

sGprAle: Self-Supervised Gaussian Process Regression for Average Localisation Error Estimation in Wireless Networks

Abhilash Singh , *Member, IEEE*, Jaiprakash Nagar , *Member, IEEE*, J. Amutha ,
and Sandeep Sharma 

Abstract—This paper presents sGprAle, a self-supervised framework for predicting Average Localisation Error (ALE) in wireless sensor networks under limited labeled data conditions. The proposed approach integrates Gaussian Process Regression (GPR) with an autoencoder-based feature learning strategy to enhance robustness, generalisation, and predictive reliability. By learning task-agnostic latent representations from unlabeled data through a reconstruction objective, sGprAle enables the GPR model to operate in a compact and informative latent space using only a small labeled subset. Bayesian optimisation is employed to adapt kernel hyperparameters, ensuring calibrated uncertainty estimates and stable performance when labeled data are scarce. The framework is systematically evaluated across multiple labeled-data scenarios using ablation studies, uncertainty analysis, and statistical comparisons. Results demonstrate that the self-supervised latent representation substantially improves model robustness and label efficiency, allowing sGprAle to achieve competitive accuracy relative to existing methods while requiring significantly fewer labeled samples. These findings highlight the effectiveness of combining self-supervised representation learning with probabilistic regression for addressing the node localisation problem. The proposed model offers a reliable and energy-efficient solution for ALE prediction, contributing to improved localisation performance and operational efficiency in wireless sensor networks.

Index Terms—Self-supervised learning, Gaussian process regression, Bayesian optimisation, average localisation error, synthetic data.

I. INTRODUCTION

WIRELESS networks have emerged as an important technology for various applications, including earth and environmental monitoring, healthcare, and military operations [1], [2], [3]. A critical aspect of a wireless network is node localisation, which determines the geographical positions of sensor nodes [4]. Accurate node localisation is essential for the efficient functioning of these networks, particularly in scenarios requiring precise data collection and analysis. However, achieving high localisation accuracy remains challenging due to wireless sensor networks' inherent uncertainties and dynamic nature [5], [6].

Predicting Average Localisation Error (ALE) is a fundamental task in assessing and improving the accuracy of node localisation algorithms in wireless sensor networks [7], [8]. Traditional methods often rely heavily on large amounts of labeled data, which can be costly and impractical to obtain in real-world scenarios [9], [10]. This challenge can be effectively mitigated by using synthetic datasets to simulate various scenarios and conditions, along with machine learning models that can predict ALE effectively using minimal labeled data.

Previous studies have explored various machine learning and statistical approaches for ALE prediction [11], [12], [13], [14]. While these methods have shown promise, they often suffer from limitations such as high computational complexity, sensitivity to input variations, and the requirement for extensive labeled datasets. These challenges highlight the need for more robust and efficient models that can maximise performance even with minimal labeled data.

In this study, we propose the sGprAle model, a self-supervised Gaussian Process Regression (GPR) approach for ALE prediction. Our model integrates feature extraction from an autoencoder, allowing it to utilise minimal labeled data while maintaining high predictive accuracy [15], [16], [17]. The autoencoder extracts meaningful features from the unlabeled data, which are then used by the GPR model to predict ALE using minimum labeled datasets. The key contributions of this paper are as follows:

- Introduction of a self-supervised GPR model that significantly reduces the reliance on labeled data for ALE prediction.

Received 28 April 2025; revised 12 February 2026; accepted 25 May 2026. Date of publication 8 June 2026; date of current version 8 September 2026. (*Corresponding authors: Abhilash Singh; Sandeep Sharma.*)

Abhilash Singh is with the School of Mathematics and Computation, University of Leeds, LS2 9JT Leeds, U.K. (e-mail: abhilash.singh@ieee.org).

Jaiprakash Nagar is with the Eurecom, 06410 Biot, France (e-mail: jaiprakash.nagar@eurecom.fr).

J. Amutha is with the Department of Artificial Intelligence and Data Science, E.G.S. Pillay Engineering College, Nagapattinam 611002, India (e-mail: amutha@egspec.org).

Sandeep Sharma is with the Department of Higher and Technical Education, Government of Jharkhand, Ranchi 834001, India (e-mail: sandeepsvce@gmail.com).

Digital Object Identifier 10.1109/TETC.2026.3699235

- Framework for synthetic data generation and utilisation to overcome the limitations and high costs of obtaining real-world data, enabling the simulation of various scenarios and conditions.
- Comprehensive evaluation of the model across various scenarios, including different percentages of labeled data (10% to 60%), to assess its robustness and performance.
- Systematic uncertainty analysis to evaluate the model's stability and reliability, revealing insights into its sensitivity to input variations.

The rest of the paper is organised as follows: Section II provides a detailed overview of related work in the field of ALE prediction in WSNs and self-supervised learning. Section III describes the datasets, methodology, including the architecture of the sGprAle model and the experimental setup. Section IV presents the results of our extensive experiments and analyses, followed by a discussion. Finally, Section V concludes the paper and outlines directions for future research.

II. RELATED WORK

The traditional localisation techniques, such as Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), and Angle of Arrival (AoA), have been widely used [18], [19], [20]. However, these methods often suffer from limitations related to accuracy, environmental interferences, and computational complexity [21]. In recent years, machine learning approaches have been increasingly adopted to address the localisation problem in WSNs [22]. Various models, including neural networks, support vector machines, and decision trees, have demonstrated improved localisation accuracy [23], [24], [25]. In particular, GPR offers a promising alternative due to its ability to model spatial data and provide uncertainty estimates. Several studies have applied GPR to wireless sensor network problems, highlighting its potential for accurate and robust predictions [26]. However, these methods typically require large amounts of labeled data for training, which can be impractical to obtain in real-world scenarios [27].

The use of synthetic data has become increasingly prevalent in WSN research, addressing the high cost and limited availability of real-world data [28], [29], [30]. Synthetic datasets allow researchers to simulate various WSN scenarios and conditions, providing valuable insights into model performance. Notable works have successfully generated synthetic data to evaluate and improve localisation models. Singh et al. [13] generated the first and only synthetic data that is widely used for studying sensor node localisation. They employed swarm intelligence to create the ALE datasets, considering anchor nodes, transmission range, node density, and iterations as input features. This process is detailed in Section III-A. This study introduces a machine learning approach to optimise network parameters for minimising ALE in wireless sensor networks. The authors propose three variations of Support Vector Regression (SVR): S-SVR (Scale-SVR), Z-SVR (Z-score-SVR), and R-SVR (Range-SVR), each employing different feature standardisation techniques to enhance prediction accuracy and efficiency. The model uses all four features extracted from simulations. Among these methods,

R-SVR stands out as the best algorithm, demonstrating superior performance in ALE prediction. This work highlights the potential of SVR-based models for precise and reliable localisation in WSNs. The authors also evaluated corresponding variants of the GPR models on the same dataset, finding that S-GPR (Scaling-GPR) emerged as the best algorithm. However, a significant limitation of both R-SVR and S-GPR is their requirement for 70% of the labeled data for training the model.

Chen et al. in [12] propose a Residual Tensor Train (ResTT) model for estimating ALE in wireless sensor networks. This paper introduces a quantum-inspired model, ResTT, which captures complex feature interactions in high-dimensional data. They have evaluated the model performance on the same ALE datasets by considering all four features. By leveraging tensor network principles, ResTT effectively addresses the challenges faced by traditional models and demonstrates superior performance in estimating ALE.

In a recent study, Altay et al. in [11] utilised the same ALE datasets and considered all four features to develop robust ML models for ALE prediction. Various machine learning algorithms, including linear regression, support vector machine with different kernels, GPR, and ensemble Least-Squares Boosting (LSBoost), have been evaluated on these datasets. Notably, the ensemble LSBoost method has shown superior performance in ALE prediction, outperforming other techniques and providing a viable strategy for accurate and efficient localisation in wireless sensor networks. They also confirm the observation made by Singh et al. [13] that the normalised version of the data is giving more accuracy.

Recent work by Ileri [14] addresses the critical issue of node localisation by accurately predicting localisation errors, leading to improved network performance, optimised resource allocation, and cost reduction. The author employs the LightGBM algorithm, enhanced with a Particle Swarm Optimisation (PSO) methodology to boost predictive accuracy. They utilise the same ALE datasets. The proposed LightGBM-PSO model is rigorously assessed against various other machine learning methods. The results indicate that the LightGBM-PSO model significantly outperforms other approaches. The feature importance analysis reveals that the number of iterations significantly influences ALE estimation. In contrast, the transmission range has the least impact, which is consistent with the observation made by Singh et al. [13]. However, a significant limitation of this study is the requirement for a substantial amount of labeled data to achieve optimal performance.

More recently, Yu and Yu [31] proposes an improved RSSI-based weighted centroid localization framework that combines signal filtering, weight correction, and swarm-intelligence optimization to reduce positioning error and energy consumption in WSNs. While effective in improving deterministic localization accuracy, the method remains algorithmic and range-dependent, providing point estimates without probabilistic uncertainty quantification or label-efficient learning.

Despite recent advances, existing ALE predictors still face practical limitations, most notably a dependence on large labeled datasets, sensitivity to noise, and computational inefficiency. While hybrid approaches combining autoencoders with

Gaussian Process Regression (GPR) have been explored in other domains, such as the probabilistic surrogate modeling framework proposed by Deshpande et al. [32], these methods primarily employ autoencoders for dimensionality reduction on fully labeled, physics-driven datasets and do not address label scarcity or self-supervised representation learning. Consequently, the prior studies discussed above, including both ALE-specific models and broader AE-GPR hybrids, exhibit three consistent gaps: (i) label dependence, as high accuracy typically requires substantial labeled fractions; (ii) limited uncertainty characterisation, with point estimates favoured over calibrated predictive distributions; and (iii) limited robustness evidence, with few works quantifying sensitivity to feature perturbations or isolating where gains arise via ablations. sGprAle addresses these gaps by learning a compact latent manifold exclusively from unlabeled samples via an autoencoder and training a Gaussian process solely in this latent space using a small labeled subset (label efficiency), selecting kernel hyperparameters via Bayesian optimisation to preserve calibration (principled uncertainty), and providing ablation studies together with targeted sensitivity tests across anchor ratio, transmission range, node density, and iterations (robustness evidence). In addition, by integrating optimisation-driven synthetic data generation, sGprAle improves coverage of operational regimes while keeping supervision costs low. Our experiments demonstrate consistent gains over raw-feature GPR in low-label regimes, deliver calibrated uncertainty for risk-aware deployment, and clarify when the self-supervised design confers the greatest benefit.

III. MATERIAL AND METHODS

A. Synthetic Data: ALE

The performance of any machine learning model is highly dependent on the quality of the data. This data can be either real (i.e., field measurements taken through sensors) or synthetic (i.e., derived from mathematical models, simulations, or rules). Obtaining real data for wireless networks is often challenging due to the associated costs. This limitation can be systematically overcome by using synthetic data. In recent years, the use of synthetic data has increased significantly, aligning well with Gartner's prediction that the exponential rise in synthetic data usage will surpass that of real data by 2030 [33], [34], [35].

In this study, we used a modified Cuckoo Search (CS) algorithm [36] to generate the synthetic node localisation data. To do so, we deployed sensor nodes randomly in an area of $A \times B$ square units. The system consists of χ anchor nodes, which serve as references for all n unknown nodes in the network that need to be localised. All sensors can transmit/receive data within a transmission range of r distance units. The anchor nodes have prior information about their locations either through GPS technology or predetermined positions. An unknown node is considered localisable only if it has at least three anchor nodes within its transmission range.

The RSSI helps unknown nodes calculate their distances from anchor nodes, modeled as log-normal shadowing [37]:

$$PL(d) = PL_0 + 10 \times \eta \times \log_{10} \left(\frac{d}{d_0} \right) + X_g \quad (1)$$

where $PL(d)$ is the path loss, PL_0 is the path loss at reference distance d_0 , d is the distance between the transmitter and receiver, η is the path loss exponent, and X_g is a Gaussian random variable for fading.

A ranging error is often observed due to the log-normal shadowing which follows a zero-mean Gaussian distribution and has a variance (ς^2) of:

$$\varsigma^2 = \varnothing^2 \times D_{ij}^2 \quad (2)$$

where $\varnothing$ is the noise factor (set to 0.1) and D_{ij} is the Euclidean distance between nodes:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

A disk model ensures network connectivity: two nodes i and j can communicate if $D_{ij} \leq r$, where r is the transmission range. The measured distance D'_{ij} is:

$$D'_{ij} = D_{ij} + \beta_{ij} \quad (4)$$

where β_{ij} is the ranging error. To accurately estimate unknown node positions, the objective function (*obf*) is defined as:

$$obf(x_i, y_i) = \frac{1}{\chi} \sum_{j=1}^{\chi} (D_{ij} - D'_{ij})^2 \quad (5)$$

where $\chi \geq 3$ since at least three anchor nodes are required for trilateration. The (x_i, y_i) corresponding to the minimum *obf* value is the estimated unknown node position.

The modified CS algorithm initialises a random population of candidate solutions for unknown nodes, calculating each solution's fitness using the *obf*. The worst candidates are replaced by new random solutions, iterating until the global best solutions' coordinates are selected as the unknown nodes' positions. For each global solution, we considered the anchor ratio, transmission range, node density, and iteration as the input features, with the corresponding ALE as the response variable. This simulation was run 107 times, which is of the size of the synthetic data reported in Singh et al. [13] (available at the UCI repository). The realism is reinforced in the use of synthetic data through a physically motivated model, which includes RSSI with log-normal shadowing, a disk-connectivity constraint with range r , and a trilateration feasibility check (requiring at least three anchors within range). We sweep the parameters η , r , anchor ratio, and node density across typical WSN ranges, rejecting degenerate layouts (e.g., disconnected graphs or ill-conditioned trilateration), and resample to maintain plausible geometry and coverage.

B. Model Development and Scenarios

For the given input feature $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features, we first normalise the features using z-score normalisation [38] technique:

$$\mathbf{X}_{\text{norm}} = \frac{\mathbf{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \quad (6)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and standard deviation vectors of $\mathbf{X}$, respectively. We then partition the entire dataset into labeled and unlabeled subsets by using the Mersenne Twister random generator [39]. The partitioning is done in six different scenarios

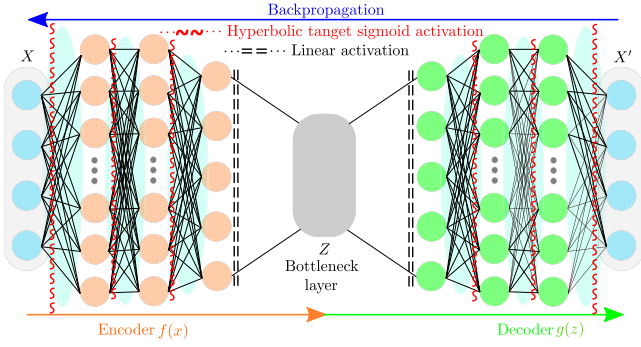


Figure 1. The autoencoder framework, denoted as ψ_T , was employed to develop the self-supervised version of Gaussian process regression. This framework is based on a fully connected feed-forward neural network architecture, which includes three hidden layers with 10, 10, and 5 neurons, respectively, leading up to the latent space (bottleneck) layer.

by leveraging the partition object cv , where the holdout ratio is varied uniformly from 10% to 60% in increments of 10%. This process results in the generation of datasets under six different scenarios.

$$\mathbf{X}_{\text{labeled}}, \mathbf{Y}_{\text{labeled}} = \mathbf{X}[\text{test}(cv)], \mathbf{Y}[\text{test}(cv)] \quad (7)$$

$$\mathbf{X}_{\text{unlabeled}}, \mathbf{Y}_{\text{unlabeled}} = \mathbf{X}[\text{training}(cv)], \mathbf{Y}[\text{training}(cv)] \quad (8)$$

All experiments were conducted using a fixed random seed to ensure reproducibility, and performance metrics were computed over the full unlabeled evaluation set for each labeled-data scenario.

Once we generated the different scenarios, we developed an autoencoder by leveraging a fully connected feed-forward neural network [40], [41], [42] (Figure 1). The autoencoder is implemented as a symmetric, fully connected neural network with an encoder-decoder structure of $4 \rightarrow 10 \rightarrow 10 \rightarrow 5 \rightarrow 10 \rightarrow 10 \rightarrow 4$, where the input dimension corresponds to the four network features (anchor ratio, transmission range, node density, and iterations), and the latent representation is constrained to a five-dimensional bottleneck. All hidden layers employ a hyperbolic tangent sigmoid activation function to capture nonlinear dependencies among the input features, while a linear activation function is used at the output layer to preserve the numerical scale of the reconstructed inputs. The autoencoder is trained exclusively on the unlabeled subset using the Levenberg-Marquardt optimization algorithm, with the reconstruction objective defined as the mean squared error between the input and reconstructed output. Training is terminated upon convergence of the reconstruction loss. To ensure consistency and reproducibility across all labeled-data scenarios, the autoencoder architecture, activation functions, latent dimensionality, and training settings are kept fixed throughout the study. This type of neural network is used to learn efficient codings of input data. It consists of an encoder $f(\mathbf{x})$ and a decoder $g(\mathbf{z})$, where $\mathbf{z}$ is the encoded representation of $\mathbf{x}$:

$$\mathbf{z} = f(\mathbf{x}) = \vartheta(\mathbf{W}_e \mathbf{x} + \mathbf{b}_e) \quad (9)$$

$$\hat{\mathbf{x}} = g(\mathbf{z}) = \vartheta(\mathbf{W}_d \mathbf{z} + \mathbf{b}_d) \quad (10)$$

where $\mathbf{W}_e$ and $\mathbf{b}_e$ are the weights and biases of the encoder, and $\mathbf{W}_d$ and $\mathbf{b}_d$ are the weights and biases of the decoder. ϑ represents an activation function. We trained this autoencoder only on the unlabeled data by minimising the reconstruction loss between the input and the output to learn a compact latent manifold:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \quad (11)$$

After training, the encoder is frozen and used to extract latent features from both labeled and unlabeled data:

$$\mathbf{Z}_{\text{unlabeled}} = f(\mathbf{X}_{\text{unlabeled}}) \quad (12)$$

$$\mathbf{Z}_{\text{labeled}} = f(\mathbf{X}_{\text{labeled}}) \quad (13)$$

These extracted features are then used as inputs to the GPR model. It is a non-parametric, Bayesian approach to regression that does not assume a specific parametric form for the underlying function [26], [43], [44]. Instead, it learns the relationship between these features and the target variables in a probabilistic manner, providing predictions along with associated uncertainties.

For example, given a set of m training points $(\mathbf{Z}, \mathbf{Y})$, where $\mathbf{Z}$ represents the input features and $\mathbf{Y}$ represents the corresponding outputs, the goal is to infer a function $f(\mathbf{z})$ that maps inputs $\mathbf{z}$ to outputs y . The GPR model is characterised by a mean function $\mu(\mathbf{z})$ and a covariance function $k(\mathbf{z}, \mathbf{z}')$, which together define the distribution over possible functions.

$$f(\mathbf{z}) \sim \mathcal{GP}(\mu(\mathbf{z}), k(\mathbf{z}, \mathbf{z}')) \quad (14)$$

For simplicity, we considered the mean function to be zero:

$$\mu(\mathbf{z}) = 0 \quad (15)$$

In this study, the covariance function (or kernel function) k is selected to be the exponential kernel [45], defined as:

$$k(\mathbf{z}, \mathbf{z}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{z} - \mathbf{z}'\|}{\ell}\right) \quad (16)$$

where σ_f is the signal variance and ℓ is the length scale.

Given the training data $\mathbf{Z}_{\text{labeled}}$ and $\mathbf{Y}_{\text{labeled}}$, the posterior distribution for a test point $\mathbf{z}_*$ is given by:

$$\mathbf{f}_* | \mathbf{Z}_{\text{labeled}}, \mathbf{Y}_{\text{labeled}}, \mathbf{z}_* \sim \mathcal{N}(\mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{Y}_{\text{labeled}}, \mathbf{k}_{**} - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*) \quad (17)$$

where $\mathbf{K}$ is the covariance matrix of the training points. $\mathbf{k}_*$ is the covariance vector between the test point and the training points. $\mathbf{k}_{**}$ is the covariance of the test point. σ_n^2 is the noise variance.

We employed Bayesian optimisation [46], [47], [48] to determine the optimal hyperparameters for the GPR model (Figure 2). This method involves constructing a probabilistic model of the objective function and using it to guide the selection of the most promising hyperparameters for evaluation. The objective function, which represents the error metric (e.g., mean squared error, MSE) to be minimised, is modeled by a surrogate model, in this study, a Gaussian process is used to approximate the

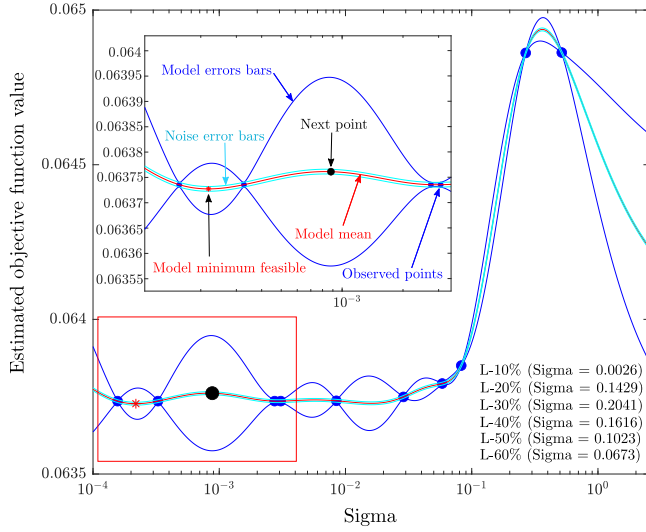


Figure 2. Illustration of the Bayesian optimisation process. The blue dots represent the observed points, while the red line indicates the model mean. The blue lines denote the model error bars, and the cyan lines show the noise error bars. The black dot marks the next point to be evaluated, and the red asterisk represents the model's minimum feasible value. The x-axis displays the sigma values, and the y-axis shows the estimated objective function value.

objective function, denoted as $\zeta(x)$:

$$\zeta(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (18)$$

where $\mathbf{x}$ represents the hyperparameters. $\mu(\mathbf{x})$ is the mean function. $k(\mathbf{x}, \mathbf{x}')$ is the covariance function.

To select the next set of hyperparameters to evaluate, we use the acquisition function $a(\mathbf{x})$, which balances exploration and exploitation. In our study, we employed the Expected Improvement (EI) as the acquisition function [49]:

$$a(\mathbf{x}) = \mathbb{E}[\max(0, \zeta(\mathbf{x}) - \zeta(\mathbf{x}^+))] \quad (19)$$

where $\mathbf{x}^+$ is the best hyperparameter set found so far. The optimised value of sigma for each scenario is shown in Figure 2. The complete optimisation process is carried out iteratively as follows:

- 1) Fit the surrogate model to the observed data.
- 2) Select the next set of hyperparameters $\mathbf{x}_{\text{next}}$ by maximising the acquisition function:

$$\mathbf{x}_{\text{next}} = \arg \max_{\mathbf{x}} a(\mathbf{x}) \quad (20)$$

- 3) Evaluate the objective function at $\mathbf{x}_{\text{next}}$.
- 4) Update the surrogate model with the new observation.

To reduce computational time, we utilise a parallel computing framework that evaluates multiple hyperparameter sets simultaneously. This approach accelerates the optimisation process significantly, with the speed-up factor depending on the number of workers employed [50].

Finally, using the trained GPR model, we predict the responses for the unlabeled data:

$$\hat{\mathbf{Y}}_{\text{unlabeled}} = \mathbb{E}[\mathbf{f}_* | \mathbf{Z}_{\text{labeled}}, \mathbf{Y}_{\text{labeled}}, \mathbf{Z}_{\text{unlabeled}}] \quad (21)$$

Algorithm 1. Pseudocode for sGprAle.

- 1: **Input:** Data $\mathbf{D}$ with features $\mathbf{X}$ and target $\mathbf{Y}$, Labeled Ratios $\mathbf{R} = [0.10, 0.20, 0.30, 0.40, 0.50, 0.60]$
 - 2: **Output:** Performance Metrics (R, RMSE)
 - 3: **Define:** GPR parameters
 - 4: MaxObjectiveEvaluations: 100,
 - 5: HyperparameterOptimisationOptions:
 - 6: Fit and active set method & Basis and Kernel function
 - 7: **for** each $\mathbf{r} \in \mathbf{R}$ **do**
 - 8: Split data $\mathbf{D}$ into features $\mathbf{X}$ and target $\mathbf{Y}$
 - 9: Normalise $\mathbf{X}$
 - 10: **Set** labeledRatio = $\mathbf{r}$ (unlabeledRatio = $1 - \mathbf{r}$)
 - 11: Create partition object cv for labeled and unlabeled data with holdout ratio = $\mathbf{r}$
 - 12: Split data into labeled and unlabeled sets:
 - 13: $\mathbf{X}_{\text{Unlabeled}} \leftarrow \mathbf{X}[\text{training}(cv), :]$
 - 14: $\mathbf{Y}_{\text{Unlabeled}} \leftarrow \mathbf{Y}[\text{training}(cv)]$
 - 15: $\mathbf{X}_{\text{Labeled}} \leftarrow \mathbf{X}[\text{test}(cv), :]$ & $\mathbf{Y}_{\text{Labeled}} \leftarrow \mathbf{Y}[\text{test}(cv)]$
 - 16: **Define** Autoencoder architecture
 - 17: ψ = Feed Forward Neural Network (4:10:10:5:10:10:4)
 - 18: Train Autoencoder on $\mathbf{X}_{\text{Unlabeled}}$:
 $\psi_T = \psi(\mathbf{X}_{\text{Unlabeled}})$
 - 19: **Extract** features using ψ_T :
 - 20: encodedFeaturesUnlabeled $\leftarrow \psi_T(\mathbf{X}_{\text{Unlabeled}})$
 - 21: encodedFeaturesLabeled $\leftarrow \psi_T(\mathbf{X}_{\text{Labeled}})$
 - 22: **for** each Iter $\in$ MaxObjectiveEvaluations **do**
 - 23: Mdl=Train GPR(encodedFeaturesLabeled, $\mathbf{Y}_{\text{Labeled}}$)
 - 24: **Optimisation:** Bayesian in parallel computing mode
 - 25: **end for**
 - 26: **Predict** on unlabeled set:
 - 27: $\mathbf{Y}_{\text{PredUnlabeled}} \leftarrow \text{predict}(\text{Mdl}, \text{encodedFeaturesUnlabeled})$
 - 28: **Evaluate** performance on $\mathbf{Y}_{\text{Unlabeled}}$:
 - 29: $\mathbf{R} \leftarrow \text{corr}(\mathbf{Y}_{\text{PredUnlabeled}}, \mathbf{Y}_{\text{Unlabeled}})$
 - 30: $\text{MSE} \leftarrow \text{mean}((\mathbf{Y}_{\text{PredUnlabeled}} - \mathbf{Y}_{\text{Unlabeled}})^2)$
 - 31: $\text{RMSE} \leftarrow \text{sqrt}(\text{MSE})$
 - 32: **Return** R and RMSE
 - 33: **end for**
-

The complete process is illustrated in pseudo-code in Algorithm 1, while the overall methodology adopted to estimate the average node localisation is depicted in Flowchart (Figure 3).

IV. RESULTS AND DISCUSSION

A. Model Performance

Figure 4 shows the linear fit between the predicted ALE and observed ALE from the sGprAle model. The different colors represent model results corresponding to six different scenarios, each indicating a varying percentage of labeled data used to train

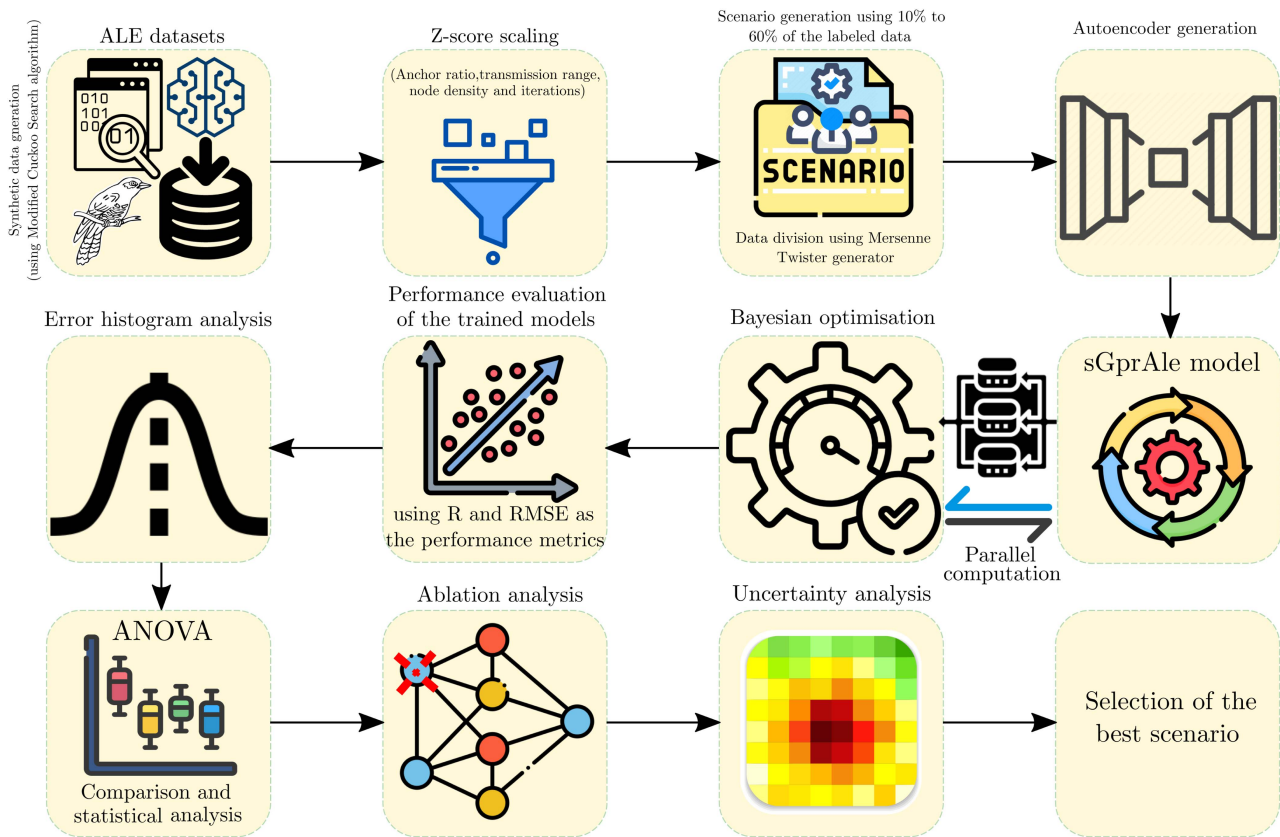


Figure 3. Stepwise methodology for the development and validation of the proposed sGprAle model.

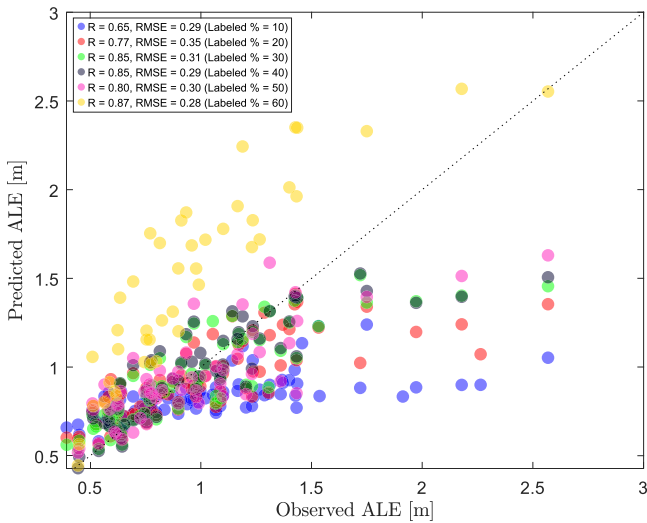


Figure 4. Performance of the self-supervised GPR. The dashed line represents the $y=x$ line.

the model. These percentages range from 10% (shown in blue) to 60% (shown in yellow). The dotted line represents the ideal $y = x$ line, where the predicted values would perfectly match the observed values. The general trend indicates a positive correlation between the predicted and the observed ALE, meaning as the observed error increases, the predicted error also tends to increase. For the 10% labeled data scenario, the predictions

are relatively close to the $y = x$ line, especially for lower ALE values. However, there are fewer points at higher ALE values. As an example, for the 10% labeled scenario, the Pearson correlation coefficient is $R = 0.65$, with a 95% confidence interval of $[0.51, 0.75]$ computed using Fisher's r -to- z transformation, confirming the statistical reliability of the reported association. The pattern for the 20% labeled data scenario is similar to the 10% scenario, but with a slight increase in the spread of points as ALE values increase. For the 20% labeled scenario, the Pearson correlation coefficient is $R = 0.77$, with a 95% confidence interval of $[0.67, 0.84]$. For the 30% and 40% labeled data scenarios, the points are more dispersed compared to the lower percentage scenarios, indicating more variability in predictions. For labeled data $> 50\%$, the points show an increased spread, particularly for higher ALE values, with the 60% scenario showing the widest spread among all, indicating less accuracy in predictions for higher ALE values.

The data points generally cluster around the $y = x$ line at lower ALE values but show increasing deviation as the ALE values increase. This indicates that the model performs better at lower ALE values but becomes less accurate as the observed ALE increases. Interestingly, as the percentage of labeled data increases (say from 10% to 20%), the scatter of the points increases, especially for higher ALE values. This suggests that simply increasing the amount of labeled data does not necessarily lead to better predictions for higher error ranges. Such non-monotonic performance behaviour at low labeled ratios is characteristic of self-supervised learning frameworks,

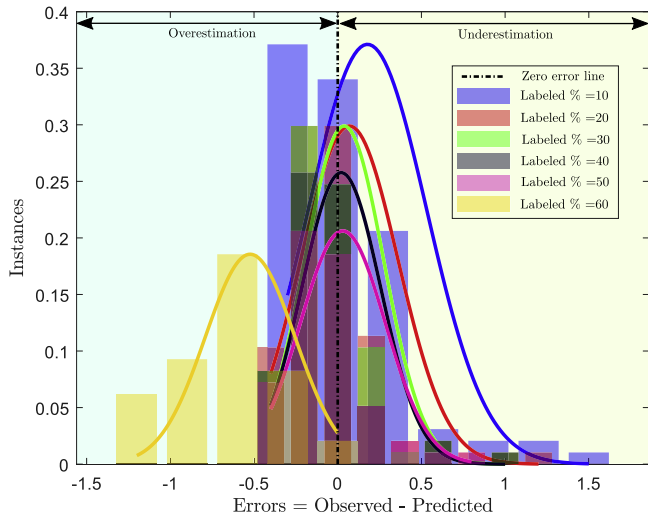


Figure 5. Error histogram analysis of the sGprAle for each scenario of the labeled data.

where changes in labeled-unlabeled balance simultaneously influence representation learning and regression alignment, particularly in data-scarce regimes [51].

B. Error Histogram Analysis

Figure 5 provides an error histogram analysis for the sGprAle model, showing the distribution of errors for different labeled data scenarios. The x-axis represents the error (observed - predicted), while the y-axis shows the instances. The dotted black line represents the zero error line, with the overestimation region (negative error) to the left and the underestimation region (positive error) to the right. Each histogram is fitted with a Gaussian curve to highlight the error distribution peak. The peak of the Gaussian fit for 20%, 30%, 40%, and 50% labeled data scenario is exactly centered around the zero error line, indicating that the model does not have any bias towards overestimation or underestimation. It indicates that most of the errors are concentrated near zero, with fewer instances of large errors in either direction. The presence of errors in both the overestimation and underestimation regions indicates a high prediction variability in these scenarios.

The distribution for the 10% labeled data scenario is centered to the right side of the zero error line, indicating that the model is giving slightly underestimated results. However, the peak of the Gaussian fit is the highest among all scenarios, suggesting fewer large errors and better overall performance even with a small amount of labeled data. The distribution of the 60% labeled data scenario exhibits a significant shift towards the negative error region, suggesting a strong tendency towards overestimation. The Gaussian fit exhibits the lowest peak among all scenarios, suggesting that the model's predictions are less reliable.

C. Comparison and Statistical Analysis

To statistically validate our results, we conducted a one-way ANOVA followed by a Tukey Honestly Significant Difference (HSD) post-hoc test. This method compares the means

TABLE I
POST HOC INTER-COMPARISON RESULTS

Group A	Group B	Lower Limit	A-B	Upper Limit	p-value
Labeled 10%	Labeled 20%	-0.011	0.108	0.228	0.103
Labeled 10%	Labeled 30%	0.015	0.139	0.263	0.018
Labeled 10%	Labeled 40%	0.027	0.157	0.286	0.007
Labeled 10%	Labeled 50%	0.011	0.147	0.284	0.026
Labeled 10%	Labeled 60%	0.553	0.701	0.849	0
Labeled 20%	Labeled 30%	-0.096	0.031	0.158	0.983
Labeled 20%	Labeled 40%	-0.084	0.049	0.181	0.903
Labeled 20%	Labeled 50%	-0.100	0.039	0.180	0.967
Labeled 20%	Labeled 60%	0.442	0.593	0.744	0
Labeled 30%	Labeled 40%	-0.119	0.018	0.154	0.999
Labeled 30%	Labeled 50%	-0.135	0.008	0.152	0.999
Labeled 30%	Labeled 60%	0.408	0.562	0.716	0
Labeled 40%	Labeled 50%	-0.158	-0.009	0.139	0.999
Labeled 40%	Labeled 60%	0.386	0.544	0.703	6.02e-23
Labeled 50%	Labeled 60%	0.389	0.553	0.718	9.97e-22

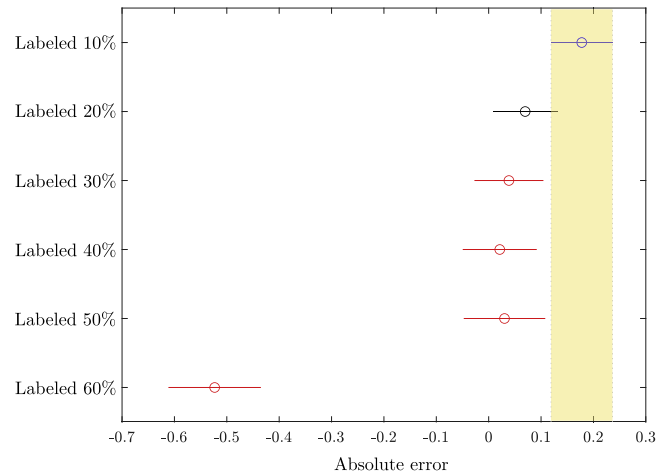


Figure 6. Tukey's HSD method for ANOVA multiple comparison test.

of all scenarios and calculates adjusted p-values for pairwise comparisons. Before the ANOVA, we performed a Kolmogorov-Smirnov normality test on the errors for each scenario. The results confirmed that the errors were normally distributed for all scenarios. Subsequently, the one-way ANOVA was conducted. The outcomes of the pairwise comparisons are summarised in Table I.

The comparison between 10% labeled data and 60% labeled data scenario shows a highly significant difference ($p = 0$), with a mean difference (A-B) of 0.701. Similar highly significant differences are observed between 20% and 60%, 30% and 60%, 40% and 60%, and 50% and 60%, all with $p = 0$. Moderate significant difference is observed between 10% and 30%, 40%, and 50% with p values of 0.018, 0.007, and 0.026, respectively. On the other hand, comparisons among 20%, 30%, 40%, and 50% show no significant differences with p values ranging from 0.90 to 0.99. This is also evident from the lower mean differences (A-B) among these scenarios, suggesting similar model performance. In contrast, the mean difference is highest between 10% and 60%, as well as other comparisons involving 60%, indicating that the model's performance significantly changes with 60% labeled data compared to lower percentages.

Furthermore, we presented pairwise comparisons graphically (Figure 6). In this figure, the performance level of the 10% labeled data scenario (shown in blue) is marked by a vertical

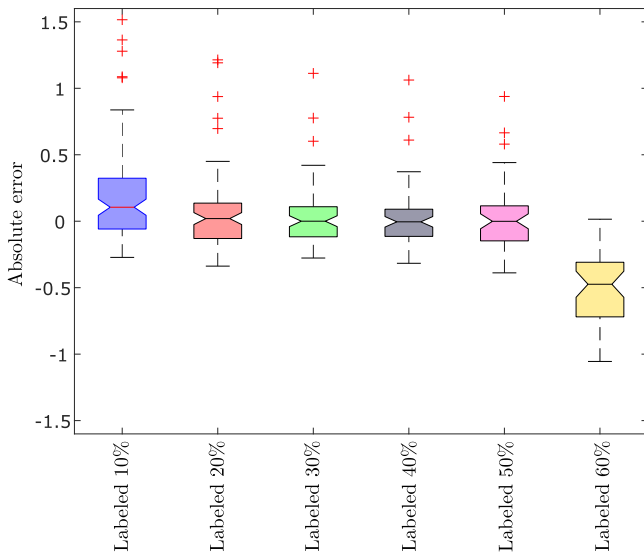


Figure 7. The error analysis for each scenario of labeled data percentage is depicted using boxplots, with the ANOVA results. In each boxplot, the top and bottom edges of the box indicate the upper and lower quartiles (75th and 25th percentiles), respectively. The line within the box denotes the sample median, while outliers are represented by red “+” symbols.

yellow patch. The performance level or interval of the 20% scenario (shown in black) slightly overlaps with the yellow patch, indicating no statistically significant difference between the two. Conversely, the performance intervals of the 30%, 40%, 50%, and 60% scenarios (marked in red) do not overlap with the yellow patch, suggesting they are statistically different from the 10% scenario. The clear separation of the performance intervals (except for the 20% scenario) suggests that increasing the labeled data percentage has a significant impact on model performance. This analysis underscores the importance of labeled data percentage in influencing model performance and guiding decisions on the optimal amount of labeled data required for training.

To visually represent the scenario errors identified in the ANOVA analysis, we plotted the box plots of the ANOVA-derived error values (Figure 7). The analysis reveals that the model maintains consistent error performance across lower labeled data scenarios (10% to 50%). The 60% labeled data scenario shows stable performance with no outliers, indicating that increasing the labeled data to 60% does not negatively impact the model’s performance. This result complements the ANOVA analysis, providing a clear depiction of error distribution and variability across different labeled data scenarios.

From a deployment perspective, the multiple-comparison analysis suggests a practical ‘sweet spot’ at moderate supervision: using ~20-40% labeled data delivers accuracy statistically indistinguishable from higher-labeled regimes while avoiding the susceptibility observed in the 50% case. Thus, for fixed labelling budgets, we recommend starting at 20% and expanding to 30-40% when tighter RMSE targets are needed, reserving additional annotation for targeted edge cases. Given the sensitivity of performance to iterations, practitioners should either

stabilise this driver (e.g., fixing optimisation schedules) or apply feature-robust strategies. When labelling costs dominate, an active-labelling loop that prioritises uncertain or high-ALE configurations can further compress the required labeled fraction without sacrificing calibration.

D. Ablation Analysis

To analyse the impact of individual components (e.g., inputs and components) on the sGprAle’s overall performance, we perform ablation analysis. It reveals critical insights into its performance with varying input feature combinations and autoencoder configurations (Table II). The input ablation shows that using all four features: Anchor Ratio (AR), Transmission Range (TR), Node Density (ND), and Iterations (I) yields the best performance, with an average correlation coefficient (R) of 0.87 and an RMSE of 0.28, highlighting the model’s capability to capture complex interactions effectively. Individual combinations of these features also performed well, but none matched the comprehensive feature set’s accuracy and correlation. Apart from all the feature scenarios, the combination of anchor ratio and transmission range shows a good balance between R and RMSE, indicating effective performance. The anchor ratio and iterations combination consistently achieve higher R values, suggesting strong predictive power, but this comes at the cost of higher RMSE compared to other feature combinations. The transmission range and iterations combination, however, results in low R values, indicating weaker performance. The combination of anchor ratio, transmission range, and node density achieves high R values, demonstrating strong performance, but does not have the best RMSE.

In the autoencoder ablation, the configuration with hidden layers of 10, 10, and 5 neurons (10:10:05) provided the optimal balance, achieving the highest R of 0.87 and the lowest RMSE of 0.28. This configuration likely avoids overfitting while sufficiently learning essential data representations. In contrast, more neurons (20:20:20) result in a slightly increased RMSE, possibly due to overfitting, whereas fewer neurons (05:05:05) result in underfitting, leading to lower R values. These results underscore the importance of a balanced feature set and appropriately configured neural network layers in optimising model performance.

E. Uncertainty Analysis

To evaluate the stability and reliability of the model, we performed an uncertainty analysis. We systematically introduced $\pm 5\%$ and $\pm 10\%$ uncertainty into each of the four features one by one while keeping the remaining features constant. We then recorded the average percentage change in model performance for all the features. This analysis was conducted for all six scenarios (Figure 8(a)–(f)). We observed that, for the 10% and 60% scenarios, the percentage change in the response variable is within -0.85% to 0.60% . The 50% scenario exhibits the highest sensitivity to uncertainty, with changes in ALE exceeding $\pm 1\%$ in some cases. This indicates that the model’s performance for the 50% labeled data scenario is more susceptible to variations

TABLE II

ABLATION STUDY OF THE SGPRALE MODEL. THE FIRST HALF OF THE TABLE REPRESENTS THE INPUT ABLATION, SHOWING THE EFFECT OF VARYING THE INPUT FEATURE COMBINATION ON THE MODEL PERFORMANCE. AR REPRESENTS THE ANCHOR RATIO, TR REPRESENTS THE TRANSMISSION RANGE, ND REPRESENTS THE NODE DENSITY, AND I REPRESENT THE ITERATIONS. THE SECOND HALF OF THE TABLE DISPLAYS THE LAYER ABLATION FOR THE AUTOENCODER, ANALYSING THE IMPACT OF DIFFERENT LAYERS ON THE MODEL'S ACCURACY AND EFFICIENCY.

Scenario	Metrics	Input variation										Autoencoder variation									
		AR+TR	AR+ND	AR+I	TR+ND	TR+I	ND+I	AR+TR+ND	TR+ND+I	AR+ND+I	All features	5:5:5	10:10:10	10:10:5	10:5:5	20:20:20	20:20:10	20:10:10	20:10:5	20:5:5	
Labeled 10%	R	0.29	0.67	0.29	0.57	0.27	0.59	0.60	0.63	0.66	0.65	0.65	0.59	0.65	0.64	0.69	0.65	0.65	0.68	0.53	
	RMSE	0.56	0.36	0.53	0.34	0.33	0.49	0.35	0.30	0.35	0.29	0.29	0.30	0.29	0.29	0.31	0.27	0.29	0.30	0.29	
Labeled 20%	R	0.28	0.58	0.36	0.60	0.14	0.79	0.62	0.71	0.77	0.77	0.74	0.72	0.77	0.76	0.74	0.75	0.74	0.74	0.44	
	RMSE	0.71	0.57	0.48	0.38	0.37	0.39	0.39	0.30	0.38	0.35	0.37	0.37	0.35	0.36	0.33	0.36	0.36	0.36	0.58	
Labeled 30%	R	0.39	0.72	0.43	0.66	0.44	0.83	0.67	0.85	0.82	0.85	0.74	0.84	0.85	0.84	0.85	0.84	0.84	0.84	0.78	
	RMSE	0.59	0.46	0.60	0.57	0.46	0.34	0.42	0.30	0.35	0.31	0.42	0.33	0.31	0.31	0.28	0.33	0.32	0.32	0.40	
Labeled 40%	R	0.38	0.72	0.20	0.69	0.34	0.78	0.72	0.84	0.80	0.85	0.77	0.83	0.85	0.85	0.83	0.85	0.85	0.84	0.77	
	RMSE	0.58	0.43	0.44	0.48	0.52	0.32	0.35	0.31	0.37	0.29	0.36	0.30	0.29	0.26	0.32	0.28	0.31	0.27	0.36	
Labeled 50%	R	0.58	0.70	0.43	0.65	0.43	0.81	0.68	0.84	0.79	0.80	0.82	0.79	0.80	0.82	0.76	0.80	0.77	0.78	0.78	
	RMSE	0.43	0.50	0.61	0.60	0.59	0.33	0.52	0.28	0.34	0.30	0.29	0.32	0.30	0.31	0.38	0.32	0.36	0.32	0.33	
Labeled 60%	R	0.64	0.69	0.43	0.67	0.46	0.82	0.70	0.79	0.77	0.87	0.78	0.86	0.87	0.85	0.85	0.85	0.75	0.82	0.71	
	RMSE	0.43	0.49	0.60	0.54	0.54	0.32	0.48	0.41	0.36	0.28	0.37	0.27	0.28	0.29	0.30	0.30	0.38	0.32	0.41	
Average	R	0.43	0.68	0.36	0.64	0.35	0.77	0.67	0.78	0.77	0.80	0.75	0.77	0.80	0.79	0.79	0.79	0.77	0.78	0.67	
	RMSE	0.55	0.47	0.54	0.49	0.47	0.37	0.42	0.32	0.36	0.30	0.35	0.32	0.30	0.30	0.32	0.31	0.34	0.32	0.40	

in the input features. The 20% labeled data scenario demonstrates the least sensitivity to uncertainty, with changes in ALE generally within the -0.15% to 0.10% range. This suggests that the model trained on 20% labeled data is more robust to variations in input features. The uncertainty for the 30% and 40% scenarios ranges between -0.30% to 0.30%. The “Iterations” feature appears to be the most sensitive across all scenarios, as changes in this feature lead to the largest fluctuations in ALE values. This suggests that the number of iterations significantly influences the model's outcome. The impact of uncertainty tends to be more pronounced for the $\pm 10\%$ variations compared to the $\pm 5\%$ variations, indicating that larger perturbations in the input features have a more significant effect on the model's performance.

F. Computational Complexity

We formalise the computational profile of *sGprAle*. Let n denote the number of samples, with n_ℓ labeled and n_u unlabeled ($n_\ell + n_u = n$); d the input dimension (here $d = 4$); the autoencoder (AE) hidden widths $[h, h, h/2]$ (here $h = 10$); T_{AE} the number of AE training epochs; m the active-set size for sparse/subset Gaussian process (GP) regression; E the number of Bayesian-optimisation evaluations (here $E = 100$); and k the number of cross-validation folds during hyperparameter optimisation. Preprocessing, z -score normalisation, and the labeled/unlabeled hold-out split scale as

$$\begin{aligned} \text{Time}_{\text{prep}} &= \mathcal{O}(nd) \\ \text{Mem}_{\text{prep}} &= \mathcal{O}(nd) \end{aligned} \quad (22)$$

F. Autoencoder (unlabeled-only training): For a fully connected AE with encoder/decoder widths bounded by h , a single forward-backward pass per sample has cost $\mathcal{O}(dh + h^2)$. Training on the unlabeled pool therefore scales as

$$\begin{aligned} \text{Time}_{AE-\text{train}} &= \mathcal{O}(T_{AE} n_u (dh + h^2)) \\ \text{Mem}_{AE-\text{params}} &= \mathcal{O}(dh + h^2) \end{aligned} \quad (23)$$

After training, *forward-only* encoding of both labeled and unlabeled inputs costs

$$\text{Time}_{AE-\text{encode}} = \mathcal{O}((n_\ell + n_u)(dh + h^2)). \quad (24)$$

Let d_{lat} denote the dimensionality of the AE bottleneck (here $d_{\text{lat}} = 5$).

F. Gaussian process regression (labeled latent codes): We fit a sparse/subset GP on the labeled latent codes using an active set of size m and an exponential kernel. For fixed hyperparameters, a single GP fit requires kernel construction and linear algebra with cost

$$\begin{aligned} \text{Time}_{GP-\text{fit}} &= \mathcal{O}(n_\ell m d_{\text{lat}} + n_\ell m^2 + m^3) \\ \text{Mem}_{GP} &= \mathcal{O}(n_\ell m + m^2) \end{aligned} \quad (25)$$

Bayesian hyperparameter optimisation repeats this fitting E times with k -fold cross-validation, yielding

$$\text{Time}_{GP+\text{HPO}} = \mathcal{O}(E k (n_\ell m d_{\text{lat}} + n_\ell m^2 + m^3)), \quad (26)$$

which typically dominates the end-to-end runtime for moderate-large E and k . Prediction on the unlabeled latent codes (kernel evals against the active set plus projection) scales as

$$\text{Time}_{GP-\text{pred}} = \mathcal{O}(n_u m d_{\text{lat}}). \quad (27)$$

F. End-to-end complexity: Collecting terms, the overall time is

$$\begin{aligned} \text{Time}_{\text{total}} &= \mathcal{O}(nd) && (\text{prep}) \\ &+ \mathcal{O}(T_{AE} n_u (dh + h^2)) && (AE\text{train}) \\ &+ \mathcal{O}((n_\ell + n_u)(dh + h^2)) && (AE\text{encode}) \\ &+ \mathcal{O}(E k (n_\ell m d_{\text{lat}} + n_\ell m^2 + m^3)) && (GP + HPO) \\ &+ \mathcal{O}(n_u m d_{\text{lat}}) && (GP\text{pred}) \end{aligned} \quad (28)$$

with memory

$$\text{Mem}_{\text{total}} = \mathcal{O}(nd) + \mathcal{O}(dh + h^2) + \mathcal{O}(n_\ell m + m^2). \quad (29)$$

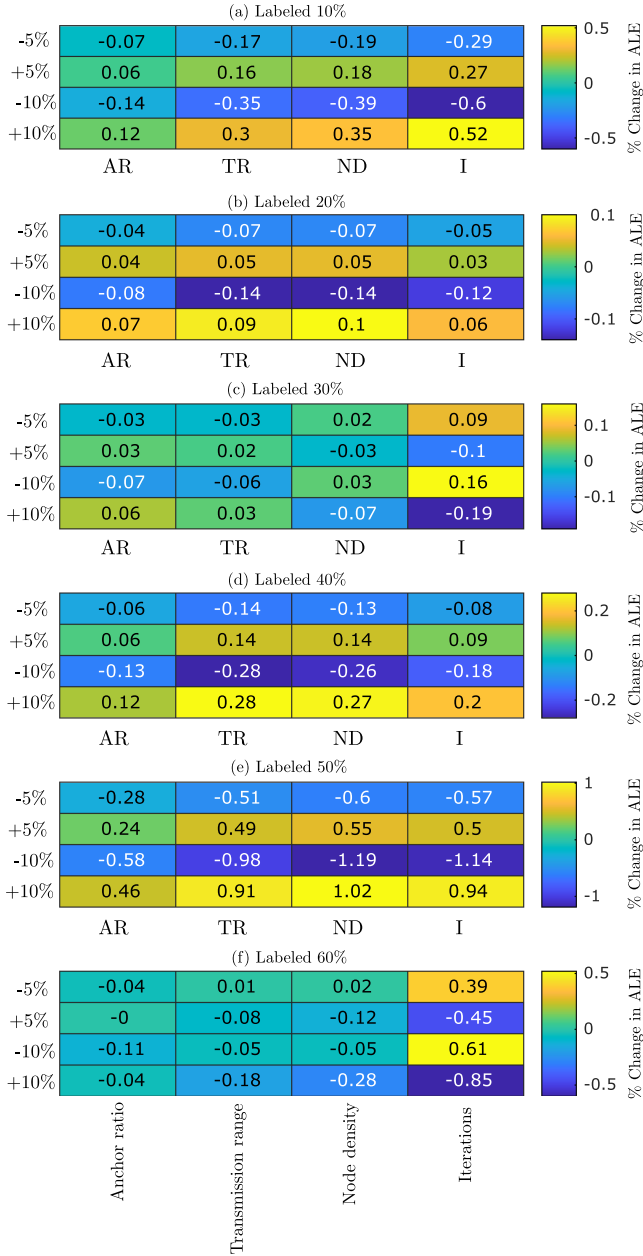


Figure 8. Performance of the different scenarios of the sGprAle model under small uncertainty in the input features ($\pm 5\%$ and $\pm 10\%$).

In our setting ($d = 4$, $h = 10$, $d_{\text{lat}} = 5$), the AE costs are linear in n with a small constant, whereas the GP hyperparameter loop grows as $\mathcal{O}(Ek n_{\ell} m^2)$ (plus $\mathcal{O}(Ek m^3)$ and the $\mathcal{O}(Ek n_{\ell} m d_{\text{lat}})$ kernel term), and thus dominates unless both n_{ℓ} and m are small. For reference, an exact GP (not employed here) would entail

$$\text{Time}_{\text{exact-GP}} = \mathcal{O}(Ek n_{\ell}^3), \quad \text{Mem}_{\text{exact-GP}} = \mathcal{O}(n_{\ell}^2), \quad (30)$$

highlighting the practical advantage of sparse/subset methods in the proposed pipeline.

G. Limitation and Future Work

Despite the promising results achieved by the sGprAle model in predicting ALE in wireless networks, several limitations warrant further investigation and improvements. Firstly, the model's

performance varies significantly with the percentage of labeled data used for training. While scenarios with 20% labeled data demonstrated robustness to uncertainty, other scenarios, particularly the 50% labeled data scenario, exhibited high sensitivity to input feature variations. This sensitivity indicates that the model's reliability can be compromised under certain conditions, suggesting a need for methods to enhance its robustness across all data labeling scenarios. Secondly, the uncertainty analysis revealed that the "Iterations" feature significantly influences the model's outcome. This dependency implies that the model's performance is particularly sensitive to this feature, highlighting a potential area for improving feature selection and engineering processes. Future research could explore advanced feature selection techniques or the integration of additional features to reduce this sensitivity. Another limitation is the scope of input variations and the autoencoder configurations used in this study. The ablation study indicated that the best performance was achieved when all input features were used and the autoencoder had specific neuron configurations. However, the exploration of other potential input combinations and autoencoder architectures was limited. Expanding this exploration could lead to discovering more optimal configurations that enhance the model's predictive accuracy and efficiency.

We observed that the proposed sGprAle model shows reasonable performance at lower ALE values. However, its accuracy diminishes for higher ALE values, particularly as the percentage of labeled data increases. The same is evident from the one-way ANOVA multiple comparison tests. This reveals significant differences in the model's performance when comparing scenarios with the highest percentage of labeled data (60%) to those with lower percentages (10%, 20%, 30%, 40%, and 50%). This indicates that increasing the percentage of labeled data to 60% leads to a significant change in model performance, suggesting potential issues or challenges in handling larger labeled datasets. This highlights a critical area for improvement in the model's capacity to handle larger labeled datasets and more complex scenarios accurately, possibly through better feature extraction, regularisation techniques, or more sophisticated model architectures.

Moreover, the current model was trained and validated on a specific dataset, which may limit its generalisability to other wireless network scenarios or environments. Future work should involve testing the sGprAle model on diverse datasets from different network settings to validate its applicability and robustness in varied real-world conditions.

Lastly, while the integration of GPR with autoencoder-extracted features showed improved performance, there is potential for exploring other machine learning or deep learning techniques. Hybrid models combining GPR with neural networks or other regression techniques might offer improved predictive capabilities and resilience to input variations.

In future work, addressing these limitations will involve several key steps:

- Developing techniques to enhance the model's robustness across different data labeling scenarios.
- Investigating advanced feature selection and engineering methods to reduce sensitivity to specific features.

- Conducting extensive experimentation with various input combinations and autoencoder configurations to identify optimal settings.
- Validating the model on diverse datasets to ensure its generalisability and reliability in different network environments.
- Exploring hybrid modeling approaches that integrate GPR with other advanced techniques for improved performance.

By addressing these areas, future research can further improve the sGprAle model, making it a more robust and versatile tool for predicting ALE in wireless networks.

V. CONCLUSION

In this study, we proposed sGprAle, a self-supervised Gaussian Process Regression framework for predicting Average Localisation Error in wireless sensor networks under limited labeled data conditions. By integrating autoencoder-based representation learning with probabilistic regression, the proposed approach effectively reduces reliance on labeled data while maintaining robust predictive performance.

Through comprehensive experimentation, including ablation studies, uncertainty analysis, and statistical validation, we demonstrated that learning compact latent representations from unlabeled data substantially enhances label efficiency and robustness. The results indicate that sGprAle achieves competitive performance relative to existing approaches while requiring significantly fewer labeled samples, making it well suited for practical deployment in resource-constrained wireless sensor networks.

The probabilistic nature of the model further enables uncertainty-aware predictions, which are essential for reliable localisation and decision-making in dynamic network environments. By accurately estimating ALE, the proposed framework supports improved localisation strategies, contributing to enhanced operational efficiency, energy savings, and network reliability.

Future work will focus on extending the framework to more diverse network configurations, improving robustness through advanced feature selection and regularisation techniques, and exploring hybrid architectures to further enhance generalisation across complex and large-scale wireless sensor network scenarios.

CREDIT AUTHOR STATEMENT

Abhilash Singh led the conceptualization, methodology, and software development, while also drafting, visualizing, and refining the manuscript. **J. Amutha and Jaiprakash Nagar** contributed to the initial draft of the manuscript (Section III A). **Sandeep Sharma** drove data management and provided critical insights and project oversight.

SOFTWARE AND DATA AVAILABILITY

- *Software name:* sGprAle (self-supervised Gaussian process regression for Average localisation error estimation)

- *Contact information:* abhilash.singh@ieee.org
- *Program language:* MATLAB
- *Software availability:* <https://abhilashsingh.net/codes.html>
- *Data availability:* The data used in this study can be downloaded from the UCI repository ([https://archive.ics.uci.edu/dataset/844/average+localization+error+\(ale\)+in+sensor+node+localization+process+in+wsns](https://archive.ics.uci.edu/dataset/844/average+localization+error+(ale)+in+sensor+node+localization+process+in+wsns))

CONFLICTS OF INTEREST

The authors have no conflicts of interest to declare.

ACKNOWLEDGMENT

Jaiprakash Nagar was associated with IIT Kharagpur while this work was carried out.

REFERENCES

- [1] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021.
- [2] X. Wei, H. Guo, X. Wang, X. Wang, and M. Qiu, "Reliable data collection techniques in underwater wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 404–431, Firstquarter 2022.
- [3] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Imran, and M. Ur-Rehman, "Challenges, applications, and future of wireless sensors in Internet of Things: A review," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5482–5494, Mar. 2022.
- [4] N. Xie, Y. Chen, Z. Li, and D. O. Wu, "Lightweight secure localization approach in wireless sensor networks," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6879–6893, Oct. 2021.
- [5] S. Zhang, O. T. Ajayi, and Y. Cheng, "A self-supervised learning approach for accelerating wireless network optimization," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 8074–8087, Jun. 2023.
- [6] Z. Yang et al., "Revolutionizing wireless networks with self-supervised learning: A pathway to intelligent communications," *IEEE Wireless Commun.*, vol. 32, no. 4, pp. 108–115, Aug. 2025.
- [7] J. Lei, J. Huang, L. Kong, G. Chen, and M. K. Khan, "DeFLoc: Deep learning assisted indoor vehicle localization atop FM fingerprint map," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19795–19806, Oct. 2022.
- [8] Q. Dai, B. Qian, G. O. Boateng, X. Guo, and N. Ansari, "GRIDLoc: A gradient blending and deep learning based localization approach combining RSS and CSI," *IEEE Wireless Commun. Lett.*, vol. 13, no. 9, pp. 2620–2624, Sep. 2024.
- [9] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The emerging trends of multi-label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7955–7974, Nov. 2022.
- [10] W. Deng and L. Zheng, "Are labels always necessary for classifier accuracy evaluation?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15069–15078.
- [11] O. Altay, M. Erel-Özçevik, E. V. Altay, and Y. Özçevik, "Average localization error prediction for 5G networks: An investigation of different machine learning algorithms," *Wireless Pers. Commun.*, vol. 136, pp. 289–319, 2024.
- [12] Y. Chen, Y. Pan, and D. Dong, "Residual tensor train: A quantum-inspired approach for learning multiple multilinear correlations," *IEEE Trans. Artif. Intell.*, vol. 4, no. 5, pp. 1101–1113, Oct. 2023.
- [13] A. Singh, V. Kotiyal, S. Sharma, J. Nagar, and C.-C. Lee, "A machine learning approach to predict the average localization error with applications to wireless sensor networks," *IEEE Access*, vol. 8, pp. 208253–208263, 2020.
- [14] K. Ileri, "LightGBM: Predicting average localization error through particle swarm optimization in wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw.*, 2024, pp. 1–4.
- [15] G. Mostafa, H. Mahmoud, T. Abd El-Hafeez, and M. E. ElAraby, "The power of deep learning in simplifying feature selection for hepatocellular carcinoma: A review," *BMC Med. Inform. Decis. Mak.*, vol. 24, no. 1, 2024, Art. no. 287.

- [16] G. Mostafa, H. Mahmoud, T. Abd El-Hafeez, and M. E. ElAraby, "Feature reduction for hepatocellular carcinoma prediction using machine learning algorithms," *J. Big Data*, vol. 11, no. 1, 2024, Art. no. 88.
- [17] A. Badawy, J. A. Fisteus, T. M. Mahmoud, and T. Abd El-Hafeez, "Topic extraction and interactive knowledge graphs for learning resources," *Sustainability*, vol. 14, no. 1, 2021, Art. no. 226.
- [18] B. Mukhopadhyay, S. Srirangarajan, and S. Kar, "RSS-based localization in the presence of malicious nodes in sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5503716.
- [19] B. Yang, L. Guo, R. Guo, M. Zhao, and T. Zhao, "A novel trilateration algorithm for RSSI-based indoor localization," *IEEE Sensors J.*, vol. 20, no. 14, pp. 8164–8172, Jul. 2020.
- [20] K. Sabale and S. Mini, "Localization in wireless sensor networks with mobile anchor node path planning mechanism," *Inf. Sci.*, vol. 579, pp. 648–666, 2021.
- [21] L. Gui, F. Xiao, Y. Zhou, F. Shu, and T. Val, "Connectivity based DV-hop localization for Internet of Things," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8949–8958, Aug. 2020.
- [22] A. Y. Lotfy, A. M. Zaki, T. Abd-El-Hafeez, and T. M. Mahmoud, "Privacy issues of public Wi-Fi networks," in *Proc. Int. Conf. Artif. Intell. Comput. Vis.*, 2021, pp. 656–665.
- [23] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, H. K. Qureshi, and M. Gidlund, "RSSI fingerprinting-based localization using machine learning in LoRa networks," *IEEE Internet Things Mag.*, vol. 3, no. 4, pp. 53–59, Dec. 2020.
- [24] J. Yan, Y. Meng, X. Yang, X. Luo, and X. Guan, "Privacy-preserving localization for underwater sensor networks via deep reinforcement learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1880–1895, 2021.
- [25] Z. Gong, C. Li, and F. Jiang, "A machine learning-based approach for auto-detection and localization of targets in underwater acoustic array networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15857–15866, Dec. 2020.
- [26] A. Singh, J. Nagar, S. Sharma, and V. Kotiyal, "A Gaussian process regression approach to predict the k-barrier coverage probability for intrusion detection in wireless sensor networks," *Expert Syst. Appl.*, vol. 172, 2021, Art. no. 114603.
- [27] A. Singh, J. Amutha, J. Nagar, and S. Sharma, "A deep learning approach to predict the number of k-barriers for intrusion detection over a circular region using wireless sensor networks," *Expert Syst. Appl.*, vol. 211, 2023, Art. no. 118588.
- [28] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4077–4085.
- [29] M. Nabati, H. Navidan, R. Shahbazian, S. A. Ghorashi, and D. Windridge, "Using synthetic data to enhance the accuracy of fingerprint-based localization: A deep learning approach," *IEEE Sens. Lett.*, vol. 4, no. 4, Apr. 2020, Art. no. 6000204.
- [30] X. Xu, R. Ansari, A. Khokhar, and A. V. Vasilakos, "Hierarchical data aggregation using compressive sensing (HDACS) in WSNS," *ACM Trans. Sensor Netw.*, vol. 11, no. 3, pp. 1–25, 2015.
- [31] W. Yu and C. Yu, "Application of weighted centroid algorithm based on weight correction in node localization of wireless sensor networks," *Sci. Rep.*, vol. 15, no. 1, 2025, Art. no. 23400.
- [32] S. Deshpande, H. Rappel, M. Hobbs, S. P. Bordas, and J. Lengiewicz, "Gaussian process regression deep neural network autoencoder for probabilistic surrogate modeling in nonlinear mechanics of solids," *Comput. Methods Appl. Mech. Eng.*, vol. 437, 2025, Art. no. 117790.
- [33] L. Ramos and J. Subramanyam, "Maverick research: Forget about your real data-synthetic data is the future of AI," *Gartner*, Jun. 24, 2021.
- [34] A. Singh, S. M. H. Mousavi, and J. Nagar, "F-TLBO-ID: Fuzzy fed teaching learning based optimisation algorithm to predict the number of k-barriers for intrusion detection," *Appl. Soft Comput.*, vol. 151, 2024, Art. no. 111163.
- [35] A. Salam, Q. Javaid, M. Ahmad, I. Wahid, and M. Y. Arafat, "Cluster-based data aggregation in flying sensor networks enabled Internet of Things," *Future Internet*, vol. 15, no. 8, 2023, Art. no. 279.
- [36] J. Cheng, L. Wang, and Y. Xiong, "Modified cuckoo search algorithm and the prediction of flashover voltage of insulators," *Neural Comput. Appl.*, vol. 30, pp. 355–370, 2018.
- [37] S. Kurt and B. Tavli, "Path-loss modeling for wireless sensor networks: A review of models and comparative evaluations," *IEEE Antennas Propag. Mag.*, vol. 59, no. 1, pp. 18–37, Feb. 2017.
- [38] H. Chen et al., "Personal-zscore: Eliminating individual difference for EEG-based cross-subject emotion recognition," *IEEE Trans. Affect. Comput.*, vol. 14, no. 3, pp. 2077–2088, Jul.–Sep. 2023.
- [39] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.
- [40] N. Goldenstein, J. Sulam, and Y. Romano, "Pivotal auto-encoder via self-normalizing relu," *IEEE Trans. Signal Process.*, vol. 72, pp. 3201–3212, 2024.
- [41] A. K. Tehrani, G. Cloutier, A. Tang, I. M. Rosado-Mendez, and H. Rivaz, "Homodyned K-distribution parameter estimation in quantitative ultrasound: Autoencoder and Bayesian neural network approaches," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 71, no. 3, pp. 354–365, Mar. 2024.
- [42] H. Li, L. Bai, W. Gao, J. Xie, and L. Huang, "Many-objective coevolutionary learning algorithm with extreme learning machine auto-encoder for ensemble classifier of feedforward neural networks," *Expert Syst. Appl.*, vol. 246, 2024, Art. no. 123186.
- [43] Z. Dang, B. Gu, C. Deng, and H. Huang, "Asynchronous parallel large-scale Gaussian process regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 8683–8694, Jun. 2024.
- [44] Z. Yuan and M. Zhu, "Lightweight distributed Gaussian process regression for online machine learning," *IEEE Trans. Autom. Control*, vol. 69, no. 6, pp. 3928–3943, Jun. 2024.
- [45] T. Teng, J. Chen, Y. Zhang, and B. K. H. Low, "Scalable variational Bayesian kernel selection for sparse Gaussian process regression," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 5997–6004.
- [46] J. Ungredda, M. Pearce, and J. Branke, "Bayesian optimisation vs. input uncertainty reduction," *ACM Trans. Model. Comput. Simul.*, vol. 32, no. 3, pp. 1–26, 2022.
- [47] J. Ungredda and J. Branke, "Bayesian optimisation for constrained problems," *ACM Trans. Model. Comput. Simul.*, vol. 34, no. 2, pp. 1–26, 2024.
- [48] P. Kent, A. Gaier, J.-B. Mouret, and J. Branke, "Bayesian optimisation for quality diversity search with coupled descriptor functions," *IEEE Trans. Evol. Comput.*, vol. 29, no. 2, pp. 302–316, Apr. 2025.
- [49] S. Ament, S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Unexpected improvements to expected improvement for Bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36, pp. 20577–20612.
- [50] A. Srivastava, S. P. Chockalingam, and S. Aluru, "A parallel framework for constraint-based Bayesian network learning via Markov blanket discovery," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 6, pp. 1699–1715, Jun. 2023.
- [51] L. Sánchez, I. Couso, and N. Costa, "Self-supervised spectral learning with an application to condition monitoring of tunnel ventilation systems under sparse data conditions," *Comput. Ind. Eng.*, vol. 211, 2025, Art. no. 111634.



Abhilash Singh (Member, IEEE) received the PhD degree from the Indian Institute of Science Education and Research Bhopal, India. He is currently a research fellow with the School of Mathematics and Computation, University of Leeds, U.K. His research interests include coupling advanced machine-learning algorithms with satellite imagery to enhance satellite observations and developing innovative machine-learning solutions for critical issues in wireless networks. He is an active member of the European Geosciences Union (EGU), American Geophysical Union (AGU), International Society for Photogrammetry and Remote Sensing, and Indian Radio Science Society. Dr. Singh was honored with the Vice-Chancellor Gold Medal for securing the first rank in his bachelor's and master's programs. He was awarded the prestigious "DST-INSPIRE" Fellowship by the Department of Science and Technology, Ministry of Science and Technology, Government of India, to pursue his doctoral studies. He has received the DAAD Fellowship and several grants from AGU and EGU.



Jaiprakash Nagar (Member, IEEE) received the integrated BTech (electronics and communication engineering) and MTech (wireless communication and networks) degrees from Gautam Buddha University, Greater Noida, India, in 2015, and the PhD degree from the Subir Chowdhury School of Quality and Reliability, Indian Institute of Technology Kharagpur, India. He was a post-doctoral researcher with Eurecom, France. He is currently a research engineer with Eurecom. His research interests include machine learning techniques for wireless communication, goal-oriented communication, network reliability, analytical modeling of wireless multihop networks, stochastic geometry, and age of information analysis.



J. Amutha received the MTech and PhD degrees from Gautam Buddha University, Greater Noida, India. She is currently an assistant professor with the Department of Artificial Intelligence and Data Science, E.G.S. Pillay Engineering College, India. Her research interests include wireless sensor networks, immersive technologies, optimization problems in WSNs, and machine learning techniques in WSNs. She was awarded the Vice-Chancellor's Gold Medal for securing first rank in her MTech degree from Gautam Buddha University. She has been recognized with Fr. Chacho Valiavettil, S.J. Academic Awards of Excellence (Best Teacher Award).



Sandeep Sharma received the PhD degree from Gautam Buddha University, Greater Noida, India. His research interests include wireless sensor networks, wireless network security, physical layer authentication, intrusion detection in wireless networks, cross-layer design, optimization problems in WSNs, and machine learning applications in WSNs. Dr. Sharma was the recipient of the Best Conference Paper at the international conference ICCCS in 2016, and the Young Scientist Award in 2019 for his research work. He is an active reviewer of *IET Communications*, *IEEE Wireless Communications Letters*, *IEEE Access*, *Journal of Information Technology* (Springer), *Wireless Personal Communications* (Springer), *Personal and Ubiquitous Computing* (Springer), *Multimedia Tools and Applications* (Springer), *Complex and Intelligent Systems* (Springer), *International Journal of Computer Applications in Technology* (Inderscience), *International Journal of Communication Systems* (Wiley), *Journal of The Institution of Engineers (India): Series B*, *Journal of Intelligent and Fuzzy Systems*, *Neural Networks* (Elsevier), *Engineering Applications of Artificial Intelligence* (Elsevier), *Expert System with Application* (Elsevier), *EURASIP Journal on Wireless Communications and Networking* (Elsevier), and *Sensor* (MDPI).

work. He is an active reviewer of *IET Communications*, *IEEE Wireless Communications Letters*, *IEEE Access*, *Journal of Information Technology* (Springer), *Wireless Personal Communications* (Springer), *Personal and Ubiquitous Computing* (Springer), *Multimedia Tools and Applications* (Springer), *Complex and Intelligent Systems* (Springer), *International Journal of Computer Applications in Technology* (Inderscience), *International Journal of Communication Systems* (Wiley), *Journal of The Institution of Engineers (India): Series B*, *Journal of Intelligent and Fuzzy Systems*, *Neural Networks* (Elsevier), *Engineering Applications of Artificial Intelligence* (Elsevier), *Expert System with Application* (Elsevier), *EURASIP Journal on Wireless Communications and Networking* (Elsevier), and *Sensor* (MDPI).