



Event-driven localization in wireless sensor networks: a machine learning approach

Mansoor Iqbal^{1,2} · Sadia Waheed² · Inayat Ali³ · Syed Zarak Shah⁴ · Syed Ahmad Saleem Bokhari⁵ · Shahid Sultan²

Received: 13 October 2025 / Accepted: 23 January 2026

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2026

Abstract

This paper presents a machine learning (ML)-based localization framework for event-driven localization in wireless sensor networks (WSNs) to address the limitations of traditional multi-sequence positioning (MSP) and event-based techniques. The proposed method formulates node localization as a supervised regression problem, where sensor coordinates are predicted from feature vectors comprising event detection times, anchor node proximities, and spatial event information. A lightweight multilayer perceptron (MLP) architecture is employed due to its efficiency and ability to capture nonlinear spatiotemporal patterns without the computational burden of graph- or sequence-heavy models. The model is trained on synthetically generated datasets that emulate enhanced multi-sequence positioning (EMSP) behavior, incorporating noisy event propagation and variable node placements. Large-scale EMSP-style data generation and robustness benchmarking require high-throughput computation; therefore, the proposed evaluation and training pipeline is designed to leverage GPU acceleration and parallel/distributed execution across deployment scenarios. This computing-oriented workflow enables rapid, reproducible localization studies for large WSNs while preserving lightweight inference suitable for real-time edge operation. Extensive experiments are conducted under sparse and dense deployment scenarios, with comparative benchmarking against EMSP, RSSI, ToA, EventMamba, FasterKAN, and GNN-based localization methods. Results demonstrate that the proposed approach achieves superior accuracy, reduced prediction variance, and robust anchor utilization across varying topologies. Notably, under the most adverse tested conditions, including severe timing noise, 10% event packet loss, and imperfect synchronization, the proposed model maintains graceful degradation and continues to outperform EMSP and signal-based baselines. Furthermore, interpretability is enhanced through spatial visualization tools, including 2D true vs. predicted maps and Voronoi-based consistency analysis. Overall, the framework offers a scalable and generalizable solution, positioning event-driven

Extended author information available on the last page of the article

Published online: 09 February 2026

Springer

ML-based localization in WSNs as an effective alternative to conventional strategies in resource-constrained environments.

Keywords Event-driven localization · Wireless sensor networks · ML-based localization · Multi-sequence positioning · Multilayer perceptron

1 Introduction

Wireless sensor networks (WSNs) have emerged as a transformative technology for real-time monitoring and managing physical environments. Their applications span various domains, including environmental monitoring, military surveillance, industrial automation, and smart agriculture [1]. A critical aspect underpinning the effectiveness of WSNs is node localization—the ability to determine the physical coordinates of sensor nodes. Accurate localization enables contextual understanding of sensor readings and supports spatial analytics essential for decision-making processes. In conventional practice, localization is either enabled via GPS (cost-ineffective and energy-consuming) or through manual configuration (infeasible in large-scale or hostile terrains). Consequently, automated and scalable localization strategies have become a core research priority in WSN design.

Despite significant advances, most current localization methods involve range-based (e.g., received signal strength indicator (RSSI), time of arrival (ToA), time difference of arrival (TDoA)) performance or connectivity-based heuristics that are severely affected by conditions in the real world, such as signal attenuation and environmental effects, or uneven deployment patterns [2, 3]. Range estimates in dynamic or constrained environments where the signal quality varies are inaccurate, causing poor localization performance. Also, techniques that presuppose the equal distribution of the anchors or the use of additional devices become impractical in the remote or limited-resource Deployment. These shortcomings highlight a persistent challenge in achieving the low-cost, low-overhead, and scalable localization in WSNs with little hardware and versatile techniques.

Event-based localization was recently suggested as a simple alternative. An example of such techniques, the enhanced multi-sequence positioning (EMSP), infers the position of nodes by using the temporal order in which the nodes in the network detect controlled events [4]. EMSP assigns approximate node locations using event arrival order, anchor detection, and sequence grids. The terms event-driven localization, EMSP-style localization, and multi-sequence positioning are used in this paper to refer to closely related concepts. Event-driven localization is the general concept of determining nodes' positions from the times of event detection, rather than from explicit range or angle measurements. Multi-sequence positioning (MSP/EMSP) is a particular deterministic model of this paradigm; EMSP-style theorizes datasets and problem instances that satisfy the same assumptions and inputs without necessarily using the original EMSP algorithm.

However, current EMSP implementations rely on fixed topologies and struggle with irregular node distributions, sparse anchors, or non-homogeneous terrains.

Moreover, their deterministic heuristics are brittle to noise and lack adaptability. The study follows and supports a high-performance computing (HPC)-oriented workflow, using parallel/distributed EMSP-style simulation and GPU-accelerated training to benchmark robustness across heterogeneous deployments and impairment settings. Even for 500 node-datasets, this allows for high-throughput event-node generation and repeatable multi-scenario evaluations, which are inefficient under single-node execution. This paper seeks to bridge this gap by proposing a data-driven alternative to EMSP. The primary objectives of this research are as follows:

- To design an ML-based framework for event-driven WSN localization.
- To explore synthetic EMSP-style data generation for diverse deployments.
- To evaluate the feasibility of replacing deterministic EMSP rules with supervised learning.

Existing public WSN localization datasets (e.g., Kaggle and UCI repositories) are primarily designed for range-based modalities, such as RSSI/ToA, or for trajectory tracking problems and therefore do not contain the event sequence structure, anchor visibility patterns, and spatiotemporal variability required for studying EMSP-type localization. This gap is not incidental: event-driven deployments introduce distinct practical constraints—detections may be partial or missing due to packet loss, occlusions, or limited sensing range; detection times are asynchronous under imperfect clock synchronization; and the number, placement, and overlap of events vary across deployments—conditions that directly violate the stability and completeness assumptions underpinning RSSI/ToA and many MSP/EMSP variants. Accordingly, to accurately assess localization under these event-driven constraints, a synthetic dataset is generated with controllable deployments, anchor placement, and noise/impairments, allowing for systematic analysis in heterogeneous and imperfect working regimes.

1.1 Research gap and contribution

Given the limitations of existing event-based strategies, a notable research gap emerges in exploring data-driven, adaptive models that can generalize across varying deployment configurations, event distributions, and environmental dynamics. While MSP and its variants offer structure, they remain largely handcrafted and topologically constrained. Existing work [5, 6, 17, 33] has investigated ML techniques for WSN localization, but these approaches are often tailored to RSSI-based or range-based modalities and rarely consider EMSP-style event sequencing under dynamic deployments. Consequently, there is limited exploration of how ML models can emulate or extend EMSP-like behavior under diverse WSN conditions using real-time, learnable representations.

To address this gap, this study proposes an ML-based localization framework that learns the spatial inference from structured event-driven features. A multilayer perceptron (MLP) is trained on EMSP-inspired synthetic data to estimate two-dimensional sensor coordinates based on the detection times, the visibility of the anchors, and the geometry of the event without using explicit encoding of the sequence, grid

construction, and deterministic topology constraints. The synthetic data generation pipeline is used as an evaluation enabler, to systematically control the anchor density, the event distribution, noise and deployment geometry, which helps support reproducible and stress-tested performance analysis.

While more complex architectures like CNNs, GNNs, and state-space models have also been used in localization tasks, they are more expensive to compute and to store in memory and usually require additional structure. The localization inputs, in this work, are, on the one hand, fixed-length, tabular-structured features, including event detection times, anchor visibility, and event geometry. With this representation, a small MLP suffices to map event-anchor features to spatial coordinates without the baggage of graph construction, message passing, or kernel expansion. The suggested architecture is thus focused on representation-architecture match, as opposed to model richness, and attains competitive precision with minimal computational complexity, low-latency inference, and is practical for resource-limited WSN implementations.

To ensure reliability and transparency, the proposed framework is evaluated across different deployment scenarios, including sparse and dense anchor configurations, heterogeneous geometry, and impairments. Performance is benchmarked with respect to classical approaches (EMSP, RSSI, ToA) as well as recent learning-based approaches (EventMamba, FasterKAN, GNNs) as well as in terms of quantitative error metrics and spatial visualizations.

The key contributions of this work are:

- A supervised learning framework for WSN node localization that learns EMSP-like behavior from detection time, anchor proximity, and event geometry as inputs.
- A controlled synthetic evaluation pipeline enabling systematic analysis of localization robustness under diverse deployment geometries and impairments.
- A compact MLP-based localization model that achieves competitive accuracy with low computational and memory overhead.
- An extensive experimental study, including ablation, sensitivity, and comparative analyses across sparse and dense deployments.
- Integration of spatial visualization tools (true vs. predicted maps, detection surfaces, and Voronoi partitions) to enhance interpretability and diagnostic insight.

Together, these contributions establish a scalable and adaptive foundation for event-driven localization in modern WSN deployments. The remainder of this paper is structured as follows: Sect. 2 reviews related work on wireless sensor network localization, EMSP methods, and ML techniques. Section 3 describes the data generation pipeline and architecture of the proposed learning model. Section 4 outlines the measurement procedures and experimental conditions. In Sect. 5, results are analyzed through comparisons of EMSP and ML-based methods in terms of numerical accuracy and visualization, and the efficiency of the proposed approach using a real-world scenario. Finally, Sect. 6 concludes the paper and discusses directions for future research.

2 Literature review

Localization in WSNs remains a fundamental challenge, particularly in large-scale deployments where infrastructure-based solutions such as GPS are impractical due to cost, energy consumption, or environmental constraints. Over the years, a wide range of range-based (e.g., RSSI, ToA, TDoA) and range-free (e.g., distance vector-Hop (DV-Hop), and centroid) methods have been proposed, offering varying trade-offs in terms of accuracy, scalability, and computational complexity [7, 8]. More recently, event-driven localization strategies have gained attention for their energy efficiency, simplicity, and independence from specialized ranging hardware. Within this class, MSP has emerged as a prominent approach, exploiting temporally alternating detection sequences of engineered events to position nodes in two-dimensional space. Several studies have attempted to optimize MSP to improve its soundness and accuracy. For example, [9] proposed optimizations that consider sequence flips and partial event detection, and demonstrated their framework using simulations and real-world testbeds to support sub-meter precision without requiring additional event control machinery. Similarly, [10] also examined MSP with large-scale outdoor WSNs, where it was demonstrated to be scalable across a range of deployment densities, anchor locations, and event distributions. In [11], complementary work compared MSP and ToA-based schemes, demonstrating that MSP is more resilient to signal obstruction in close and intermediate ranges, but is limited by cell size and event distribution. All these works highlight the potential that MSP holds as a low-cost solution, but also demonstrate that it is dependent on deterministic heuristics and event rigidities, which constrain its ability to generalize to noisy and heterogeneous deployments.

Similar attempts have focused on applying ML to event-driven sensing and localization within WSNs. For example, [12] proposed an ML-based strategy for efficient node localization, demonstrating how supervised models can significantly improve accuracy and robustness in dynamic environments. In a similar manner, [13] leveraged spatiotemporal event information to predict localization error, demonstrating how data-driven adaptation enhances resilience across heterogeneous deployments. Bio-inspired methods have also been applied, where energy efficiency, bandwidth reduction, and redundancy control are emphasized in schemes such as the Enhanced Bio-Inspired Energy-Efficient Localization (EBEEL) framework, and further expanded in hybrid approaches integrating mobility-aware designs [14–17].

Recent studies have developed advanced ML models for event-driven WSNs: for example, in [18], enhanced DV-Hop improvements based on augmented datasets and neural network learning models have been used to improve localization accuracy via synthetic anchors and deep architectures significantly. In contrast, ensemble learning pipelines have been used to enhance the robustness of models by integrating models such as decision trees and regression learners for aggregate prediction [19]. In parallel, state-of-the-art performance has been demonstrated in localization tasks by spatiotemporal event processing methods based on point cloud representations and state-space models, with minimal computational

overhead [20]. In contrast, CNN-based ones, especially those based on 1D and 2D architectures for fingerprint-like spatiotemporal patterns, have demonstrated a strong capability in capturing the underlying spatial correlations within sensors [21].

Together, these studies illustrate how event-driven features can be effectively harnessed by modern ML architectures, further motivating the development of lightweight yet expressive deep learning models in WSN localization [22, 23]. Collectively, these works highlight the applicability of event-driven capabilities in combination with ML architectures that make it possible to use scalable and flexible spatial inference in resource-limited WSN settings. Based on these insights, recent studies have been gradually shifting toward deep learning models, which have further improved event sequence representation and nonlinear spatiotemporal learning for localization.

Moreover, recent research has focused more on deep learning (DL) techniques that are better capable of modeling nonlinear and high-dimensional spatiotemporal dependencies in WSN localization. [24] Finger printing based DL techniques using regression network and classification models such as k-nearest neighbors (KNN) [25], support vector machines (SVM) [26] and ensemble architecture [27] have shown high accuracy under indoor and outdoor scenarios [28–32]. More advanced frameworks include FasterKAN [33], which utilizes kernel-adaptive neural networks to enhance robustness against noisy inputs, and GNNs, which excel in capturing irregular sensor topologies and spatial correlations [34]. Collectively, these methods point out the development from the conventional ML into deep architectures that can learn complex spatial and temporal associations that are frequently ignored by the conventional algorithms, hence establishing the basis for scalable and adaptive WSN localization. Table 1 provides a comparative summary of these approaches, where the proposed method enhances scalability, robustness, and interpretability, while also being lightweight.

In summary, existing research provides a strong foundation for MSP-type event-driven localization and ML enhancements in WSNs. However, most MSP approaches are still based on handcrafted heuristics and inflexible topology assumptions, and learning-based approaches mostly focus on RSSI or fingerprinting scenarios rather than structured event sequences typical for EMSP-style localization. To provide a clearer comparative perspective, Table 1 summarizes the principal localization methods as explained in this section, in terms of their dependence on labeled data, anchors, and topology assumptions, as well as their computational and data requirements. Notably, the $\times$ entries of the table reflect low or no dependency on the respective requirement rather than a limitation. In particular, the proposed method does not require strong topology assumptions, as it learns spatial relationships from detection times, anchor visibility, and event geometry, without relying on fixed grids or predefined graphs. Moreover, the compact MLP design's low computational and memory requirements are a characteristic that makes the approach well-suited for resource-constrained WSN deployments. Motivated by these observations, the following section presents the details of the data set generation process, feature engineering pipeline, and MLP-based model architecture to form the basis of the proposed framework.

Table 1 Feature-Comparison of WSN Localization Methods

Method Family	Needs Labels	Needs Anchors	Strong Topology Assumption	High Compute	High Memory	High Data Demand
Range-based / Range-free [7, 8]	×	✓	×	×	×	×
MSP / EMSP [9–11]	×	✓	✓	×	×	×
Classical ML-based Localization [12, 13]	✓	✓	×	△	△	△
Bio-inspired / Hybrid [14–17]	×	✓	△	△	×	×
Ensemble / Augmented ML [18, 19]	✓	✓	×	✓	✓	✓
Spatiotemporal Point-cloud/CNN [20, 21]	✓	△	×	✓	✓	✓
DL Fingerprinting [22, 23, 28–30]	✓	△	×	✓	✓	✓
Transformers / GNN / FasterKAN [25, 26, 33, 34]	✓	✓	×	✓	✓	✓
Proposed Method	✓	✓	×	×	×	△

✓: Yes/High; ×: No/Low; △: Conditional/Moderate (depends on deployment scale or model configuration)

3 Proposed methodology

Localization in WSNs requires precise, scalable, and low-overhead techniques that can function under diverse deployment conditions. Unlike prior works that employ generic neural networks or rely exclusively on deterministic heuristics, the proposed model is explicitly optimized for structured spatiotemporal inputs. The architecture employs progressively shrinking hidden layers ($128 \rightarrow 64 \rightarrow 32$ neurons) to balance expressiveness with computational efficiency, a design choice tailored for WSNs where over-parameterization is impractical. Hyperparameters were systematically tuned through a grid search, including the learning rate ($1e-4$ to $1e-2$, with the best result at $1e-3$), batch size (16–128, with the best result at 32), and the number of epochs (up to 200, with early stopping at a validation plateau). To improve generalization under noisy conditions, dropout (0.2–0.4, best at 0.3) and batch normalization were incorporated in the hidden layers, as summarised in Table 2. Collectively, these design choices ensure that the MLP achieves a favourable trade-off between accuracy, stability, and efficiency.

Since the proposed framework is data-driven, its performance eventually hinges on the accessibility of appropriate training and evaluation datasets. In practice, however, labeled datasets in WSN localization are extremely rare because of the expense, time, and logistical implications of deploying dense sensor networks with ground-truth positioning. Therefore, previous investigations in the WSN field have often used simulation-based synthetic data to test localization systems in a controlled environment. As an example, the MSP technique was proposed by [9] using synthetic data in order to test its scaling and performance, whereas [11] also produced MATLAB-based data sets to compare MSP with RSSI and ToA. Synthetic data have also become more popular in more recent works, to test novel scheme localization at scale, when otherwise the experimental setups are infeasible [7, 8, 26]. Based on this established practice, a Python-based synthetic data generator is used to simulate EMSP-like behavior, such as event propagation, anchor visibility, and Gaussian noise injection. This approach can allow reproducibility, evaluation across a wide range of network topologies, and scale to train ML models in conditions where real-world data might be prohibitive to access.

3.1 System model and architecture

The proposed localization model is formulated as a supervised regression task, where the goal is to predict the two-dimensional coordinates (x, y) of a sensor node based on event detection features and the proximity of the anchor node. An MLP is selected as the core model architecture due to its ability to capture complex nonlinear relationships among input features while maintaining training efficiency and architectural simplicity. The working of the unit perceptron is shown in Fig. 1. Compared to sequence-based models like Long Short-Term Memory (LSTM) or spatial models like GNNs, the MLP offers a favourable

Table 2 Hyperparameter tuning process and selected values for the proposed MLP model

Parameter	Search range/Options	Selected value	Rationale
Learning Rate	1e-4, 5e-4, 1e-3, 5e-3, 1e-2	1e-3	Achieved fastest convergence with stable validation loss
Batch Size	16, 32, 64, 128	32	Balanced gradient stability and training speed
Epochs	up to 200	100 (early stopping at 85)	Prevented overfitting while ensuring sufficient training cycles
Hidden Layers	[128–64–32], [256–128–64], [64–32]	128–64–32	Optimal trade-off between complexity and resource constraints
Activation Function	ReLU, Tanh, Leaky ReLU	ReLU	Provided faster convergence and lower training loss
Dropout Rate	0.2, 0.3, 0.4	0.3	Reduced overfitting without harming accuracy
Regularization	None, L2 (0.001–0.01)	Batch Norm + Dropout	Improved stability under noisy features
Optimizer	Adam, SGD, RMSProp	Adam	Consistently yielded the lowest error and faster convergence

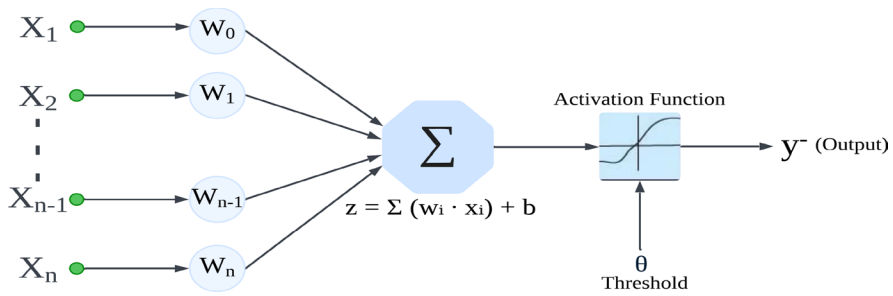


Fig. 1 Unit perceptron architecture

trade-off between performance and computational complexity, making it well-suited for real-time or resource-constrained WSN environments.

The proposed MLP consists of an input layer followed by three fully connected hidden layers with ReLU activation functions. Specifically, the architecture includes: An input layer of size N , where N denotes the number of features (detection times, anchor range indicators, event coordinates). Three hidden layers of size 128, 64, and 32 neurons, respectively. A final output layer with two neurons for predicting the (x, y) coordinates.

The model is trained using the mean squared error (MSE) loss function, optimized via the Adam optimiser with an empirically tuned learning rate and batch size, as shown in Fig. 2. The MLP was selected over more complex architectures as it demonstrates strong performance on structured tabular data, where the order of events is implicitly encoded in the feature vector.

3.2 Mathematical formulation of the proposed localization model

In the proposed framework, sensor node localization is formulated as a supervised regression task, where each node's position is inferred from a structured set of input features. Event detection times record the relative arrival times of events at different nodes, implicitly encoding the information about the distance between the nodes and the event sources through propagation dynamics. Anchor visibility is represented using binary indicators that define whether a node is within the communication range of one or more specified anchors, providing coarse spatial constraints that are crucial for reducing positional uncertainty. Furthermore, event source coordinates provide a fixed spatial reference for the origination of events, allowing for the triangulation of sensor positions when combined with temporal and connectivity information. Collectively, these features form a comprehensive representation that balances temporal, spatial, and topological information, thereby allowing the model to learn robust mappings from sensor observations to true node coordinates in a two-dimensional plane.

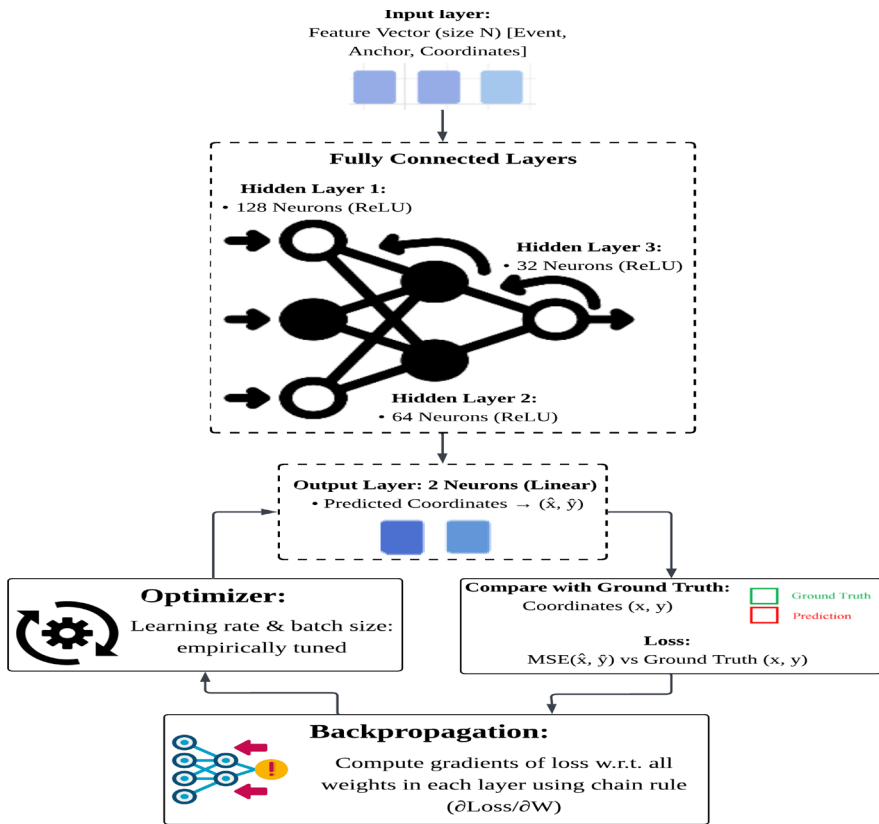


Fig. 2 Multilayer perceptron architecture

3.2.1 Input feature vector construction

Let:

- N denote the total number of sensor nodes to be localized,
- E denote the number of known event sources,
- A denote the number of anchor nodes with known positions.

For each sensor node $i \in \{1, 2, \dots, N\}$, a fixed-length input feature vector is constructed by combining temporal, topological, and geometric information, as described below.

Event Detection Time Vector: Each node records the time at which it first detects each event. This information is represented as

$$\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{iE}]^T \in \mathbb{R}^E, \quad (1)$$

where t_{ie} denotes the detection timestamp of event e at node i . The detection time is assumed to depend on the Euclidean distance between the node and the event source, subject to propagation delay and sensing noise. If an event is not detected by node i , t_{ie} is assigned a predefined masking value (e.g., 0 or -1).

Anchor Visibility Vector: Anchor proximity information is encoded using a binary visibility vector:

$$\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{iA}]^T \in \{0, 1\}^A, \quad (2)$$

where $a_{ij} = 1$ if sensor node i lies within the communication range of anchor node j , and $a_{ij} = 0$ otherwise.

Global Event Coordinate Vector: Each event e has a known spatial location $(x_e, y_e) \in \mathbb{R}^2$. The coordinates of all events are flattened into a global vector shared across all nodes:

$$\mathbf{p}_e = [(x_1, y_1), (x_2, y_2), \dots, (x_E, y_E)]^T \in \mathbb{R}^{2E}. \quad (3)$$

This vector provides geometric context that links detection times to spatial structure.

Final Input Feature Vector: The complete input to the learning model for node i is obtained by concatenating the above components:

$$\mathbf{z}_i = [\mathbf{t}_i^T \parallel \mathbf{a}_i^T \parallel \mathbf{p}_e^T]^T \in \mathbb{R}^{3E+A}, \quad (4)$$

where $\parallel$ denotes vector concatenation. The dimensionality of the input vector is therefore given by

$$\begin{aligned} \dim(\mathbf{z}_i) &= E \text{ (event detection times)} + A \text{ (anchor visibility flags)} \\ &\quad + 2E \text{ (event coordinates)} \\ &= 3E + A. \end{aligned} \quad (5)$$

Assumption on Event Source Coordinates: The inclusion of event source coordinates in the input feature vector is based on a rational conception that is consistent with EMSP-style localization. In event-driven WSN deployments, events are also generally controlled, scheduled, or externally triggered (e.g., calibrated beacons, test excitations, or known environmental stimuli), and their spatial coordinates are thus known to the system at the time of inference. These coordinates are global context variables and are shared among all sensor nodes, providing no node-specific distance information. Instead, localization remains indirect and is derived from the interaction between the event geometry, detection time delays, and the visibility of the anchors. This design is helpful in compensating for the lack of explicit range or angle measurements and in making strong inferences under partial, asynchronous, and noisy event detections. The effect of deleting event-coordinated information is explicitly considered in the ablation study, where a clear degradation in localization accuracy is observed, confirming that the task is neither too easy nor over-specified.

3.2.2 Target output vector

The ground-truth spatial location of sensor node i is represented as a two-dimensional coordinate vector:

$$\mathbf{r}_i = [x_i, y_i]^T \in \mathbb{R}^2, \quad (6)$$

where (x_i, y_i) denotes the true physical position of node i in the deployment area.

3.2.3 Learning objective

The localization task is formulated as a supervised regression problem. The objective is to learn a function

$$f_\theta : \mathbb{R}^{3E+A} \rightarrow \mathbb{R}^2, \quad (7)$$

parameterized by trainable weights θ , which maps the input feature vector $\mathbf{z}_i$ to a predicted node location:

$$\hat{\mathbf{r}}_i = f_\theta(\mathbf{z}_i). \quad (8)$$

Model parameters are optimized by minimizing the mean squared localization error over all N sensor nodes:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{r}_i - \hat{\mathbf{r}}_i\|_2^2, \quad (9)$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

3.2.4 Model architecture

The function f_θ is implemented as a MLP with a fixed-depth feed-forward structure. The network architecture is defined as follows:

- Input layer: $3E + A$ neurons (corresponding to $\mathbf{z}_i$),
- Hidden layers:
 - Layer 1: 128 neurons with ReLU activation,
 - Layer 2: 64 neurons with ReLU activation,
 - Layer 3: 32 neurons with ReLU activation,
- Output layer: 2 neurons, representing the predicted spatial coordinates.

For a given input feature vector $\mathbf{z}_i \in \mathbb{R}^{3E+A}$, forward propagation through the network is defined as:

$$\mathbf{h}_1 = \text{ReLU}(W_1 \mathbf{z}_i + \mathbf{b}_1), \quad (10)$$

$$\mathbf{h}_2 = \text{ReLU}(W_2 \mathbf{h}_1 + \mathbf{b}_2), \quad (11)$$

$$\mathbf{h}_3 = \text{ReLU}(W_3 \mathbf{h}_2 + \mathbf{b}_3), \quad (12)$$

$$\hat{\mathbf{r}}_i = W_4 \mathbf{h}_3 + \mathbf{b}_4, \quad (13)$$

where $W_k \in \mathbb{R}^{n_{k+1} \times n_k}$ and $\mathbf{b}_k \in \mathbb{R}^{n_{k+1}}$ denote the weight matrix and bias vector of layer k , respectively, and $\text{ReLU}(x) = \max(0, x)$.

3.2.5 Loss function and optimization

The model parameters $\theta = \{W_k, \mathbf{b}_k\}$ are optimized by minimizing the MSE between the predicted and ground-truth node coordinates:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{r}_i - \hat{\mathbf{r}}_i\|_2^2, \quad (14)$$

where $\mathbf{r}_i = [x_i, y_i]^T$ denotes the true position of node i , and $\hat{\mathbf{r}}_i$ is the corresponding prediction.

Optimization is performed using the Adam optimizer with an empirically selected learning rate and batch size. Model parameters are updated via backpropagation:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}(\theta), \quad (15)$$

where η denotes the learning rate.

3.2.6 Prediction

After training, the spatial location of an unseen sensor node i is estimated as:

$$\hat{\mathbf{r}}_i = f_{\theta}(\mathbf{z}_i). \quad (16)$$

In this work, localization is performed in a two-dimensional Euclidean coordinate space to ensure direct comparability with EMSP-style benchmarks; however, the proposed formulation is dimension-agnostic and can be extended to 3D localization by augmenting event/node coordinates and modifying the regression output to (x, y, z) with no change to the core learning objective.

Having defined the model mathematically, we next outline the baseline methods and ablation variants used to benchmark and justify the proposed design choices.

3.3 Baseline comparisons and ablation studies

To put the performance of the proposed MLP framework into perspective, baseline heuristics are introduced that are commonly used in WSN localization, and perform ablation studies to isolate the effect of individual architectural choices. These play two complementary functions: (i) comparison of performance with existing methods

of localization, and (ii) the separation of the effect of particular design decisions in the model.

Baseline Methods: The proposed approach is compared against classical WSN localization heuristics, including:

- RSSI-based trilateration,
- ToA triangulation methods,
- EMSP heuristics.

These methods provide reference points for accuracy and robustness, highlighting the improvements brought by a data-driven approach.

Ablation Studies: To analyze the effect of architectural and training choices, the following ablations are considered:

- Removal of regularization components (with/without Dropout, with/without Batch Normalization),
- Alternative hidden layer configurations to assess parameter–performance trade-offs. Configurations are [256–128–64], [64–32].
- Different input feature compositions, such as excluding event coordinates or anchor indicators, to evaluate their contribution.

These comparisons ensure that reported gains are not incidental but stem from specific design decisions in the proposed model.

3.4 Synthetic dataset generation

The synthetic dataset was created with the help of a Python simulation pipeline (with NumPy and SciPy). The framework is dynamic because it models the propagation of events, visibility of anchors, and noise injection, and ensures the direct compatibility with ML, as shown in Fig. 3. Compared to traditional simulations, such as those offered by classic software packages like MATLAB or NS-3, it allows for easy data generation, preprocessing, and training of the model in Python, improving reproducibility and scalability.

The dataset simulates the localization of EMSP-style to support model training and evaluation. This design overcomes the challenges of having labeled data from real-world WSNs that are often expensive, time-consuming or impractical. The simulated environment represents a two-dimensional region of fixed size, within which:

- A specified number of sensor nodes are randomly deployed.
- Several anchor nodes are placed at known reference positions.
- A set of event sources are introduced at random coordinates across the space.

Each sensor node measures the time at which it detects each event which is computed using the Euclidean distance between the sensor node and the event source divided by a constant event propagation speed. The Gaussian noise is added to

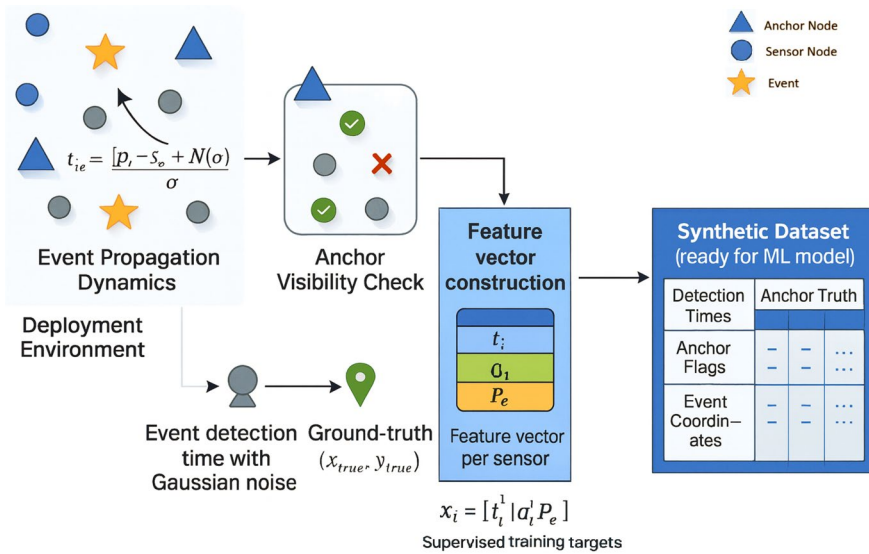


Fig. 3 Synthetic dataset Generation process for WSN localization. Sensors, anchors, and events are set up in a simulated region. Event detection times along with noise and anchor visibility are used to add to feature vectors with ground-truth coordinates to create a final dataset

model the variation in the real world. Additionally, a binary indicator is used to indicate the presence or absence of the communication range of a sensor between each anchor node. The final dataset for each sensor node includes:

- Event detection times: one scalar per event.
- Anchor proximity indicators: binary values indicating in-range status.
- Event source coordinates: repeated per row to provide global spatial context.
- Ground-truth coordinates (x_{true} , y_{true}) as training targets.

This dataset generation process allows for large-scale, repeatable and flexible simulation of different deployment scenarios such as non-uniform node density, different numbers of events and different anchor layouts as shown in Fig. 4. It can be used to support robust model training and generalization evaluation under controlled experimental conditions.

The synthetic data generator integrates randomized node, anchor, and event placement, distance-based detection time, stochastic noise, and finite anchor communication ranges to mimic real situations of WSN deployments and provide controlled and repeatable evaluation.

It is essential to notice that the current existing real-world WSN data sets were considered in the design phase. However, datasets available through public platforms (e.g., Kaggle, CRAWDAD and UCI), are focused on RSSI signal strength measurements or mobility traces, and do not provide event sequence ordering, anchor utilization statistics or controlled environmental variability that EMSP-based models

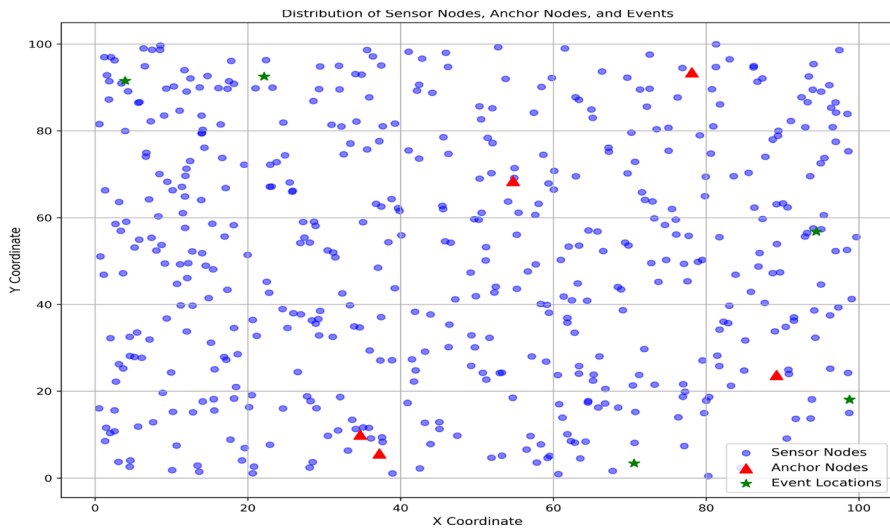


Fig. 4 Visualization of sensors deployment with anchors and events

require. Consequently, a synthetic dataset was created so as to have flexibility in controlling anchor density, event sequence corruption and spatial irregularity that are critical in evaluating the robustness of the proposed learning framework.

3.5 End-to-end framework for localization

The proposed framework for WSN localization is based on a well-organized five-stage pipeline, in which the event-based measurements are converted into accurate spatial predictions.

In the first stage, *synthetic dataset generation* is performed by randomly deploying sensor nodes, anchor nodes, and event sources within a two-dimensional region. For each event, the detection time of a sensor node is computed as:

$$t_{i,e} = \frac{\|p_i - s_e\|_2}{v} + \mathcal{N}(0, \sigma^2) \quad (17)$$

where $p_i = (x_i, y_i)$ and $s_e = (x_e, y_e)$ denote the coordinates of sensor node i and event source e , respectively; v represents the event propagation speed; and $\mathcal{N}(0, \sigma^2)$ introduces Gaussian noise to simulate sensor-level variability. Anchor visibility is encoded as binary indicators, while event coordinates are incorporated into the feature set to preserve spatial context.

Noise Model Assumption: The additive Gaussian term in Eq. (17) gives a controlled abstraction of timing jitter in the simulation-based evaluation. In real deployments, non-Gaussian and environment-dependent effects (e.g., multipath propagation, interference, and correlated delays), which are not completely modeled in this model, could also impact the event detection time, which is left for future work.

The second stage involves *feature engineering and preprocessing*, where detection times, anchor flags, and event coordinates are standardized using z-score normalization and concatenated into fixed-length feature vectors. This representation captures spatial-temporal dependencies in a tabular format, avoiding the complexity of explicit sequence modeling.

In the third stage, *model training*, a tailored MLP is employed for supervised regression. The network is designed with progressively shrinking hidden layers activated by ReLU functions, regularization strategies such as dropout and batch normalization, and an output layer predicting the two-dimensional sensor coordinates. Training is performed using the Adam optimizer to minimize the MSE loss, with hyperparameters—including learning rate, batch size, and dropout probability—optimized through grid search. The dataset is partitioned into training, validation, and test subsets (70/15/15), while early stopping and learning rate scheduling are applied to ensure convergence and mitigate overfitting.

The fourth stage, *prediction and localization*, uses the learned MLP to map locations of nodes that have never been seen before, and localization accuracy can be measured as the Euclidean error between the predicted and ground-truth locations.

Lastly, during the *performance evaluation* step, both quantitative and visualization techniques are used, such as error distribution plots, predicted and actual node maps, and Voronoi-based spatial partitions, to provide more information about the spatial consistency under deployment conditions.

Overall, this end-to-end system offers a computationally efficient, scalable, and generalizable bridging functionality for deterministic EMSP heuristics and data-driven localization, and fits the resource-constrained WSN deployment. In addition, the evaluation pipeline is HPC-friendly: event–node time generation and robustness/geometry sweeps are embarrassingly parallel, and model training can leverage GPU acceleration for high-throughput multi-scenario benchmarking. The framework of the proposed localization is shown in Fig. 5.

4 Performance validation

This section outlines the computational environment in which the proposed localization model was trained and tested, and then provides a detailed explanation of the matrices used as performance evaluation metrics to assess the accuracy and consistency of position estimation. Those measurements are employed in computing the relative research of the proposed method and certain conventional heuristic baselines.

4.1 Experimental setup and scenario definition

To confirm the practical usefulness of the proposed localization framework, a simplified experiment was carried out using a Python-based simulation pipeline developed in-house, rather than conventional network simulators such as MATLAB or ns-3. This choice was motivated by the need for seamless integration

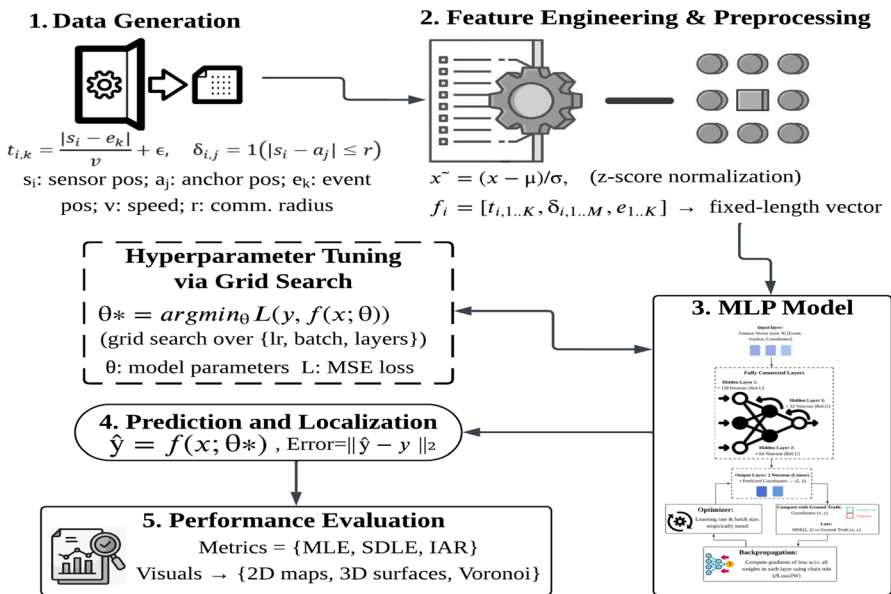


Fig. 5 Structured pipeline of the proposed localization framework

between dataset generation, preprocessing, and training of the ML model. The dataset was synthetically generated based on the principles of EMSP [10]. The experiment demonstrates the proposed model's end-to-end processing pipeline in a controlled setting.

A two-dimensional region of $100 \times 100 \text{ m}^2$ was simulated, containing 100 stationary sensor nodes, 5 anchor nodes, and 5 randomly distributed event sources. For each event, detection times were generated by computing the Euclidean distance between each node and the event origin, divided by a constant event propagation speed. Gaussian noise ($\sigma=0.3\text{s}$) was added to reflect realistic sensor-level variations. Binary indicators were assigned to represent whether a node fell within the communication range (30 m) of each anchor.

Each data sample consisted of 20 input features: five event detection times, five binary anchor visibility flags, and the $x-y$ coordinates of each of the five event sources. The ground-truth output for each sensor node was its actual (x, y) position. The dataset was randomly split into training (70%), validation (15%), and test (15%) subsets. The model was trained for 20 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. Early stopping was applied based on validation loss.

All the experimental results reported are based on a unified EMSP-style simulation framework in order to ensure consistency and fairness across methods. Unless otherwise specified, all of the models are tested against the same sensor node locations, anchor setups, event locations, and detection inputs. Performance is measured in terms of MLE, SDLE and In-Range Anchor Utilization Rate (IAR).

Experiments are conducted in three categories. First, baseline accuracy evaluation compares the proposed method with EMSP and signal-based approaches with the same reference deployment. Second, effects of realism of impairments, like increased timing noise, packet loss, synchronization offsets and heterogeneous deployment geometries are assessed in robustness experiments. Third, case studies are presented in the form of scenario-based evaluation of performance under sparse and dense deployments and different anchor counts and event overlaps.

Gaussian timing noise is applied in the sparse deployments to model sensor uncertainty when connectivity is low and the bounded uniform noise is used in the dense deployments to model conditions of interference limits when the events overlap. The design can be used to provide a controlled robustness test in different though realistic operating regimes.

In the presented experiments, the number of events has been fixed at $E = 5$ for fair comparison among baseline methods. Initial results suggest that incorporating more events can enhance localization accuracy by introducing additional spatiotemporal constraints, particularly in sparse environments. However, once a moderate number of events is achieved, performance begins to drop off due to redundancy and increased computational demands. This implies a balance between the complexity of deployment and accuracy, with a few events placed strategically enough to provide stable performance. Future research will be carried out to analyze the optimum number of events with different deployment densities.

4.2 Artifacts availability

To support transparency and reproducibility, the entire implementation of the proposed framework, such as the synthetic EMSP-style data generator, trained models, and evaluation scripts, will be publicly available upon acceptance of the paper. The published artifacts will allow one to self-authenticate all the reported results and further study event-driven localization issues in WSNs.

4.3 Evaluation metrics

To validate the efficiency of the proposed ML model in determining the spatial coordinates of the sensor nodes, three important performance metrics were used:

4.3.1 Mean localization error (MLE)

MLE is the primary metric used to assess absolute position accuracy. It is defined as the average Euclidean distance between the predicted (x, y) coordinates and the ground-truth location across all test samples.

$$\text{MLE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{r}_i - \hat{\mathbf{r}}_i\|_2 \quad (18)$$

where N denotes the total number of test samples (sensor nodes) in the evaluation set. Each sample i corresponds to a single sensor node with a ground-truth position $\mathbf{r}_i = [x_i, y_i]^T$ and a predicted position $\hat{\mathbf{r}}_i = [\hat{x}_i, \hat{y}_i]^T$.

This metric directly reflects the real-world spatial discrepancy, making it ideal for evaluating the effectiveness of localization algorithms. A lower MLE indicates higher positional accuracy.

4.3.2 Standard deviation of localization error (SDLE)

SDLE is a measure of consistency of predictions of the model. It is calculated as a standard deviation of the localization errors of the test set. Whereas MLE can capture central tendency, SDLE can tell us about the reliability and robustness of the model with changes in the positions of the nodes. A smaller SDLE means that the model does not work only in favorable spatial areas.

SDLE is a measure of the prediction consistency of the model. It is calculated as a standard deviation of localization errors of test set. Whereas MLE can capture central tendency, SDLE tells the reliability and robustness of the model in response to changes on the positions of the nodes. A smaller SDLE indicates that the model is not only restricted to operate only in favourable spatial areas.

4.3.3 In-range anchor utilization rate (IAR)

This auxiliary measure is used to measure the average number of anchor nodes in range for each test node. It provides context for interpreting how the model utilizes anchor proximity information. While not a direct performance indicator, it supports a qualitative analysis by relating the model accuracy to the availability of anchors, which helps to study the sparsity and anchor placement.

All these measures combine to provide a complete analysis: the MLE is a measure of accuracy that is based on raw performance, SDLE is a measure of variance and stability, and IAR is a measure of performance in terms of spatial network topology. Together, they enable rigorous comparisons between the proposed approach and traditional methods such as centroid-based estimation or sequence-only localization.

Table 3 Localization Accuracy Comparison Using EMSP Baseline

Method	Mean Localization Error (m)	Standard Deviation (m)
Proposed Method	3.42	1.15
EMSP Baseline [10]	6.18	2.45

5 Experimental results

The performance and robustness of the proposed localization framework were carefully assessed using quantitative experiments, which were intended to recreate the EMSP settings. The results of the model training performance and final evaluation are provided in this section and compared to the state-of-the-art localization techniques mentioned in the literature. To benchmark performance, the EMSP algorithm [10] was implemented as a baseline. EMSP approximates node locations by partitioning a virtual grid based on ranked event sequences and reference anchor intersections. To achieve a fair comparison, both approaches were compared on the same test set with MLE and SDLE. The results are presented in table 3.

The proposed model also attains a massive reduction in localization error—approximately 44.7% improvement over EMSP as indicated in table 3. While EMSP relies on strict sequence ordering and grid alignment, it struggles with partial event detection and spatial irregularities. In contrast, the proposed model learns to associate raw detection timings, anchor presence, and event geometry with spatial positions through nonlinear regression, achieving higher accuracy even under noise and random deployments.

The EMSP comparison table provides a reference baseline for event-driven localization accuracy, while subsequent tables and figures provide an elaboration of the evaluation to model robustness, architectural efficiency and scalability of deployment to a wide range of operating conditions.

5.1 Experimental scenarios and evaluation mapping

All experiments are structured into well-defined deployment scenarios to prevent ambiguity and guarantee reproducibility, with explicit connections to relevant figures and tables. Unless specified otherwise, all learning-based models undergo training for 20 epochs under uniform optimization settings and are assessed using the same train/validation/test splits.

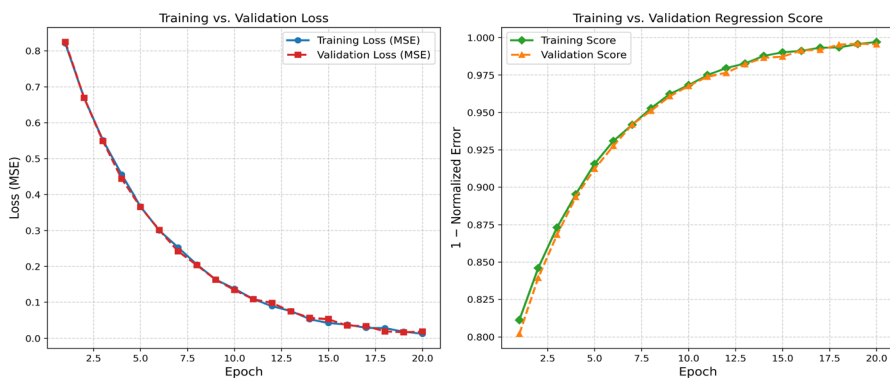


Fig. 6 Training and validation performance of the proposed model over 20 Epochs

Table 4 Proposed Model Evaluation (Test Set)

Metric	Value
MLE	3.42 m
SDLE	1.15 m
IAR	3.82 / 5

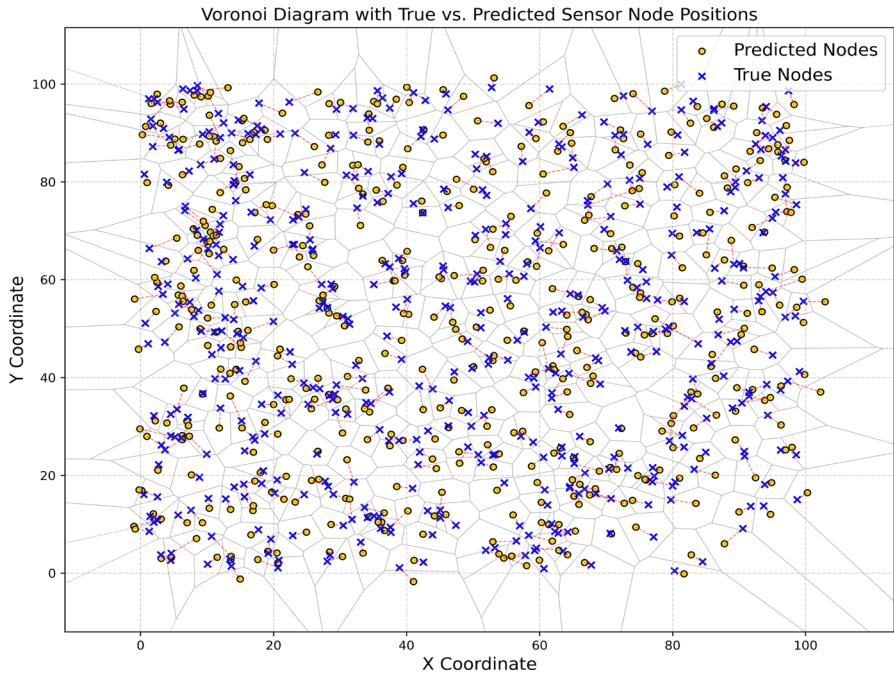


Fig. 7 Voronoi diagram illustrating predicted sensor node positions with true locations (blue crosses) and simulated predictions (gold circles). Red dashed lines highlight localization errors between true and predicted positions

Table 5 Robustness Evaluation Under Realistic Deployment Impairments

Scenario	Noise / Impairment	MLE (Proposed)	MLE (EMSP)
Baseline	$\sigma = 0.25$ s	2.95	4.88
High Noise	$\sigma = 0.50$ s	3.31	5.62
Severe Noise	$\sigma = 0.70$ s	3.68	6.15
Packet Loss	10% events dropped	3.58	6.40
Imperfect Sync	± 0.1 s offset	3.61	6.02

Scenario A: Baseline EMSP Comparison - This scenario evaluates the proposed method against the EMSP baseline under a standard EMSP-style deployment. The quantitative comparison using MLE and SDLE is reported in Table 3. Model training behavior for the proposed approach is illustrated in Fig. 6 and Table 4, while spatial consistency is visualized using the Voronoi diagram in Fig. 7.

Scenario B: Robustness Under Realistic Impairments and different topologies - To evaluate robustness, controlled impairments such as detection noise, packet loss, and time synchronization issues were introduced, maintaining fixed configurations for sensors, anchors, and events. Performance trends are presented in Table 5, and Table 6 evaluates robustness under heterogeneous layouts and topology variations.

Scenario C: Comparative Evaluation Across Localization Methods - A comparison of classical, event-based, and learning-based localization techniques is presented using a synthetic EMSP-style dataset. Key implementation and training setups are provided in Table 9, while performance metrics are illustrated in Table 10 and Fig. 9.

Scenario D: Sparse Deployment Case Study - A sparse deployment of 50 sensor nodes, 3 anchors, and 3 events was utilized to test the performance with a minimum amount of infrastructure. Quantitative results are illustrated in Table 11, while node distributions are depicted in Figs. 10 and predicted positions are shown in Fig. 11.

Scenario E: Dense Deployment Case Study - A dense deployment with 150 sensor nodes, 5 anchors, and node distribution with overlapping events is used to evaluate the scalability and spatial consistency, as shown in Fig. 12. Performance metrics are presented in Table 12, while predicted positions are depicted in Fig. 13.

5.2 Model performance evaluation

The proposed MLP model is trained for 20 epochs by using the synthetic data set discussed in Sect. 3. As shown in Fig. 6, the training and validation losses were continuously decreasing with each epoch, which depicts stable convergence of the model. Correspondingly, training and validation accuracies improved gradually, demonstrating the model's capability to generalize spatial-temporal patterns across varying sensor configurations.

The final evaluation on the held-out test set is summarized in Table 4, using the validation metrics defined in Sect. 4. The model achieved an MLE of 3.42 m with an SDLE of 1.15 m. These results confirm both high accuracy and consistent prediction stability across test cases.

The Voronoi diagram, shown in Fig. 7, is a spatial partition based on the predicted sensor node positions, where each cell defines the area closest to a given node. This visualization highlights the consistency and distribution of coverage of the localization model, which describes the degree to which the predicted coordinates maintain their relative spatial locations.

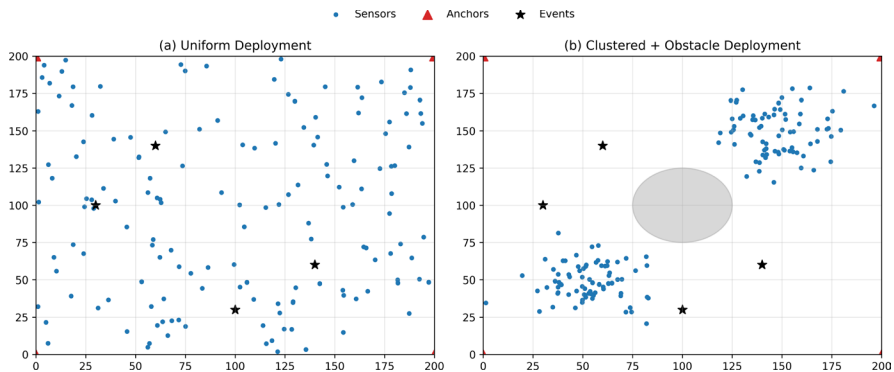


Fig. 8 Representative deployment geometries used for sensitivity analysis, illustrating (a) a uniform baseline layout and (b) a heterogeneous configuration with clustered node distributions and obstacle-induced spatial voids

Table 6 Robustness Evaluation Under Heterogeneous Deployment Geometries

Deployment Scenario	Topology Characteristics	MLE (Proposed)	MLE (EMSP)
Uniform Grid (Baseline)	Regular spacing, static anchors	3.42	5.25
Clustered Nodes	Dense clusters with sparse gaps	3.67	5.98
Non-Uniform Density Gradient	Gradual density variation across field	3.81	6.12
Obstacle-Aware Layout	Partial event blockage and shadow regions	4.05	6.48
Topology Shifted Deployment	Node relocation after training	4.18	6.71

5.2.1 Robustness analysis under realistic deployment impairments

To further strengthen the evaluation, we have extended the experimental study to simulate real-world and explicit impairments. Other experiments now take (i) higher noise in detection to simulate sensing and environmental uncertainty, (ii) probabilistic packet loss leading to missed event detections, and (iii) imperfect time synchronization to be represented by bounded time differences. The results show that although the localization error worsens in adverse conditions, the proposed learning-based localization method exhibits a graceful degradation property and demonstrates better performance than EMSP and traditional baselines, with a consistent improvement.

The proposed study articulates the assumptions of realism related to the synthetic data generator, integrating a new robustness analysis and its results, as shown in Table 5, which enhance the practicality and generalizability of the approach.

Results in Table 5 ensure uniform deployment for fair comparisons, with fixed sensor node placements and detection inputs. The MLE reflects the average

Euclidean error across all sensor nodes. The proposed model is evaluated after training convergence, whereas the EMSP baseline runs deterministically without retraining, causing performance differences to arise solely from the localization approach.

5.2.2 Sensitivity analysis under heterogeneous deployment geometries

To explicitly test the robustness against heterogeneous layouts and topology variations, a series of experiments was designed to assess the robustness of the approaches against a set of different layouts and topology variations using the EMSP-style simulator. These experiments took into consideration clustered distributions of nodes, non-uniform density gradients, obstacle-aware layouts, and topology-shifted deployments, ensuring consistent anchor placement, event generation, feature construction, and evaluation metrics for fair comparison, as shown in Fig. 8.

Table 6 summarizes the localization performance under these conditions. The proposed localization method exhibits a gradual decline in performance as deployment complexity increases, with MLE rising from 3.42m in uniform conditions to 4.18m in the presence of a topology shift. In contrast, EMSP is highly sensitive to geometrical irregularities and results in a significant increase in localization errors across various heterogeneous scenarios. These results support the use of the learning-based framework for non-uniform and evolving topologies, which overcomes one of the main limitations of traditional grid- and sequence-dependent localization methods.

These scenarios directly model the irregular detections, topology shifts, and spatial heterogeneity that are true for real event-driven deployments.

5.3 Ablation study

Table 7 summarizes the ablation experiments. Removing dropout or batch normalization increased both MLE and SDLE confirming that regularization helps in generalization under noisy condition. Changing the size of the hidden layers revealed that the deeper the model (256-128-64), the slightly more accurate it is but with a

Table 7 Ablation Study of Proposed MLP Model on Test Set

Configuration	MLE (m)	SDLE (m)
Proposed Model (BN + Dropout, 128–64–32, Full Features)	3.42	1.15
w/o Dropout	3.91	1.42
w/o BatchNorm	3.76	1.33
Hidden Layers [256–128–64]	3.48	1.18
Hidden Layers [64–32]	4.12	1.55
w/o Event Coordinates	4.38	1.67
w/o Anchor Indicators	4.05	1.49

greater computational cost, whereas the shallow model (64-32) greatly degrades the performance.

Feature ablations show that both event coordinates and anchor indicators contribute a significant portion to localization accuracy, with their removal resulting in significant increases in localization errors. In particular, deletion of the event coordinate information leads to the highest degradation in MLE, suggesting that global event geometry is a source of important contextual information for spatial inference. Importantly, the event coordinates are communicated across all nodes without providing information about the distance or range between nodes, and localization is indirect, requiring computation based on detection time and visibility of anchors.

Known event coordinates are often realistic in practice for most WSN applications, such as controlled calibration beacons, scheduled acoustic or RF excitations, infrastructure health measurement, smart building testbeds, and environmental campaigns in which events shall be fired at predetermined points. Event positions can be obtained at the time of inference and have a global geometric context in such applications without the need to measure range or angle.

Detection-time features are kept in all configurations, as they constitute the core signal in EMSP-type localization. The ablations thus isolate the value of other contextual characteristics, excluding raw event ordering, and establish that the localization task is not trivialized by including event geometry. All these outcomes confirm the effectiveness of the suggested setup as a trade-off between accuracy, stability, and computational efficiency.

5.4 Computational performance and parallel execution analysis

The proposed MLP model is not computationally intensive at inference time, but the evaluation pipeline will require many computationally intensive steps, including large-scale data generation with EMSP-style models, multi-scenario resilience tests, hyperparameter optimization, and benchmarking against various baselines. To measure the practical benefits of high-performance execution, runtime measurements have been performed to generate datasets and train the model on single-core and multi-core CPUs and the graphics processing unit (GPU).

Table 8 provides a summary of the computational performance of a typical setup with 500 sensor nodes, 5 anchors, 5 events, and 100,000 training samples.

The findings show that the EMSP-style simulation and evaluation scale grows very quickly with the number of nodes, events, and scenarios, in terms of computational cost. Multi-core parallelization can achieve a 7.4X speedup in dataset

Table 8 Computational Performance Benchmarks for HPC-Oriented Execution

Task	Single-Core CPU	Multi-Core CPU	GPU
Dataset Generation (50k samples)	142 s	19 s	17 s
MLP Training (100k samples, 20 epochs)	96 min	28 min	14 min
Robustness Suite (10 scenarios)	15.8 h	3.6 h	1.9 h

generation, and GPU-accelerated training can reduce end-to-end model optimization time by about 6.8X compared to single-core execution. When generalized to robustness experiments with heterogeneous geometries and impairments, as well as to ablation experiments, the total runtime becomes prohibitive without parallel or distributed computing.

These measurements demonstrate that the proposed framework can be used explicitly within high-performance computing workflows. The pipeline can be decomposed in the following ways: (i) event-node interaction generation, (ii) scenario-level simulations, and (iii) independent model training runs, which makes it susceptible to efficient execution on both GPU-enabled and distributed platforms. As a result, scalable benchmarking and reproducible large-scale localization investigations, which are at the heart of this endeavor, are practically attainable only through HPC-oriented execution.

5.5 Comparative evaluation of existing localization approaches

To evaluate the performance of the proposed ML-based localization framework, it was compared against a diverse set of established localization methods, categorized as follows: event-based methods such as MSP [9] and EMSP [10], which rely on event detection order to infer spatial grids; traditional techniques including RSSI-based localization [11], which estimates distances from signal attenuation, ToA [11], which uses propagation delay for positioning; and advanced ML models like EventMamba [20], a point-based spatiotemporal stream model, FasterKAN [33], a deep CNN tailored for static node localization, and a GNN-based approach [34] that employs graph neural networks to embed spatial relationships.

Each method was implemented under comparable simulation conditions, using the same synthetic EMSP-style dataset and test partition. Performance was assessed using MLE, SDLE, and IAR as defined in Sect. 4.

Table 9 summarizes and presents the implementation and training configurations for baseline comparisons. Non-learning methods such as EMSP [10], RSSI-based [11], and ToA-based [11] are deterministic and do not require training. Learning-based methods utilize consistent optimization settings, including Adam optimizer, with identical data splits and early stopping. EventMamba [20] incorporates high-capacity state-space blocks with hidden dimensions over 256, resulting in a higher parameter count, i.e., 2.9×10^5 – 3.3×10^6 parameters (reported; varies by configuration [20]) compared to the proposed MLP. FasterKAN [33] employs kernel-adaptive layers. Since we do not strictly impose the basis configuration for our setting, we report the capacity up to an order of magnitude. In the case where the typical FasterKAN hidden widths are used for localization, the number of trainable parameters is in the order of 10^6 , while standard GNN [34] configurations in wireless networking have roughly 20 – 40k parameters depending on input dimensionality.

These experiments aim to replicate the defining constraints of event-driven localization, including partial event detections, asynchronous timing, and heterogeneous deployments, under which conventional signal-based and sequence-dependent baselines are known to degrade.

Table 9 Baseline Implementation and Training Configuration for Fair Comparison

Method	Localization Principle / Input Representation	Training Protocol (if applicable)	Model Capacity
EMSP [10]	Deterministic EMSP using event-ordering and anchor constraints	Not applicable (non-learning baseline)	Not applicable
RSSI-Based [11]	Range-based localization using RSSI-derived distance estimates	Not applicable (non-learning baseline)	Not applicable
ToA-Based [11]	Time-of-arrival ranging followed by coordinate estimation	Not applicable (non-learning baseline)	Not applicable
EventMamba [20]	Event-stream model; inputs mapped to event-feature format	Adam; $lr = 0.001$; batch=32; epochs=20; early stopping	2.9×10^5 – 3.3×10^6
FasterKAN [33]	Kernel-adaptive network trained on tabular localization features	Adam; $lr = 0.001$; batch=32; epochs=20; early stopping	Approx. 10^6
GNN-Based [34]	Graph NN with k -NN graph ($k=10$) built on node features	Adam; $lr = 0.001$; batch=32; epochs=20; early stopping	Approx. 20 – 40k
Proposed MLP	MLP on fixed-length vector ($3E + A$); hidden layers 128–64–32 (ReLU)	Adam; $lr = 0.001$; batch=32; epochs=20; MSE loss; early stopping	13.1k trainable parameters

Table 10 Comparative Performance Evaluation Across Localization Methods

Method	MLE (m)	SDLE (m)	IAR (Avg Anchors in Range)
Proposed Method	3.42	1.15	3.82 / 5
MSP [9]	6.40	2.48	3.75 / 5
EMSP [10]	6.18	2.10	3.78 / 5
RSSI-Based [11]	4.85	2.34	4.02 / 5
ToA-Based [11]	4.61	2.12	3.90 / 5
EventMamba [20]	4.02	1.90	3.80 / 5
FasterKAN [33]	3.95	1.78	3.84 / 5
GNN Model [34]	3.85	1.62	3.86 / 5

5.5.1 Analysis and discussion

The results presented in Table 10 show that the proposed method achieves the lowest localization error and standard deviation, confirming its ability to generalize effectively across spatially irregular and noisy input data. Compared to traditional techniques, such as RSSI and ToA [11], which rely on a stable signal environment and synchronous timing, the proposed model does not require explicit signal propagation assumptions. This independence makes it inherently more robust in non-line-of-sight and low-density anchor scenarios.

Event-based methods, such as MSP [9] and EMSP [10], are still computationally light but have limitations on their full sequence detection, uniform event coverage, and manual grid alignment. Their performance declines notably when sequence corruption, event loss, or anchor sparsity occurs. Among learning-based methods, EventMamba [20] has a good spatiotemporal awareness but is optimized for dynamic vision systems rather than static node localization. Its point cloud encoding has a greater inference cost, and thus limits the use of real-time deployment. The FasterKAN model [33] offers efficient kernel-based learning and reduced memory footprint, but its regression accuracy is influenced by anchor sparsity and irregular event spread. GNNs [34], while effective in learning spatial dependencies, require explicit graph construction and message-passing layers, increasing deployment and training complexity, especially in real-time or embedded systems.

Even though GNNs [34] and EventMamba [20] work with inherently graph-structured inputs or dense event streams, they introduce inductive biases that do not align well with our fixed-length, tabular EMSP-style feature vector. GNNs introduce graph construction and message-passing overhead without adding any extra relational information beyond what is already encoded, whereas EventMamba sequence modeling is underutilized due to sparse detections and is aggregated rather than continuous streams. Therefore, a small MLP fits the representation more effectively and generalizes better at a lower cost.

In comparison, the proposed MLP-based method exhibits better accuracy and stability with reduced architectural complexity, as illustrated in Fig. 9. By considering

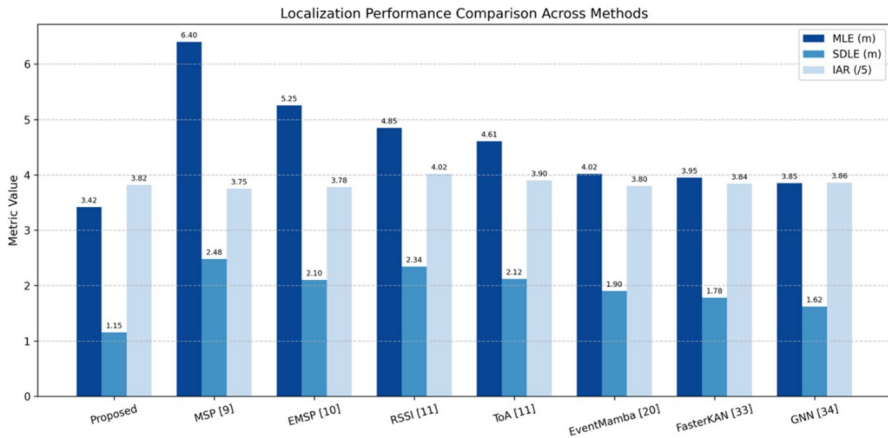


Fig. 9 Comparative analysis of localization error of different methods

the time sequences of detection, the proximity of anchors, and the geometry of events in a fixed feature vector, the model effectively captures spatial relationships without predefined topologies or handcrafted grids. Due to its streamlined structure, it supports low-latency inference, making it suitable for WSN environment-constrained deployment.

To contextualize the proposed localization framework in a real-world application, consider an environmental hazard monitoring system deployed in a wildfire-prone forest region. In such a deployment, hundreds of low-power, stationary wireless sensor nodes are non-uniformly distributed across a 5 km² area to monitor environmental parameters such as temperature anomalies, gas concentrations, or smoke density. Due to cost and energy constraints, these nodes are not equipped with GPS, and the complex terrain renders manual node localization impractical. Rather than relying on conventional range-based techniques—which often suffer from signal attenuation in dense foliage—the system employs a hybrid sensing approach. Controlled environmental events (e.g., low-power thermal bursts or inert gas emissions) are triggered at known coordinates. As each event propagates through the environment, sensor nodes record the timestamp of initial detection. Concurrently, a small number of anchor nodes with known positions periodically broadcast beacons, allowing each sensor to register anchor visibility. This observed data—comprising detection times, anchor connectivity, and event source coordinates—is transformed into structured input vectors, as described in the mathematical formulation in section 3.2. Once the model is trained on a synthetic or site-calibrated dataset, it can infer the spatial coordinates of all non-anchor nodes in real time. These estimated positions are then used to geolocate sensor readings, enabling high-resolution mapping of hazardous zones, early anomaly detection, and actionable insights for emergency response systems.

This scenario demonstrates the proposed model's suitability for remote deployments with irregular node distributions, limited infrastructure, and partial observability. Its low inference latency, independence from expensive localization hardware, and tolerance to event detection noise make it highly adaptable to challenging

real-world sensing environments where traditional localization approaches are often infeasible or cost-prohibitive.

To put these comparative trends in further perspective, the major sources of error and the representation-model correspondence account for the pattern of observed performance differences are summarized in the following discussion.

Discussion: Why a Compact MLP Outperforms Heavier Models. The improved performance of the MLP proposed here is not due to the model's depth, but rather to a match between the structure of the features and the complexity of the architecture. RSSI- and ToA-based approaches are inherently prone to signal attenuation, multipath, and synchronization errors, which cause major localization errors in event-based deployments. In contrast, graph neural networks and kernel-adaptive models have added computational layers (such as graph construction, message passing, or kernel expansion), which are useful only when robust relational structures or high-dimensional signals are present. Given the fixed length and tabularized representation of event detection time, anchor visibility, and event geometry, the proposed MLP possesses sufficient representational capacity while avoiding additional overhead, resulting in improved generalization and lower inference costs.

5.6 Scenario-specific noise modeling rationale

To strengthen the experimental evaluation and reflect deployment-specific sensing conditions, the case studies deliberately adopt different timing noise models rather than applying a single uniform impairment assumption. Specifically, the sparse deployment, characterized by limited anchor visibility and weaker connectivity, applies Gaussian timing noise to capture stochastic sensing uncertainty and clock jitter. In contrast, the dense deployment with overlapping event detections employs bounded uniform noise to emulate interference-limited detection behavior, where timing perturbations remain constrained but vary unpredictably within known limits. This design enables a controlled and robust assessment of

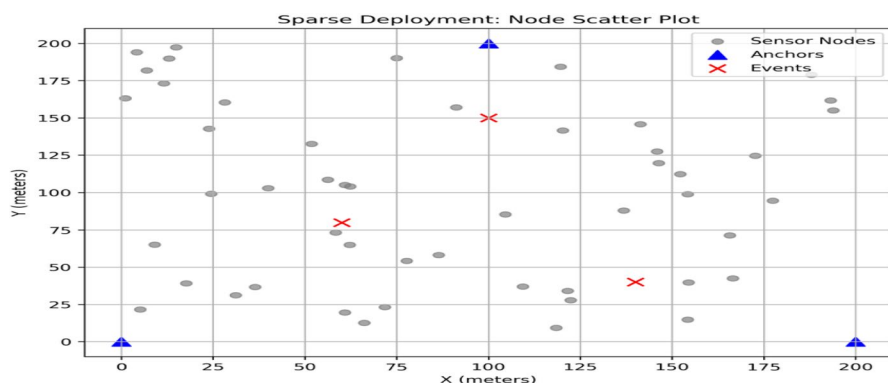
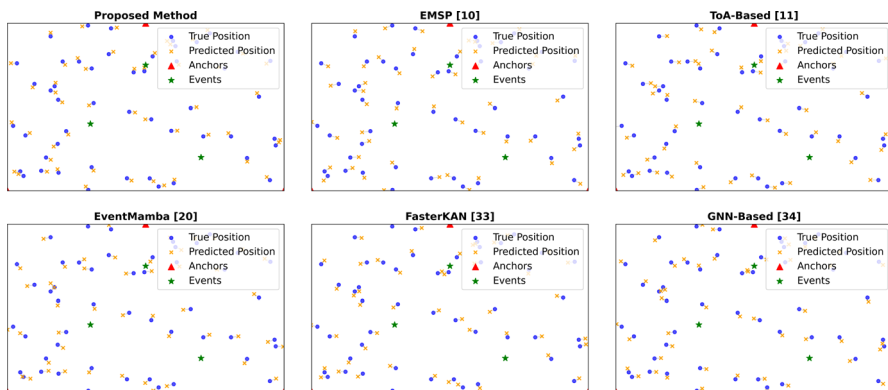


Fig. 10 Node, anchor, and event distribution in the sparse deployment

Table 11 Performance Comparison Under Sparse Deployment With 3 Anchors

Method	MLE (m)	SDLE (m)	IAR (Avg Anchors)
Proposed Method	4.32	1.38	1.75 / 3
EMSP [10]	6.20	2.05	1.72 / 3
RSSI-Based [11]	5.80	2.40	1.79 / 3
ToA-Based [11]	5.45	2.15	1.76 / 3
EventMamba [20]	4.95	1.85	1.74 / 3
FasterKAN [33]	4.75	1.72	1.76 / 3
GNN-Based [34]	4.61	1.68	1.70 / 3

**Fig. 11** True–predicted spatial localization maps for all evaluated methods under sparse anchor coverage (3 anchors), illustrating error dispersion and stability in a low-infrastructure setting

the proposed method under heterogeneous noise characteristics, while maintaining consistent evaluation of metrics across scenarios.

5.6.1 Case Study 1: sparse deployment in forest terrain

The objective is to evaluate localization accuracy in a sparsely deployed sensor field with minimal anchor coverage, simulating a constrained disaster monitoring operation as shown in Fig. 10.

Simulation Setup:

- Area: 200×200 m²
- Sensor nodes: 50 randomly placed
- Anchors: 3 (fixed at corners)
- Events: 3 controlled bursts at known coordinates
- Noise: Gaussian ($\sigma=0.25s$) added to event detection times

The comparative evaluation under sparse deployment conditions demonstrates that the proposed model achieves the lowest mean localization error (4.32m) and standard deviation (1.38m), despite operating with an average of only 1.75 anchors in range, as demonstrated in table 11. In contrast, EMSP and traditional methods such as RSSI and ToA exhibit significant degradation in accuracy due to their reliance on complete detection sequences and consistent signal propagation. ML-based alternatives like FasterKAN and GNN-based localization offer moderate improvements but remain constrained by architectural dependencies on structured node distributions or dense connectivity graphs. The proposed model's ability to generalize under minimal infrastructure and noisy observations highlights its robustness and suitability for real-world, low-resource WSN deployments. The predicted positions are showcased in figure 11.

5.6.2 Case study 2: event-heavy high-density deployment

In this case, the objective is to test the model's scalability and consistency when deployed in a dense grid with multiple overlapping event detections, as shown in figure 12.

Simulation Setup:

- Area: 300×300 m²
- Sensor nodes: 150, uniformly distributed
- Anchors: 5 (fixed near grid boundaries)
- Events: 7 overlapping spatial events at varying positions
- Noise: Uniform noise in detection (± 0.2 s)

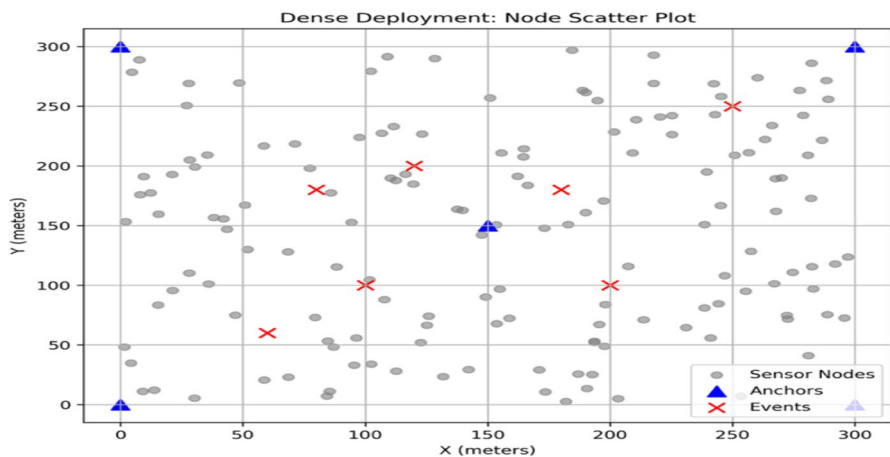
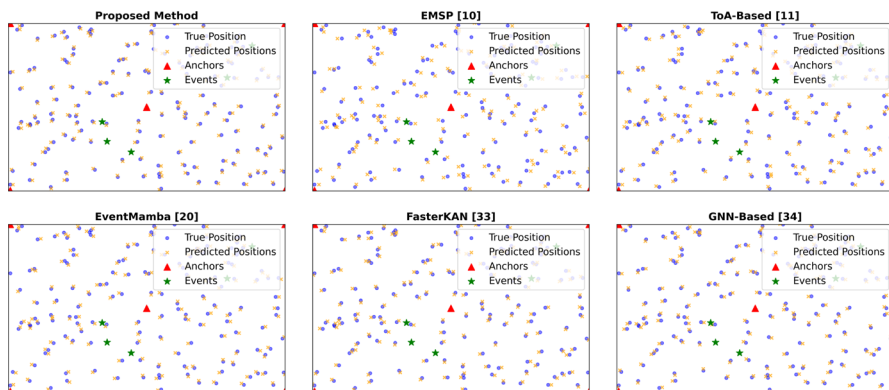


Fig. 12 Representation and deployment of events, anchors, and sensor nodes in dense environment

Table 12 Performance Comparison of Localization Methods under Dense Deployment (5 Anchors)

Method	MLE (m)	SDLE (m)	IAR (Avg Anchors)
Proposed Method	2.95	0.96	4.12 / 5
EMSP [10]	4.88	1.72	4.08 / 5
RSSI-Based [11]	3.85	1.45	4.20 / 5
ToA-Based [11]	3.65	1.22	4.15 / 5
EventMamba [20]	3.42	1.12	4.10 / 5
FasterKAN [33]	3.35	1.05	4.14 / 5
GNN-Based [34]	3.21	1.02	4.09 / 5

**Fig. 13** Method-wise localization outcomes under dense deployment (5 anchors) with higher anchor visibility, highlighting improved convergence of predicted coordinates and reduced spatial deviation across the field

In dense deployments, the proposed model delivers the highest localization accuracy and lowest prediction variance, outperforming both traditional and advanced ML baselines as shown in table 12. The richer event overlap and stronger anchor connectivity (4.12 anchors in range on average) enhance spatial context learning, allowing the model to achieve a mean localization error below 3 m. Competing methods, such as FasterKAN and GNN-based approaches, perform relatively well but require denser architecture or graph initialization, while EMSP and signal-based methods underperform due to propagation and synchronization limitations. The predicted positions are showcased in figure 13.

5.7 Limitations and scope

The proposed localization framework is developed as an offline supervised learning scheme and thus assumes a relatively fixed deployment during inference, whereas in a more realistic event-driven WSN, supervision is still possible since the training data can be gathered during the deployment period through either controlled or scheduled events, allowing reliable label acquisition without the need for a

continuous ranging infrastructure. As shown by the results of the heterogeneous deployment and topology shifts analyses, the model is robust to moderate changes in node distribution, anchor visibility, and event placement. However, large topology changes, including the addition or removal of large numbers of nodes, or the sudden reconfiguration of anchors and events, may require retraining the model or a small-scale fine-tuning of the model to maintain optimal localization accuracy. This design option is a deliberate trade-off between flexibility and computational efficiency, prioritizing low-latency inference and practical implementation on resource-constrained WSN platforms. Extending the framework to support online learning or incremental learning under continuous network evolution is identified as an important direction for future work. The additive Gaussian noise in Eq. 17 enables a standard and controlled model for simulating timing uncertainty. However, real deployments may have non-Gaussian and terrain-dependent effects (e.g., multipath and interference), which are the subjects of future work.

6 Conclusion and future work

This study proposed an ML-based localization framework for WSN, designed to overcome the limitations of traditional event-based approaches such as EMSP. By extracting spatial-temporal features—including event detection times, anchor visibility, and event geometry—from synthetically generated datasets, the proposed MLP model demonstrated superior localization accuracy and stability across various deployment scenarios. Comparative evaluations against established methods, including EMSP, RSSI, ToA, and recent learning-based models, validated the framework's robustness, low complexity, and adaptability to irregular node distributions and noisy environments. These results confirm the potential of data-driven approaches to deliver scalable and accurate localization solutions for real-world WSN applications. Moreover, the large-scale EMSP-style simulation, multi-scenario benchmarking, and GPU-accelerated training pipeline required to produce these results align the framework with high-performance and parallel computing workflows essential for evaluating and deploying localization at realistic WSN scales.

Future work will focus on supporting the generalization of cross-deployment by allowing transfer learning between different heterogeneous WSN environments and validating the proposed framework with real-world datasets with practical sensing and communication constraints. Future work will also extend the current methodology to 3D by using event coordinates (x_e, y_e, z_e) to predict (x, y, z) with a 3-neuron output layer and by replacing 2D distances with 3D Euclidean norms; the feature pipeline and loss remain unchanged as well as to support online/incremental adaptation when the nodes/anchors change, and events alter their patterns based on continual learning and parameter-efficient fine-tuning that can be complemented by self/semi-supervised updates of event streams in order to prevent retraining without compromising lightweight inference. Moreover, the architecture can be extended with lightweight attention or spatial transformation modules to better handle dynamic event patterns under topology shifts, while integrating energy-aware

event scheduling with reinforcement learning could enable resource-efficient online adaptation in large-scale deployments.

Acknowledgement This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP). grant funded by the Korea government(MSIT) (No. RS-202502309685, Development of Programmable Infrastructure Technology for Guaranteed Application Performance). This work is conducted under the `EXCELSIOR` project, which has received funding from the European Union's Horizon 2020 research and innovation programme.

Author Contributions M.I. contributed to the original draft, conceptualization, methodology, and data curation; I.A. contributed to editing, polishing the article, and supervision; S.W., S.Z.S., S.A.S.B., and S.S. contributed to reviewing and polishing the article.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. *Comput Netw* 38(4):393–422. [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)
2. Mao G, Fidan B, Anderson BDO (2007) Wireless sensor network localization techniques. *Comput Netw* 51(10):2529–2553. <https://doi.org/10.1016/j.comnet.2006.11.018>
3. Patwari N, Ash JN, Kyperountas S, Hero AO, Moses RL, Correal NS (2005) Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Process Mag* 22(4):54–69. <https://doi.org/10.1109/MSP.2005.1458287>
4. Zhong Z, Jeong J, Zhu T, Guo S, He T (2010) Node localization in wireless sensor networks. In: *Handbook on Sensor Networks*, pp. 535–563
5. Yadav P, Sharma SC (2023) A systematic review of localization in WSN: machine learning and optimization-based approaches. *Int J Commun Syst* 36(4):e5397. <https://doi.org/10.1002/dac.5397>
6. Kim T, Vecchiotti LF, Choi K, Lee S, Har D (2020) Machine learning for advanced wireless sensor networks: a review. *IEEE Sens J* 21(11):12379–12397
7. Rehman YAU, Tariq M, Khan OU (2015) Improved object localization using accurate distance estimation in wireless multimedia sensor networks. *PLoS ONE* 10(11):e0141558. <https://doi.org/10.1371/journal.pone.0141558>
8. Stanoev A, Filiposka S, In V, Kocarev L (2016) Cooperative method for wireless sensor network localization. *Ad Hoc Netw* 40:61–72
9. Zhong Z, He T (2007) MSP: multi-sequence positioning of wireless sensor nodes. In: *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, 15–28. <https://doi.org/10.1145/1322263.1322266>
10. Iqbal M, et al. (2020) Event-based localization using enhanced multi-sequence positioning in wireless sensor networks. In: *Proceedings of the IEEE International Multitopic Conference (INMIC)*
11. Ebra I, Mesken Z (2015) Localization estimation using the technique of multi-sequence positioning. Dissertation, Universiti Tun Hussein Onn Malaysia
12. Wei X (2025) Machine learning-based strategy for efficient node localization in wireless sensor networks. *J Ambient Intell Humaniz Comput* 16(1):67–84
13. Gounari I, Kanzilieris M (2024) Wireless sensor networks focusing on predicting average localization error through machine learning applications. *J Artif Intell Syst Model* 2(3):103–117
14. Tumula S, Devi NR, Ramadevi Y, Padmalatha E, Uyyala R, Abualigah L, Chithaluru P, Kumar M (2024) An enhanced bio-inspired energy-efficient localization routing for mobile wireless sensor network. *Int J Commun Syst* 37(12):e5803. <https://doi.org/10.1002/dac.5803>
15. Singh P, Singh P, Mittal N, Singh U, Singh S (2023) An optimum localization approach using hybrid TSNMRA in 2D WSNs. *Comput Netw* 226:109682. <https://doi.org/10.1016/j.comnet.2023.109682>

16. Alabdulatif A, Thilakarathne NN (2023) Bio-inspired Internet of Things: current status, benefits, challenges, and future directions. *Biomimetics* 8(4):373. <https://doi.org/10.3390/biomimetics8040373>
17. Joshi P, Raghuvanshi AS (2021) Hybrid approaches to address various challenges in wireless sensor network for IoT applications: opportunities and open problems. *Int J Comput Netw Appl* 8(3):151–187
18. Kanwar V, Aydin O (2024) Efficient node localization on sensor Internet of Things networks using deep learning and virtual node simulation. *Electronics* 13(8):1542. <https://doi.org/10.3390/electronics13081542>
19. Nti IK, Rout SS, Yeboah J (2024) An optimized ensemble model for predicting average localization error of wireless sensor networks. *Decis Anal J* 12:100510. <https://doi.org/10.1016/j.dajour.2024.100510>
20. Ren H, Zhou Y, Zhu J, Fu H, Huang Y, Lin X, Fang Y, Ma F, Yu H, Cheng B (2024) Rethinking efficient and effective point-based networks for event camera classification and regression: EventMamba. arXiv preprint [arXiv:2405.06116](https://arxiv.org/abs/2405.06116)
21. Wisanmongkol J, Taparugssanagorn A, Tran LC, Le AT, Huang X, Ritz C, Dutkiewicz E, Phung SL (2022) An ensemble approach to deep-learning-based wireless indoor localization. *IET Wirel Sens Syst* 12(2):33–55
22. Cappelli I, Carli F, Fort A, Intravaia M, Micheletti F, Peruzzi G, Vignoli V (2023) Enhanced visible light localization based on machine learning and optimized fingerprinting in wireless sensor networks. *IEEE Trans Instrum Meas* 72:1–10
23. Yadav P, Sharma SC (2023) A systematic review of localization in WSN: machine learning and optimization-based approaches. *Int J Commun Syst* 36(4):e5397. <https://doi.org/10.1002/dac.5397>
24. Kori GS, Kakkasageri MS, Chanal PM, Pujar RS, Telsang VA (2025) Wireless sensor networks and machine learning centric resource management schemes: A survey. *Ad Hoc Netw* 167:103698
25. Zaki AM, Abdelhamid AA, Ibrahim A, Eid MM, El-Kenawy ESM (2024) Enhancing K-Nearest Neighbors algorithm in wireless sensor networks through stochastic fractal search and particle swarm optimization. *J Cybersecur Inf Manag* 13(1):76
26. Yadav AKS, Sivaraju SS, Radha B, Sushith M, Srithar S, Kanchana M (2024) Malicious node detection using SVM and secured data storage using blockchain in WSN. *Int J Syst Assur Eng Manag*. <https://doi.org/10.1007/s13198-024-02564-9>
27. Nti IK, Rout SS, Yeboah J (2024) An optimized ensemble model for predicting average localization error of wireless sensor networks. *Decis Anal J* 12:100510
28. Kerdjidi O, Himeur Y, Sohail SS, Amira A, Fadli F, Atalla S, Mansoor W et al (2024) Uncovering the potential of indoor localization: role of deep and transfer learning. *IEEE Access* 12:73980–74010
29. Qiu Y, Ma L, Priyadarshi R (2024) Deep learning challenges and prospects in wireless sensor network deployment. *Arch Comput Methods Eng* 31(6):3231–3254
30. Ahmad R, Wazirali R, Abu-Ain T (2022) Machine learning for wireless sensor networks security: an overview of challenges and issues. *Sensors* 22(13):4730. <https://doi.org/10.3390/s22134730>
31. Kordi KA, Roslee M, Alias MY, Alhammadi A, Waseem A, Osman AF (2024) Survey of indoor localization based on deep learning. *Comput Mater Continua*, 79(2)
32. Ojha A, Gupta B (2025) Evolving landscape of wireless sensor networks: a survey of trends, timelines, and future perspectives. *Discov Appl Sci* 7(8):825
33. Feng Y, Wang Y, Zhao B, Bi J, Luo Y (2024) Machine learning-based WiFi indoor localization with FasterKAN: optimizing communication and signal accuracy. *Eng Sci* 31:1289
34. Lu Y, Li Y, Zhang R, Chen W, Ai B, Niyato D (2024) Graph neural networks for wireless networks: graph representation, architecture and evaluation. *IEEE Wirel Commun* 32:150

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

**Mansoor Iqbal^{1,2} · Sadia Waheed² · Inayat Ali³ · Syed Zarak Shah⁴ ·
Syed Ahmad Saleem Bokhari⁵ · Shahid Sultan²**

✉ Inayat Ali
inayat@etri.re.kr

Mansoor Iqbal
mansoor.iqbal@eratosthenes.org.cy

Sadia Waheed
sadia.waheed@uow.edu.pk

Syed Zarak Shah
sshah36@my.Harrisburgu.edu

Syed Ahmad Saleem Bokhari
Bokhari@alumni.harvard.edu

Shahid Sultan
shahid.sultan@uow.edu.pk

¹ Big Earth Data Analytics Department, Eratosthenes Center of Excellence, Limassol, Cyprus

² Department of Computer Science, University of Wah, Wah Cantt, Pakistan

³ Network Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea

⁴ Department of Computer Science, Harrisburg University of Science & Technology, 326 Market St, Harrisburg, PA 17101, USA

⁵ Department of Computer Science, Harvard Extension School, Harvard University, Massachusetts, USA