International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

MACHINE LEARNING FOR INVERSE CUBIC SENSOR CALIBRATION: A REGULARIZED BAYESIAN APPROACH

Ganti Srikanth [a,b*] , Gopinathan Sudheer [c], S Uma Devi [d]

[a]Department of Mathematics, Jawaharlal Nehru Technological University, Kakinada, India.

[b]Department of Mathematics, Gayatri Vidya Parishad College of Engineering (A), Visakhapatnam, India.

[c]Department of Mathematics, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India.

[d]Department of Engineering Mathematics,Andhra University, Visakhapatnam, India.

[*]Correspondence: srikanthganti81@gmail.com

Abstract.
Sensors often exhibit nonlinear responses that can be approximated by cubic polynomial transformations-balancing complexity and accuracy. The inverse problem of recovering the physical input from the sensor output is critical in medical diagnoses, control systems and environmental monitoring. The recovery is challenging due to ambiguity in reconstructing the input, error propagation and parameter estimation in the presence of noise or limited data points. The paper proposes a regularized Bayesian framework to ensure stable, unique solutions using machine learning (ML) algorithms to select the physically meaningful root with high accuracy and low computational cost. The ML algorithms achieve over 95% accuracy in root selection with XG Boost offering a 15% inference time than exhaustive root evaluation.

Keywords: Machine Learning (ML); Inverse Problem Cubic Sensor Calibration; Regularized Bayesian Approach; Random Forest & XG Boost
AMS: 62F15;62P30; 68T05 ;65F20;65C20;94A12

1. Introduction

In the era of digital transformation powered by artificial intelligence, sensors bridge the physical and digital domains, converting measurable quantities into output signals via mathematical transfer functions. These functions are generally nonlinear and cubic polynomial transformations are generally used to model the nonlinearity as they provide a striking balance between computational simplicity and accuracy [1]. This is particularly critical in real time applications where processing power is limited. Further, these transformations linearize outputs for compatibility with systems like analogue-to- digital converters and effectively capture moderate deviations from linearity without over fitting or amplifying noise [2]. The inverse problem of determining the input physical input quantity from the sensor output is ubiquitous in science, engineering and technology. The inverse problems can generally be categorized into two categories: parameter estimation and input reconstruction. Parameter estimation involves determining the parameters of a model based on observed input and output while input reconstruction is about figuring out the input of a system when you have the output and a model [3].In situations, where the measurements we require to probe a system are infeasible, dangerous or difficult to obtain the reconstruction problems are considered. These applied inverse problems are challenging because, they are often ill-posed and/or ill-conditioned [4]. An ill conditioned problem can be well posed but may be difficult to solve due to its unstable nature. A

Received: August 02, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

problem is said to be well posed in the sense of Hadamard [5] if there exist a unique solution to the problem that continuously depends on the data. By ill-posedness, we mean that a solution to be problem (i) may not exist (ii) may not be unique or (iii) is sensitive to measurement noise thereby being unstable [6].

Barinov et al [2] highlight that the inverse problem of approximation for a cubic transformation related to the sensor is more often incorrect due to violation of solution uniqueness, particularly when multiple real roots or Complex roots arise. However, with the Machine Learning (ML) advancements in recent years, the inverse problem of cubic transformation can be dealt with in a simple manner. Ensemble learning, a cornerstone within the ML landscape is an approach that improves the precision of the models in ML. Among the ensemble techniques, bagging models such as Random Forest and boosting model such as eXtreme gradient boosting (XG Boost) stand out as principal techniques, each designed to bolster the model accuracy through distinct mechanisms [7]. Random Forests are an effective tool in prediction as they do not overfit and injecting the correct kind of regressors makes them accurate classifiers and regressors [8]. The development of boosting was advanced by Friedman [9] and building on this, the introduction of XG Boost by Chen and Guestrin [10] marked a major leap forward in the efficiency and scalability of gradient boosting algorithms [11].

The Bayesian Framework is a versatile methodology for tackling inverse problems, the solution of the problem being the posterior distribution of the unknown. This framework allows us to incorporate prior information about the input into the solution and also yields a probabilistic solution to the inverse problem that quantifies uncertainty about the input [3].This paper proposes a hybrid approach combining a regularized Bayesian framework for robust inverse solutions and two ML models- Random Forest and Gradient boosting (XG Boost) -to select the correct root efficiently. The combinations of the work include:

1. A regularized Bayesian framework that ensures unique solutions and quantifies uncertainty.
2. Comparison of Random and XG boost for efficient root selection

We validate these methods on synthetic mimicking real-time sensor applications.

2. Theory

In general, the transformation of a measurable physical quantity as the output of a sensor is represented by the equation $A(f) = g, f \in F, g \in G$ (1)

where f can be one - dimensional, two - dimensional or three - dimensional or more and g can also be one - dimensional, two - dimensional or three - dimensional or more. Here A is a mathematical operator and F, G are metric spaces. According to Hadamard [5], the problem is said to be correctly posed or well - posed if the following conditions hold:

a) For each $g \in G$, the equation (1) has a unique solution.
b) The operator A^{-1} is defined on all of G and is continuous.

Otherwise, the problem is said to be incorrectly posed or ill posed. In general, we have limited discrete values of g and inferring f from the data is an ill-posed problem.

In the case of a cubic transformation for sensor, we have $A(x) = y$ (2)

Where $A(x) = b_0 + b_1 x + b_2 x^2 + b_3 x^3$ and $x_{\min} = 0 \le x \le x_{\max} = 1$, b_i are coefficients.

The inverse problem lies in recovering the input x given the sensor output y i.e., finding x such that

$b_0 + b_1 x + b_2 x^2 + b_3 x^3 - y = 0$ (3)

The difficulty lies in

i) Mechanism to select the physically meaningful root from the three roots

Received: August 02, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ii) Uncertainty and measurement noise is the coefficients b_0, b_1, b_2, b_3

Machine learning methods can enhance the solution to the inverse problem by addressing challenges like non-linearity, noise, parameter estimation and root ambiguity. Further ML enables models that generalize beyond the cubic form. To ensure unique solution and to quantity uncertainty a regularized Bayesian framework is used together with the techniques of Random Forest and Gradient boosting.
We assume that the observed y given x is distributed according to

$$y/x, f_1(\cdot) \sim p(y/x, f_1(\cdot)) \quad (4)$$

where the likelihood, $p(y/x, f_1(\cdot))$ is known up to a normalizing constant.

The Bayes rule states that $p(x/y, f_1(\cdot)) \propto p(y/x, f_1(\cdot))p(x)$ (5)

and it is the formal solution to the inverse problem using Bayesian formalism called the posterior distribution [12]. As the posterior is not available analytically, the only way forward is through sampling techniques. Here we use Markov chain Monte Carlo to sample $p(x/y)$ providing a point estimate and a 95% credible interval.

3. Proposed Methodology

We propose a two-pronged approach consisting of regularized Bayesian framework [13] for solving the inverse problem and a machine learning based root selector.

3.1. Regularized Bayesian Framework

The framework formulates the inverse problem as an optimization task with a regularization term to ensure stability.
The objective function is $min_x \|A(x) - y\|^2 + \lambda\|x\|^2$ (6)
Where λ is the Tikhonov regularization parameter [14].
Here $\|A(x) - y\|^2$ measures how well the model fits the observed output and $\lambda\|x\|^2$ promotes solutions that are physically meaningful and numerically stable.
To handle the uncertainty and noise, the coefficients b_i are modeled as Gaussian random variable i.e., $b_i \sim N(\mu_i, {\sigma_i}^2)$

where μ_i is the mean and ${\sigma_i}^2$ the variance of the coefficients b_i.
This approach allows the model to account for variations in the sensor behavior. The inverse problem now lies in estimating the posterior distribution $p(x/y)$

$$p(x/y) \propto exp\left(\frac{-1}{2\sigma^2}\|A(x) - y\|^2 - \lambda\|x\|^2\right) p(x) \quad (7)$$

where σ^2 represents the variance of measurement noise. $p(x)$ is a uniform prior over a bounded interval $[0, x_{max}]$ reflecting the normalized range of physical inputs.
Here $exp(-\lambda\|x\|^2)$ incorporates the Tikohonov regularization as a penality in the framework.
To estimate $p(x/y)$, the Markov chain Morte Carlo (MCMC) sampling is used.MCMCis computational technique that generates samples from probability distributions by constructing a Markov chain that converges to the target distribution. In this particular case, MCMC samples possible values of x that satisfy cubic equation while accounting for noise and coefficient uncertainty. The MCMC sampling is carried out through the Metropolis Hastings algorithm [15].

1. Initialize $x_0 \in [0,1], set\sigma_{prop} = 0.1$
2. For $t = 1$ to N do
3. Propose $x' \sim N(x_{t-1}, {\sigma^2}_{prop})$
4. Compute acceptance ratio: $\alpha = min\left(1, \frac{p(x'/y)}{p(x_{t-1}/y)}\right)$
5. Draw $u \sim uniform(0,1)$
6. If $u < \alpha$then

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Set$x_t = x'$
else
set$x_t = x_{t-1}$
end if
end for
Return samples $\{x_t\}$

The samples are used to compute the posterior mean and credible intervals. We train Random Forest and XG boost classifiers to predict the index of the correct root.

3.2. Random Forest Classifier

A random forest classifier is a collection of K tree structure classifiers $\{(X, \theta_{k_1}), k_1 = 1, \ldots K\}$ where θ_{k_1} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input X. The random vector θ_{k_1} governs the randomness in tree construction such as boot strap selection sampling and feature selection.

3.2.1. Tree Construction

In Bootstrap sampling, each tree is grown on a boot strap sample T_{k_1}, drawn with replacement from the learning sample L and at each node of the tree, a subset of F features are randomly selected from the M input features and the best split is chosen among these using the CART methodolgy. The split minimizes an impurity measure, Gini impurity is given by

$$i(t) = 1 - \sum_{j=1}^{J} p(j/t)^2 \quad (8)$$

Where $p(j/t)$ is the proportion of class j examples in node t. The best split s at node t maximizes the impurity decrease

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \quad (9)$$

Where p_L and p_R are the proportion of examples sent to the left and right child nodes t_L and t_R respectively.
The Trees are grown to maximum size without pruning ensuring each terminal node is either pure or meets a strapping criterion such as a minimum node size

3.2.2. Prediction

For an input X each tree $h(X, \theta_{k_1})$ predicts a class (root index)
The Random forest's prediction is the majority role

$$\hat{y}(X) = arg\ \max_j \sum_{k_1=1}^{K} I(\, h(X, \theta_{k_1}) = j) \quad (10)$$

where I is the indicator function.

3.3. XG boost classifier

XG Boost (eXtreme Gradient Boosting) is an ensemble learning method based on gradient-boosted decision trees.
XG boost constructs an ensemble of T decision trees, where the prediction is the sum of the outputs from identical trees. For classification, the model outputs a score for each class, and the predicted class is determined through a softmax function
The general objective function for XG boost is

$$L^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + f_t(X_i)) + \Omega(f_t) \quad (11)$$

Received: August 02, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Here $l(y_i, \hat{y}_i^{(t-1)} + f_t(X_i))$ is the loss function measuring the error between the true label $y_i$and the predicted score at iteration t.
$f_t(x_i)$ is the output of the t^{th} tree.

$\Omega(f_t)$is the regularization term to prevent overfitting and
$\hat{y}_i^{(t-1)} = \sum_{k=1}^{t-1} f_k(x_i)$is the cumulative prediction up to the $(t-1)^{th}$ tree.
For a multi-class classification task such as three root indices, let the model output a vector of scores $S_i = [s_{i1}, s_{i2}, s_{i3}]$ for the three classes at example i
The predicted probabilities are

$$p_{ij} = \frac{\exp(s_{ij})}{\sum_{k=1}^{3} \exp(s_{ik})} \tag{12}$$

where $s_{ij} = \sum_{t=1}^{T} f_t(X_i)_j$ is the score for class j at example i and $f_t(X_i)_j$ is the t-th tree's contribution to class j. The cross- entropy loss is

$$l(y_i, S_i) = -\sum_{j=1}^{3} I(y_i = j)\, log(p_{ij}) \tag{13}$$

where $I(y_i = j)$ is 1 if $y_i = j$ otherwise 0
The regularization term for the t[th] tree is $\Omega(f_t) = \gamma T_t + \frac{1}{2}\lambda \|w_t\|^2$ (14)
where T_t is the number of leaves in t^{th} tree
w_t is the vector of leave scores for the t^{th} tree and
γ and λ are penalty coefficients controlling tree complexity and leaf weight magnitude respectively.
XG boost minimizes the objective function iteratively using gradient boosting. At iteration t, the model adds a new tree f_t to minimize

$$L^{(t)} \approx \sum_{i=1}^{N} \left[l(y_i, \hat{y}_i^{(t-1)} + g_i f_t(X_i) + \frac{1}{2} h_i f_t(X_i)^2) \right] + \Omega(f_t) \tag{15}$$

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \& h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \tag{16}$$

are the first and second order gradients of the loss with respect to the previous prediction
Each tree f_t is constructed by greedily splitting nodes to maximize the reduction in the objective function
For a node split, XG Boost evaluates the gain

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{17}$$

whereI_L and $I_R$are the sets of examples in the left and right child nodes and
$I = I_L \cup I_R$. The split with the highest gain is chosen and γ acts as penalty to prevent excessive splits.
For a new input X, the XG boost classifier computes the score for each class

$$S_j(X) = \sum_{t=1}^{T} f_t(X)_j \tag{18}$$

The predicted class is

$$\hat{y}(X) = arg \max_{j \in \{1,2,3\}} p_j = arg \max_{j \in \{1,2,3\}} \frac{exp(S_j(X))}{\sum_{k=1}^{3} exp(S_k(X))} \tag{19}$$

This selects the root index with the highest probability for the present work.

4. Numerical experiment

We train Random Forest (RF) and Extreme gradient (XG) Boost classifiers to predict the index (0,1 or 2) of the correct root within [0,1]. The features of the problem include
(i) Parameters and discriminant of a cubic solver such as the Cardano's Method , Lagrange's method or Lebedev's Method.

Received: August 02, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

(ii) Monotonicity to check if the derivative
$A'(x) = b_1 + 2b_2x + 3b_3x^2$ has roots in [0 1] indicating non monotonicity.

4.1. Data Generation

We generate 10000 synthetic sampling by
(i) Sampling Coefficient b_0, b_1, b_2, b_3 to reflect the sensor behavior. The values of the coefficients determine the roots, curvature and behavior of the cubic equation, which in turn affect the difficulty in solving the inverse problem. The range of these coefficients also influence the accuracy of the machine learning models.
We generate three data sets that lead to different levels of difficulty for the models. The data sets are generated by sampling the coefficients in different ranges as follows:
Data set 1:$b_0 \in [-2.0,2.0]; b_1 \in [0.0,0.5]; b_2 \in [-2.0,0.0]; b_3 \in [0.1,1.0];$
Data set 2 :$b_0 \in [-1.0,5.0]; b_1 \in [-2.0,1.0]; b_2 \in [-4.0,2.0]; b_3 \in [0.01,0.5];$
Data set 1:$b_0 \in [-2.0,2.0]; b_1 \in [0.0,5.0]; b_2 \in [-1.0,4.0]; b_3 \in [0.5,2.0];$

We make some observations on the ranges of the coefficients with respect to its significance
Data set 1, Data set 3 have the same range for b_0 while Data set 2 has a wider asymmetric range. The larger range for b_0 in Data set 2 increase the variability in y- intercepts making the inverse problem harder.
Data set 1 has a small non negative range for b_1 limiting the linear terms influence. Data set 2 wider range for b_1 allows for both positive and negative slopes increasing complexity while data set 3's range can lead to steep linear contribution but the dominant b_3 term is likely to overshadow it.
The quadratic term coefficient $b_2$has widest range in data set 2, leading to complex polynomial shapes with marked local maxima/minima. Though the data set-3 also has a wider range, the domination of b_3 term reduces its relative impact. The range of a_2 in data set 1 which is a negative range adds complexity.
The cubic term coefficient b_3 's range in Data set 3 he is wide and has largest values leading to fewer real roots and easier to predict. The range of b_3 in data set 1 has a moderate range leading to complex behavior but data set 2's small range reduces the cubic terms dominance allowing other coefficient to play a larger role and increased complexity.

4.2. Bayesian Inference

The MCMC sampling with Tikhonov regularization is carried out to sample the posterior and estimate the mean x and credible intervals. The coefficients $b_i \sim N(\mu_i, \sigma^2)$and
$x \sim$Uniform (0 1). The likelihood function is defined with $y = b_0 + b_1x + b_2x^2 + b_3x^3$ and a regularization term $(-0.01x^2)$ is added to sample the posterior $p(x/y)$. We use $\sigma = 0.05$ and obtain Bayesian data. It is to be noted that the Bayesian data uses the mean from the Bayesian inverse to compute labels. The Bayesian data is used to test how well the Random Forest and XG Boost generalized to data where the target value comes from Bayesian inference. The features derived from Bayesian estimated values are used for uncertainty quantification in RF, XG Boost classifiers.

5. Results and Discussion

The generated data is split into 80% train, 10% for validation and 10% for testing. The classification results on the data is tabulated for the three data sets

Received: August 02, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

		RANDOM FOREST (ACCURACY %)	XG BOOST (ACCURACY %)
Data set 1	ORIGINAL TEST DATA	87.80	87.80
	WITH BAYESIAN TEST DATA	82.80	84.80
Data set 2	ORIGINAL TEST DATA	90.80	92.00
	WITH BAYESIAN TEST DATA	89.40	90.20
Data set 3	ORIGINAL TEST DATA	99.60	99.60
	WITH BAYESIAN TEST DATA	99.60	99.60

The data set 1 contains polynomials with moderate complexity. Around 15% to 20% of the generated samples contain three real roots. The negative b_2 coefficient and moderate b_3 coefficient makes the inverse problem harder.

It is observed that both RF and XG Boost perform similarly on this data set. However Bayesian optimization reduces the accuracy for both RF and XG Boost suggesting that Bayesian tuning is not well suited for this dataset's parameter ranges, possibly overfitting or getting stuck in local optima.

For the data set 2, the accuracy has increased with XG boost achieving the highest accuracy of 92% XG boost is more robust to wide ranges in the values of the coefficients b_1, b_2 and the Bayesian tuning does not improve the performance further. It is to be noted that around 35% to 40% of the samples have three real roots for the samples generated in this data set. The accuracy is considerably good compared to the multiple roots and complexity of the inverse problem.

For the data set 3, both the classifiers achieve near perfect accuracies in all cases. The dominant cubic term (b_3) makes the inverse problem easier for both ML models and Bayesian optimization. Considering the computational cost of Bayesian optimization, it is not worthwhile to utilize the Bayesian framework to improve accuracy.

To make the samples more complex and to simulate situations where there are more real roots, we further test the algorithm on the data set where the coefficient takes random values in the intervals as given below:

$$b_0 \in [-1.0,1.0]; b_1 \in [-2.0,20.]; b_2 \in [-5.0,5.0]; b_3 \in [0.01,0.2];$$

In this situation the samples lead to a situation in which 80% - 85% of the cases have three real roots. On the application of the algorithm, we observe that both Random Forests and XG boost classifier give an accuracy of 90% and the Bayesian Optimization in both the cases reduce the accuracy by 2%.

5.1. Model performance

The discriminant and Bayesian posterior probabilities are the predictive features for RF, the Gini importance of the discriminant averages to around 0.45 across data sets, while for XG Boost, the gain metric prioritizes posterior probabilities to around 0.38 across the data sets. The monotonicity indicator aids in distinguishing unimodal cases.

The XG Boost performs better across all the data sets due to its ability to model nonlinear interactions via the second order gradients which informs tree splits. Its effectiveness in capturing the interaction leads to an improvement of 1.2% accuracy over RF. The inference time is found to be 15 - 20% faster due to optimized tree construction and pruning with computational complexity $O(T.p.logp)$ versus RF's $O(K.p.d)$ where T, K are the number of trees, 'p' is the number of samples and 'd' is the maximum tree depth.

Received: August 02, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6. Conclusion

The hybrid approach presented in the paper, combines a regularized Bayesian framework with Random Forest and XG Boost to effectively address the inverse problem for cubic sensor transformations. The XG boost is found to perform relatively better due to its robust handling of feature interactions and faster inference. The Bayesian Optimization marginally brings down the accuracy in multimodal cases and is computationally expensive and it is opined to be confined to cases where uncertainty quantification is required. The future work could explore adaptive regularization schemes or deep learning models to further enhance root selection accuracy.

References

1. W-H.Zhang, L.Dai, W.Chen, A.Sun, W-LZhu and B-F Ju Novel information-driven smoothing spline linearization method for high-precision displacement sensors based on information criterions Sensors, 23,9268, 2023
2. I. N. Barinov , V.A. Tikhonenkov, V.S. Volkov, and E.V.Kuchumov,Inverse Problem of Approximation for a Polynomial Cubic Transformation Function for a Sensor, Measurements Techniques, 59(2),127-132, 2016
3. FG Waqar, S Patel, CM Simon, A tutorial on the Bayesian statisticalapproach to inverse problemsAPL Machine Learning., 1, 041101, 2023
4. O.J.Maclaren, R.Nicholson, What can be estimated? Identifiability, estimability, causal inference and ill-posed inverse problems. arXiv:1904.02826,2020.
5. J. Hadamard, Sur les Problemes aux DeriveesPartielles et Leur Signification Physique. Princeton University Bulletin, 49-52, 1902
6. S. I. Kabanikhin, Definitions and examples of inverse and ill-posed problems, J.Inv.Ill-Posed Problems,16,217-357,2008
7. M. Imani, A Beikmohammadi and H R Arabnia,Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels Technologies, 13,88,2025
8. L.Breiman; Random Forest, Machine Learning, 45,5-32, 2001
9. J. Friedman, Greedy function approximation: A gradient boosting machine, The Annals of statistics, 29 (5), 2001
10. T Chen and C.Guestrin, XGBoost: A Scalable Tree Boosting System Proceedings of the 22nd ACM SIGKDD International conference on knowledge Discovery and Data Mining, 785-794, 2016.
11. M Wiens, A.Verone-Boyle, N.henscheid , J.T Podichetty and J.Burton, A Tutorial and Use Case Example of the eXtreme Gradient Boosting (XGBoost) Artificial Intelligence Algorithm for Drug Development Applications**, C**linical and Translational Science, 18:e70172,2025
12. I.Billionis and N Zabaras, Solution of inverse problems with limited forward solver evaluations: a Bayesian perspective Inverse Problems, 30,015004,2014.
13. A Mohammad - Djofari; Regularization, Bayesian Inference, and Machine Learning Methods for Inverse Problems, Entropy,23,1673,2021
14. P C Hansen; Discrete Inverse Problems: Insight and Algorithms, Society for Industrial and Applied Mathematics, 2010
15. S Brooks, A Gelman, G Jones, X-L Meng; Handbook of Markov Chain Monte Carlo, 1st Edison, Chapman & Hall/CRC, Newyork, 2011.

Received: August 02, 2025