illustrates the estimated positions for test dataset-2. From Fig. 4, it is evident that the UKF struggles to provide good estimations for both the x and y coordinates, whereas our proposed model consistently delivers good estimations for the x coordinate.

TABLE II: Performance comparison of the proposed model and UKF-based method.

	Proposed Model	UKF
	MSE	
Test dataset-1	0.500764	8.748231
Test dataset-2	1.778423	3.809604
Test dataset-3	0.763363	8.226274
Test dataset-4	0.735885	12.49711

VI. Conclusion

It is difficult to estimate the positions of the nodes from data that does not have node-specific information. To address this problem, we propose a mixer-based model to estimate the current positions of the nodes from the history of the cumulative power signals received by interrogators. We compare the performance of our model with the unscented Kalman filter-based estimation method. Through extensive evaluations using our simulator, we find that our model consistently outperforms the UKF method across all test datasets. Although our experiments focus on nodes moving along an inclined surface, our model applies to flat and inclined surfaces.

References

[1] J. P. Dominguez-Morales, A. Rios-Navarro, M. Dominguez-Morales, R. Tapiador-Morales, D. Gutierrez-Galan, D. Cascado-Caballero, A. Jimenez-Fernandez, and A. Linares-Barranco, "Wireless sensor network for wildlife tracking and behavior classification of animals in doñana," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2534–2537, 2016.

[2] J. John, V. S. Palaparthy, S. Sarik, M. S. Baghini, and G. S. Kasbekar, "Design and implementation of a soil moisture wireless sensor network," in *2015 Twenty First National Conference on Communications (NCC)*. IEEE, 2015, pp. 1–6.

[3] K. Sangrit, J. Karnjana, S. Laitrakun, K. Fukawa, S. Fugkeaw, and S. Keerativittayanun, "Distance estimation between wireless sensor nodes using rssi and csi with bounded-error estimation and theory of evidence for a landslide monitoring system," in *2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, 2021, pp. 40–45.

[4] A. Ali, Y. K. Jadoon, S. A. Changazi, and M. Qasim, "Military operations: Wireless sensor networks based applications to reinforce future battlefield command system," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*. IEEE, 2020, pp. 1–6.

[5] C. L. Nguyen, O. Georgiou, and V. Suppakitpaisarn, "Improved localization accuracy using machine learning: Predicting and refining rss measurements," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–7.

[6] W. Suzhe and L. Yong, "Node localization algorithm based on rssi in wireless sensor network," in *2012 6th International Conference on Signal Processing and Communication Systems*. IEEE, 2012, pp. 1–4.

[7] T. Alhmiedat, "Fingerprint-based localization approach for wsn using machine learning models," *Applied Sciences*, vol. 13, no. 5, p. 3037, 2023.

[8] F. Liu, H. Chen, L. Zhang, and L. Xie, "Time-difference-of-arrival-based localization methods of underwater mobile nodes using multiple surface beacons," *IEEE Access*, vol. 9, pp. 31 712–31 725, 2021.

[9] T.-K. Le and K. Ho, "Joint source and sensor localization by angles of arrival," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6521–6534, 2020.

[10] F. Cotter, "Uses of complex wavelets in deep convolutional neural networks," Ph.D. dissertation, Apollo - University of Cambridge Repository, 2019. [Online]. Available: https://www.repository.cam.ac.uk/handle/1810/306661

[11] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[12] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[13] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.