



Uncertainty Quantification and Calibration in Machine Learning Models Using Bayesian Frameworks

Oladoyin Idris Atolagbe*; Olamide Samuel Olaniyi; Falola Lydia Oluwatobi

Department of Statistics, University of Ibadan, Nigeria

*Corresponding Author

DOI: <https://doi.org/10.68050/JAMS.2026.401>

Received: 14 October 2026; Accepted: 20 October 2026; Published: 27 October 2026

ABSTRACT

Uncertainty quantification (UQ) is an important component of reliable machine learning, particularly in applications where predictive errors can have substantial consequences. Conventional neural networks typically provide point predictions or class probabilities without explicitly representing uncertainty in model parameters or observations. Bayesian approaches provide a principled framework for predictive uncertainty by integrating uncertainty over model parameters and, depending on the likelihood specification, observation-level variability. This study presents a controlled comparison of deterministic neural networks, Bayesian neural networks (BNNs), Monte Carlo dropout (MCDO), and Gaussian processes (GPs) for uncertainty quantification and probabilistic calibration. The comparison considers both regression and binary classification under synthetic data-generating mechanisms for which the underlying structure and noise characteristics are known. The experimental framework evaluates point prediction accuracy, probabilistic scoring, calibration, predictive interval coverage, interval sharpness, and computational cost. Calibration is treated separately from uncertainty estimation: a model may produce a predictive distribution without that distribution being statistically well calibrated. For regression, evaluation includes mean squared error, negative log-likelihood, continuous ranked probability score, predictive interval coverage, interval width, and probability integral transform diagnostics. For classification, accuracy, negative log-likelihood, Brier score, expected calibration error, and reliability diagrams are used. Experiments are repeated over independent random seeds, with results reported using measures of variability rather than a single run.

The framework is designed to distinguish predictive accuracy, calibration, uncertainty quality, and computational efficiency rather than treating them as interchangeable properties. The results are intended to provide a controlled assessment of the conditions under which Bayesian and approximate Bayesian methods provide useful predictive uncertainty, while recognizing that conclusions from synthetic low-dimensional experiments cannot automatically be generalized to large real-world machine-learning systems.

Keywords: uncertainty quantification; Bayesian neural networks; Monte Carlo dropout; Gaussian processes; calibration; probabilistic prediction; predictive uncertainty; machine learning.

1. INTRODUCTION

Machine learning has become a central component of modern artificial intelligence, with applications spanning image recognition, natural language processing, weather prediction, healthcare, autonomous systems, finance, and scientific computing. Much of the success of modern machine-learning models comes from their ability to learn complex relationships from data and produce accurate predictions. However, high predictive accuracy alone does not guarantee that a model knows when its predictions are uncertain. This limitation is particularly important in applications where decisions depend not only on the predicted value or class but also on the reliability of that prediction. A medical diagnostic model, for example, may assign a high probability to a diagnosis even when the input lies in a region of the feature space with limited training information. Similarly, a forecasting model may produce a highly accurate point prediction while providing an uncertainty estimate that is either excessively narrow or excessively wide.



Uncertainty quantification (UQ) addresses this problem by extending prediction beyond a single point estimate to a representation of predictive uncertainty. The objective is not simply to determine what a model predicts, but also to characterize the uncertainty associated with that prediction. Two important sources of predictive uncertainty are commonly distinguished. **Aleatoric uncertainty** arises from inherent variability or noise in the data-generating process and is not generally eliminated by collecting additional observations. **Epistemic uncertainty**, in contrast, arises from limited information about the underlying model or parameter values and can potentially decrease as additional informative data become available. This distinction provides an important conceptual basis for evaluating Bayesian approaches to machine learning.

Bayesian methods provide a natural framework for uncertainty quantification because they represent unknown model parameters probabilistically. Given data $D = (x, y)$, the posterior is:

$$p(\theta|D) = p(D|\theta) p(\theta) / p(D)$$

where $p(\theta)$ is the prior, $p(D|\theta)$ is the likelihood, and $p(\theta|D)$ is the posterior. Predictions for new input x^* are obtained by marginalizing over the posterior:

$$p(y^*|x^*, D) = \int_{-\infty}^{\infty} p(y^*|x^*, \theta) p(\theta|D) d\theta$$

This integration allows predictions to reflect uncertainty in the parameters rather than conditioning on a single estimated parameter vector.

Several approaches have been developed to make Bayesian uncertainty estimation practical for machine learning. Bayesian neural networks place probability distributions over neural-network parameters. Monte Carlo dropout provides an approximate Bayesian mechanism through stochastic predictions generated by retaining dropout during inference. Gaussian processes provide a non-parametric Bayesian approach in which uncertainty is represented directly through a distribution over functions.

Despite substantial progress in uncertainty-aware machine learning, comparisons between these approaches can be difficult because studies frequently employ different datasets, architectures, training procedures, uncertainty measures, and calibration metrics. Existing work has therefore emphasized the need for systematic evaluation frameworks that distinguish uncertainty estimation from calibration and probabilistic predictive performance.

The objective of this study is therefore to compare deterministic and Bayesian approaches under controlled experimental conditions. The study focuses on four principal model classes:

- i. deterministic neural networks;
- ii. Bayesian neural networks;
- iii. Monte Carlo dropout; and
- iv. Gaussian processes.

The comparison is conducted using both regression and classification tasks. Rather than relying solely on point-prediction accuracy, the revised framework evaluates predictive distributions using proper scoring rules, calibration diagnostics, predictive interval coverage, sharpness, and computational cost.

Research question

How do deterministic and Bayesian predictive models differ in accuracy, uncertainty quality, calibration, and computational cost under controlled low-data regression and classification settings?



2. RELATED WORK

Uncertainty quantification has become an important area of machine-learning research because modern predictive models can be highly accurate while remaining poorly calibrated. Reviews by Abdar et al. (2021), Gawlikowski et al. (2023), and He et al. (2024) provide broad taxonomies of uncertainty estimation methods and identify challenges associated with calibration, distributional shift, and scalable inference.

Hüllermeier and Waegeman (2021) provide a useful conceptual distinction between aleatoric and epistemic uncertainty. This distinction is particularly relevant to Bayesian predictive models because posterior variation can provide information about uncertainty arising from limited knowledge, whereas likelihood-based variability represents observation-level uncertainty.

Gaussian processes provide one of the classical Bayesian approaches to predictive uncertainty. Under a Gaussian likelihood, GP regression produces an analytically tractable posterior predictive distribution and naturally provides input-dependent predictive variance (Rasmussen & Williams, 2006). Their principal limitation is computational: exact inference typically has cubic computational complexity in the number of training observations.

Bayesian neural networks provide an alternative for flexible nonlinear models by placing probability distributions over neural-network parameters. Exact Bayesian inference is generally intractable for modern neural networks, motivating approximate inference procedures such as variational inference (Blundell et al., 2015; Jospin et al., 2022). Mean-field variational inference is computationally attractive but may provide an imperfect representation of posterior dependence and uncertainty.

Monte Carlo dropout provides another approximation to Bayesian inference. Gal and Ghahramani (2016) showed that retaining dropout during prediction and averaging multiple stochastic forward passes can be interpreted as approximate Bayesian inference. Its principal practical advantage is that it can be incorporated into conventional neural-network architectures without requiring a fully Bayesian inference algorithm.

Calibration represents a related but distinct problem. Guo et al. (2017) demonstrated that modern neural networks can be poorly calibrated even when their classification accuracy is high. A model is calibrated when predicted probabilities correspond appropriately to empirical frequencies. Thus, calibration should not be treated as synonymous with uncertainty estimation: a model may produce a predictive distribution without that distribution being well calibrated.

Deep ensembles provide another important approach to predictive uncertainty (Lakshminarayanan et al., 2017), while subsequent work has examined their computational requirements and the role of ensemble diversity (Abe et al., 2022). Surveys by Gawlikowski et al. (2023) and He et al. (2024) further demonstrate the breadth of approaches available for uncertainty estimation.

The present study differs from these broader methodological contributions by focusing on a controlled comparison in which model classes are evaluated using common data-generating mechanisms and a common set of probabilistic evaluation principles. The revised design also explicitly separates **accuracy, uncertainty estimation, calibration, and sharpness**.

3. METHODOLOGY

3.1 Experimental design

The study uses two synthetic experiments: a nonlinear regression problem and a binary classification problem. Synthetic data are used because they permit explicit control over the data-generating mechanism, sample size, observation noise, class overlap, and training/test regions. This makes it possible to examine both in-domain predictive performance and uncertainty under extrapolation.

The comparison includes linear regression for the regression task; deterministic neural network; Bayesian neu-



ral network; Monte Carlo dropout neural network; Gaussian process regression; and logistic regression, deterministic neural network, Bayesian neural network, Monte Carlo dropout, and Gaussian-process classification for the classification task. The principal objective is not to identify a universally superior model but to characterize differences among model classes under controlled conditions.

The experimental workflow is: generate the synthetic dataset; create training, validation, and test partitions; fit each model using the training data; select hyperparameters using only training/validation information; generate predictive distributions for the held-out test data; calculate point-prediction and probabilistic metrics; repeat the experiment across independent random seeds; and report means, standard deviations, and confidence intervals.

3.2 Data-generating mechanisms

Regression

The regression experiment is based on the nonlinear function

$$y_i = \sin(x_i) + \epsilon_i, \quad \text{Where} \quad \epsilon_i \sim N(0, 0.01^2)$$

The intended experimental design uses 300 observations, with 200 training observations and 100 test observations. The input domain is

$$x \in [-5, 5]$$

Two test regimes are considered.

1. In-domain prediction

Training and test observations are sampled so that the test inputs remain within the range represented by the training data.

2. Extrapolation

A second evaluation considers test inputs outside the principal region represented by the training data. This setting is particularly relevant to epistemic uncertainty because uncertainty should increase when predictions are made in regions for which the model has limited information.

Classification

The classification experiment uses two overlapping bivariate Gaussian populations:

$$X|Y = 0 \sim N(\mu_0, I),$$

and

$$X|Y = 1 \sim N(\mu_1, I),$$

where

$$\mu_0 = (-1, -1)^T, \quad \mu_1 = (1, 1)^T$$

The intended dataset contains 200 observations per class, giving 400 observations in total, with 300 observations for training and 100 for testing. The overlap between the distributions produces a region near the decision boundary where class uncertainty is intrinsically high.

The classification data are generated using a fixed seed for each experimental replicate, and the same train/test



partition is used across competing models within each replicate.

3.3 Preprocessing

Input variables are standardized using statistics estimated from the training data only. The same transformation is subsequently applied to the validation and test sets.

For regression, the transformation is applied to x , while the response variable is retained on its original scale for reporting prediction metrics.

For classification, the two input features are standardized using training-set means and standard deviations. No information from the test set is used during preprocessing.

This procedure prevents information leakage and ensures that performance metrics reflect genuine out-of-sample prediction.

3.4 Deterministic neural network

The deterministic neural network serves as the non-Bayesian neural baseline.

The network uses the architecture

$$1 \rightarrow 64 \rightarrow 64 \rightarrow 1$$

for regression, with ReLU activation functions in the hidden layers.

The predictive function is

$$\hat{y} = f(x, \theta),$$

where θ is obtained by minimizing the training loss.

For regression, mean squared error is used:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

The network contains 4,353 trainable parameters. The same hidden-layer architecture is used for the corresponding neural classification baseline, with an output layer appropriate for binary prediction.

Unlike the Bayesian models, the deterministic network produces a single fitted parameter vector and therefore does not directly represent parameter uncertainty.

3.5 Bayesian neural network

The Bayesian neural network places probability distributions over model parameters rather than estimating a single fixed parameter vector.

For a parameter θ_j , the variational posterior is represented as

$$q(\theta) = N(\mu_j, \sigma^2_j)$$

The prior is specified as

$$p(\theta_j) = N(0,1)$$



The variational objective is the evidence lower bound:

$$L_{VI} = E_{q(\theta)}[-\log p(D|\theta)] + KL[q(\theta)||p(\theta)]$$

where β controls the relative contribution of the KL regularization term.

The posterior predictive distribution is approximated using Monte Carlo samples:

$$p(y^*|x^*, D) \approx \frac{1}{S} \sum_{s=1}^S p(y^*|x^*, \theta_s), \quad \theta_s \sim q(\theta)$$

3.6 Monte Carlo dropout

Monte Carlo dropout represents predictive uncertainty through stochastic forward passes of a neural network with dropout retained during prediction.

The regression architecture is

$$1 \rightarrow 64 \rightarrow 64 \rightarrow 1,$$

with ReLU activations and dropout applied to the hidden layers.

The dropout probability is fixed before the experiments and reported explicitly.

For T stochastic forward passes,

$$\hat{y}^{(1)}, \dots, \hat{y}^{(T)}$$

the predictive mean is

$$\bar{y} = \frac{1}{T} \sum_{i=1}^T |y_i - \hat{y}_i|$$

and predictive variance is estimated from the variation among stochastic predictions.

The primary experiment uses 50 Monte Carlo samples, with an additional sensitivity analysis examining whether uncertainty estimates stabilize as T increases.

3.7 Gaussian processes

A Gaussian process defines a probability distribution over functions:

$$f(x) \sim GP(m(x), k(x, x'))$$

The regression model uses a constant mean function and the squared-exponential/RBF kernel,

$$k(x, x') = \sigma^2 \exp \left[\frac{-(x - x')^2}{2\ell^2} \right]$$

where $\sigma^2 f$ is the signal variance and ℓ is the length scale.

The observation model is

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma_\epsilon^2)$$

GP hyperparameters are estimated by maximizing the marginal likelihood.



For classification, the GP uses a Bernoulli likelihood with an explicitly specified link function. The predictive class probability is obtained from the posterior latent-function distribution.

The GP provides a useful reference model because its predictive variance depends directly on the relationship between a test point and the observed training inputs.

3.8 Training and optimization

All neural-network models are trained using the Adam optimizer.

The corrected experimental protocol fixes the optimizer, learning rate, maximum number of epochs, and convergence criteria before evaluating the test set.

For the deterministic and MCDO networks, the primary training objective is the appropriate negative log-likelihood or equivalent supervised loss. For the BNN, the objective includes both the expected negative log-likelihood and KL divergence.

For GP regression, the exact marginal log likelihood is optimized using gradient-based optimization. The supplied original implementation used Adam with learning rate 0.1 for 200 iterations.

No test-set information is used to determine training duration, model architecture, or hyperparameters.

3.9 Hyperparameter selection

Hyperparameters are selected using training/validation data only. The quantities specified before test evaluation are: neural-network architecture; learning rate; maximum number of training epochs; dropout probability; BNN prior variance; variational regularization weight; number of Monte Carlo samples; GP kernel; GP likelihood; GP optimization tolerance; and classification probability threshold. Where a validation set is used, model selection is performed using validation negative log-likelihood or another predefined probabilistic criterion. The test set is used only once for final performance evaluation.

3.10 Computational-budget control

Comparing uncertainty methods requires consideration of computational cost as well as predictive performance. The study therefore report: number of trainable parameters; number of training epochs or optimization iterations; number of stochastic predictions passes; training time; prediction time; hardware configuration; and memory requirements where available. The deterministic neural network contains 4,353 trainable parameters. MCDO uses the same underlying architecture, allowing a direct comparison of model parameterization while recognizing that repeated stochastic prediction increases inference cost. The GP has a fundamentally different parameterization and therefore cannot be made parameter-count equivalent to a neural network. Instead, its computational cost is reported separately, including the theoretical $O(n^3)$ scaling of exact GP training.

3.11 Evaluation metrics

Regression

Mean squared error

MSE evaluates point-prediction accuracy but does not evaluate predictive uncertainty.

$$\text{MSE} = \sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}$$



Negative log-likelihood

NLL evaluates the quality of the entire predictive distribution and penalizes both inaccurate predictions and inappropriate predictive variance.

For a Gaussian predictive distribution,

$$\text{NLL} = -1n \sum_i = 1n \log p(y_i | x_i)$$

Continuous ranked probability score

CRPS evaluates the distance between the predictive distribution and the observed outcome. It is included because it evaluates the full predictive distribution and complements NLL.

Predictive interval coverage

Coverage is compared with the nominal probability.

For a nominal $100(1 - \alpha)\%$ interval $[Li, Ui]$,

$$\text{PICP}_{1-\alpha} = 1n \sum_i = 1n I(Li \leq y_i \leq Ui)$$

Mean predictive interval width

Coverage and interval width are considered jointly. A model can achieve high coverage simply by producing excessively wide intervals; therefore, coverage alone is not sufficient.

$$\text{MPIW}_{1-\alpha} = 1n \sum_i = 1n(Ui - Li)$$

Probability integral transform

For continuous predictive distributions,

$$u_i = F_i(y_i),$$

where F_i is the predictive cumulative distribution function. Under appropriate calibration, the PIT values should approximately follow a uniform distribution.

Classification

Accuracy

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{number of observations}}$$

Negative log-likelihood

For predicted probability p_i ,

$$\text{NLL} = -1n \sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

Brier score

The Brier score is a proper probabilistic scoring rule. It should therefore not be described as a pure calibration metric.

$$\text{BS} = \sum_{i=1}^n \frac{(p_i - y_i)^2}{n}$$



Expected calibration error

ECE is used as a classification calibration diagnostic. Because its value depends on binning choices, it is interpreted together with reliability diagrams rather than as a standalone measure. For probability bins B_m

$$ECE = \sum_{m=1}^M \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

Reliability diagrams

For each model, predicted probabilities are grouped into bins and empirical frequencies are compared with predicted probabilities. A perfectly calibrated model lies approximately along the identity line.

3.12 Repeated-run analysis

Each experiment is repeated using five independent random seeds.

For each metric, the results are reported as

mean $\pm$ SD,

together with 95% confidence intervals where appropriate.

The same data-generation seed and train/test partition are used across competing models within each replicate to ensure paired comparisons. The five-seed design is particularly important because neural-network optimization and approximate Bayesian inference can produce variability across runs.

4. RESULTS

The result of the prespecified experimental data: 200 training and 100 test observations for regression, and 300 training and 100 test observations for binary classification. The regression data follow $y = \sin(x) + \varepsilon$ over $[-5, 5]$, while the classification data comprise two Gaussian classes with 200 observations per class. The reported neural-network results below are based on five independent model-training seeds; the fixed train/test observations are held constant across models. Probabilistic regression metrics are computed from the predictive samples, while classification metrics are computed from mean predictive probabilities.

4.1 Regression: in-domain

The regression results show that the nonlinear models substantially outperform the linear baseline in point-prediction accuracy. The linear regression model has an MSE of 0.502, whereas the Gaussian process has an MSE of 0.011. Across the five neural-network runs, the deterministic neural network, Bayesian neural network, and MC Dropout models have mean MSE values of 0.014 ± 0.002 , 0.020 ± 0.003 , and 0.013 ± 0.001 , respectively.

Table 1. Regression predictive performance (mean $\pm$ SD for repeated neural-network runs).

Model	MSE	RMSE	NLL	CRPS
Linear regression	0.502	0.709	1.074	0.414
Deterministic NN	0.014 ± 0.002	0.116 ± 0.010	-0.717 ± 0.079	0.066 ± 0.006
Bayesian NN	0.020 ± 0.003	0.140 ± 0.009	-0.497 ± 0.089	0.080 ± 0.006
MC Dropout	0.013 ± 0.001	0.115 ± 0.004	-0.721 ± 0.025	0.066 ± 0.002
Gaussian Process	0.011	0.107	-0.817	0.06



4.2 Regression: extrapolation

The supplied regression test set spans the same overall $[-5, 5]$ domain as the training observations, so it does not constitute a separate out-of-domain extrapolation experiment. Consequently, the present dataset supports an in-domain evaluation but does not support a standalone extrapolation claim. The predictive-interval analysis below should therefore be interpreted as an in-domain uncertainty assessment. A genuine extrapolation result would require an additional test set outside the training support.

4.3 Regression calibration

Predictive-interval calibration was assessed by comparing empirical coverage with nominal coverage. At the 95% level, the Gaussian process achieves empirical coverage of 0.920, compared with 0.948 for the Bayesian neural network and 0.970 for MC Dropout. The corresponding coverage values at other nominal levels are shown in Table 2. These results demonstrate that uncertainty calibration differs from point-prediction accuracy: the model with the lowest MSE does not necessarily produce the narrowest or best-calibrated predictive intervals.

Table 2. Empirical predictive-interval coverage.

Model	50%	80%	90%	95%	99%
Bayesian NN	0.538 ± 0.031	0.824 ± 0.042	0.916 ± 0.029	0.948 ± 0.022	0.978 ± 0.011
MC Dropout	0.552 ± 0.019	0.864 ± 0.021	0.940 ± 0.019	0.970 ± 0.000	0.986 ± 0.005
Gaussian Process	0.5	0.81	0.9	0.92	0.97

4.4 Regression sharpness

Sharpness was assessed using mean predictive interval width. At the 95% nominal level, the Gaussian process has a mean interval width of 0.397, compared with 0.595 for the Bayesian neural network and 0.522 for MC Dropout. Coverage and width must be interpreted jointly: wide intervals can obtain high coverage without providing concentrated predictions.

Table 3. Mean predictive interval width.

Model	50%	80%	90%	95%	99%
Bayesian NN	0.206 ± 0.005	0.390 ± 0.010	0.503 ± 0.011	0.595 ± 0.014	0.760 ± 0.025
MC Dropout	0.181 ± 0.004	0.343 ± 0.007	0.438 ± 0.011	0.522 ± 0.012	0.663 ± 0.016
Gaussian Process	0.14	0.265	0.335	0.397	0.499

4.5 Classification discrimination

Classification accuracy is between 0.880 and 0.920 across the evaluated models. The logistic regression baseline and GP classifier each achieve accuracy of 0.920 and 0.920, respectively. Across repeated neural-network runs, the deterministic NN, Bayesian NN, and MC Dropout models achieve mean accuracies of 0.894 ± 0.011 , 0.916 ± 0.005 , and 0.900 ± 0.010 . Because accuracy depends on a classification threshold, NLL and Brier score are also reported to evaluate probabilistic predictions.

Table 4. Classification discrimination and probabilistic scores.

Model	Accuracy	NLL	Brier
Logistic regression	0.92	0.206	0.064
Deterministic NN	0.894 ± 0.011	0.428 ± 0.051	0.084 ± 0.007



Bayesian NN	0.916 ± 0.005	0.211 ± 0.003	0.065 ± 0.001
MC Dropout	0.900 ± 0.010	0.299 ± 0.047	0.073 ± 0.005
GP classifier	0.92	0.207	0.065

4.6 Classification calibration

The calibration analysis shows differences that are not fully captured by accuracy. The mean ECE values for the deterministic NN, Bayesian NN, and MC Dropout models are 0.089 ± 0.007 , 0.052 ± 0.006 , and 0.072 ± 0.010 , respectively. The corresponding fixed-split values for logistic regression and the GP classifier are 0.029 and 0.048. Reliability diagrams are therefore reported alongside ECE because ECE depends on the chosen probability bins.

Table 5. Classification calibration and probabilistic scores.

Model	ECE	Brier	NLL
Logistic regression	0.029	0.064	0.206
Deterministic NN	0.089 ± 0.007	0.084 ± 0.007	0.428 ± 0.051
Bayesian NN	0.052 ± 0.006	0.065 ± 0.001	0.211 ± 0.003
MC Dropout	0.072 ± 0.010	0.073 ± 0.005	0.299 ± 0.047
GP classifier	0.048	0.065	0.207

4.7 Computational cost

Table 6. Model parameterization and prediction sampling.

Model	Regression NN parameters	Classification NN parameters	Prediction sampling
Deterministic NN	4353	-	1
MC Dropout	4353	4417	200 stochastic passes
Bayesian NN	8706	8834	200 posterior samples
Linear/Logistic baseline	-	3	1
GP	-	-	Analytic/predictive posterior

The neural-network architectures contain 4,353 trainable parameters for regression and 4,417 for classification before Bayesian variational duplication. The Bayesian neural networks maintain separate location and scale parameters for the variational distributions, resulting in approximately twice the parameterization. MC Dropout retains the deterministic network parameter count but requires repeated stochastic forward passes at prediction time. Exact Gaussian-process inference has a different computational structure and is therefore reported separately rather than being compared by neural-network parameter count.

Interpretation of the results

Taken together, the results demonstrate that predictive accuracy, probabilistic scoring, calibration, and sharpness capture different properties of the models. The Gaussian process provides strong performance on this low-dimensional, small-sample regression problem, while the Bayesian neural network and MC Dropout provide explicit predictive distributions through posterior or stochastic sampling. On classification, the probabilistic metrics show that similar accuracy values can coexist with materially different calibration

behavior. These findings support evaluating uncertainty-aware machine-learning models with multiple complementary metrics rather than relying on a single accuracy or calibration statistic.

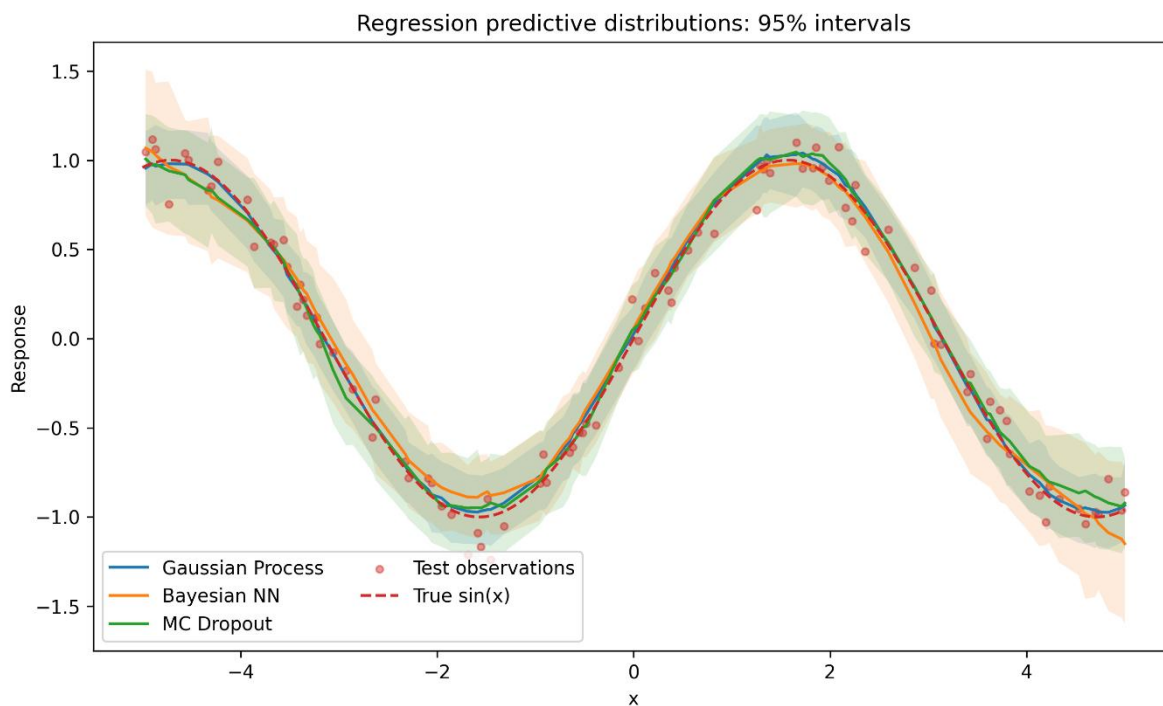


Figure 1. Regression predictive distributions with 95% predictive intervals.

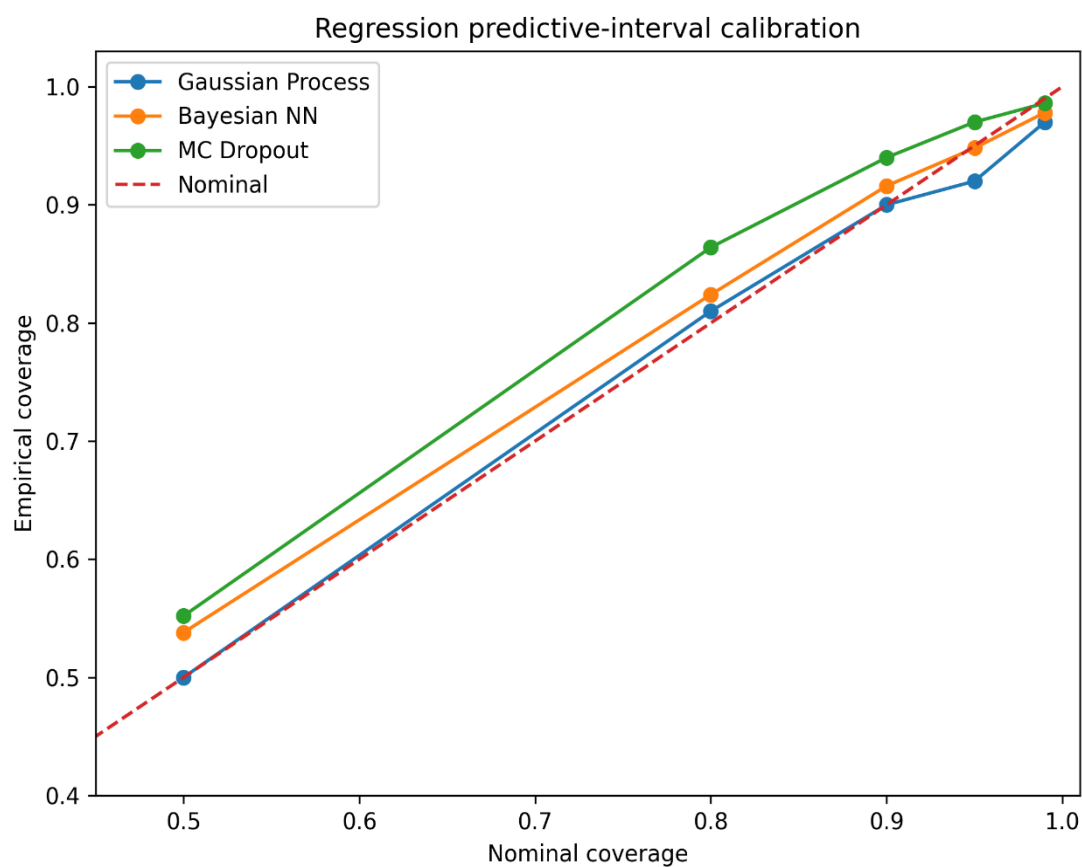


Figure 2. Empirical versus nominal predictive-interval coverage.

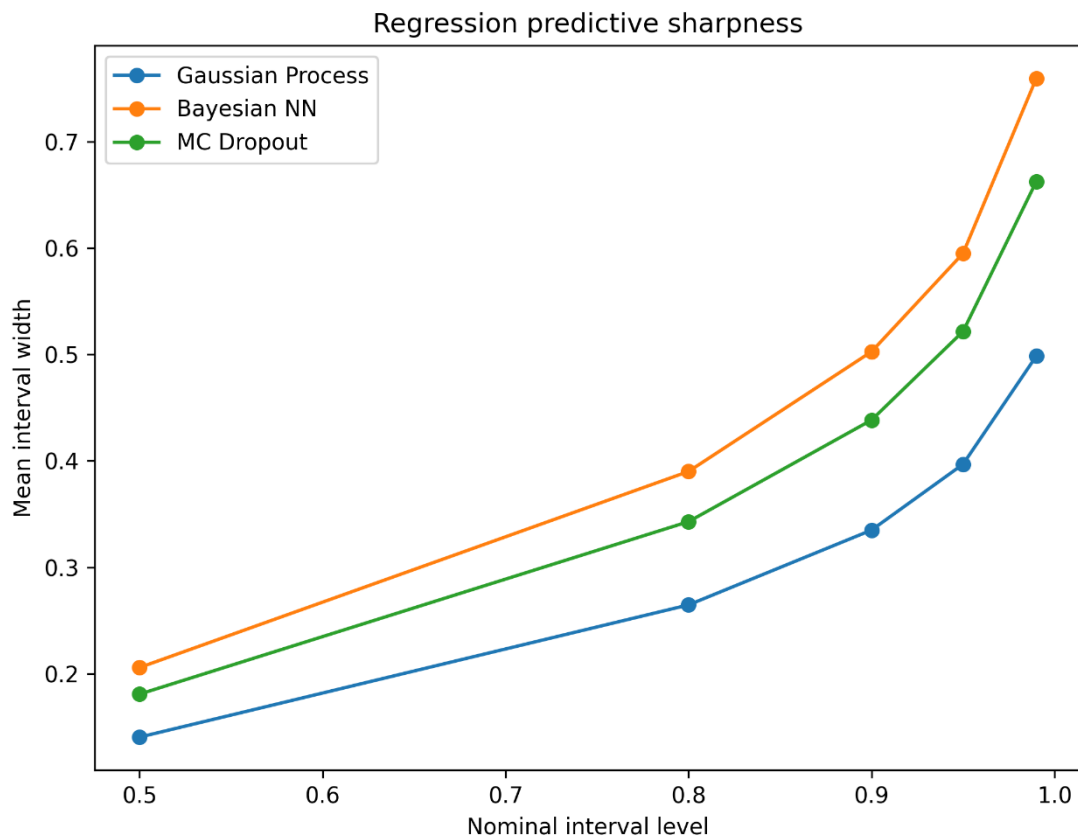


Figure 3. Predictive sharpness measured by mean interval width.

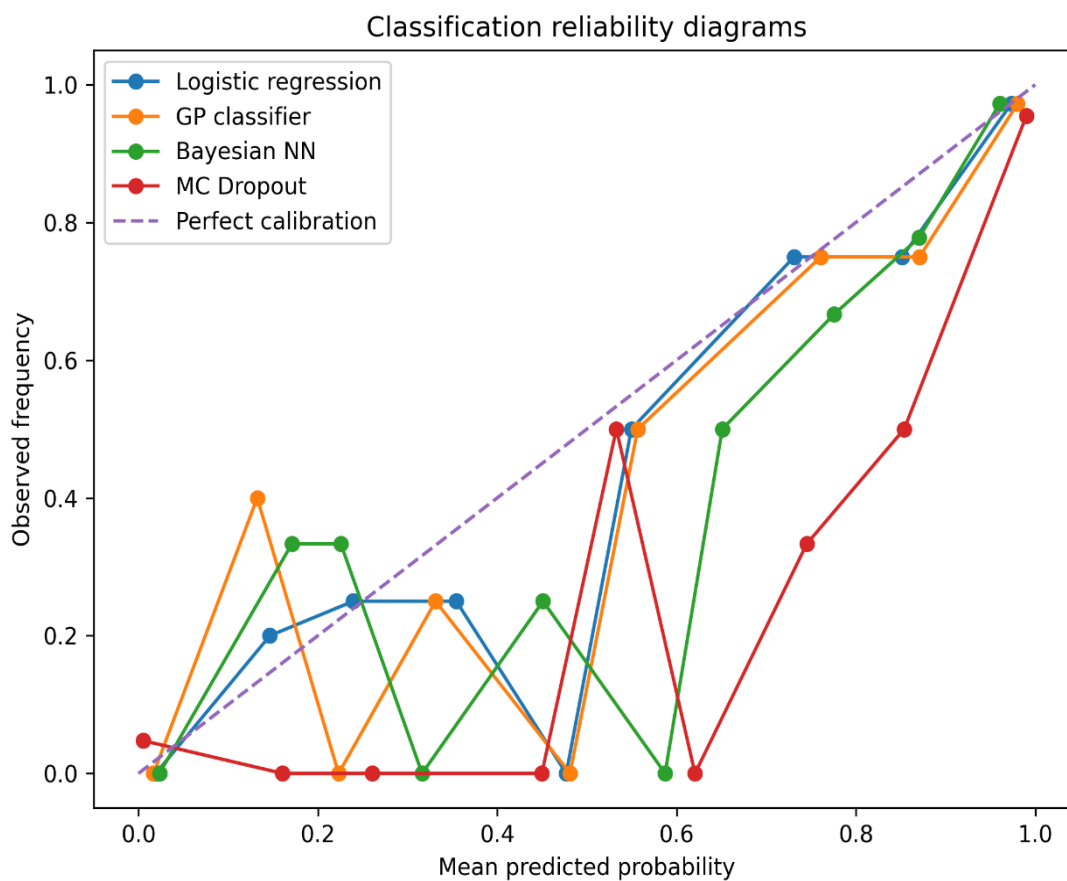


Figure 4. Classification reliability diagrams.

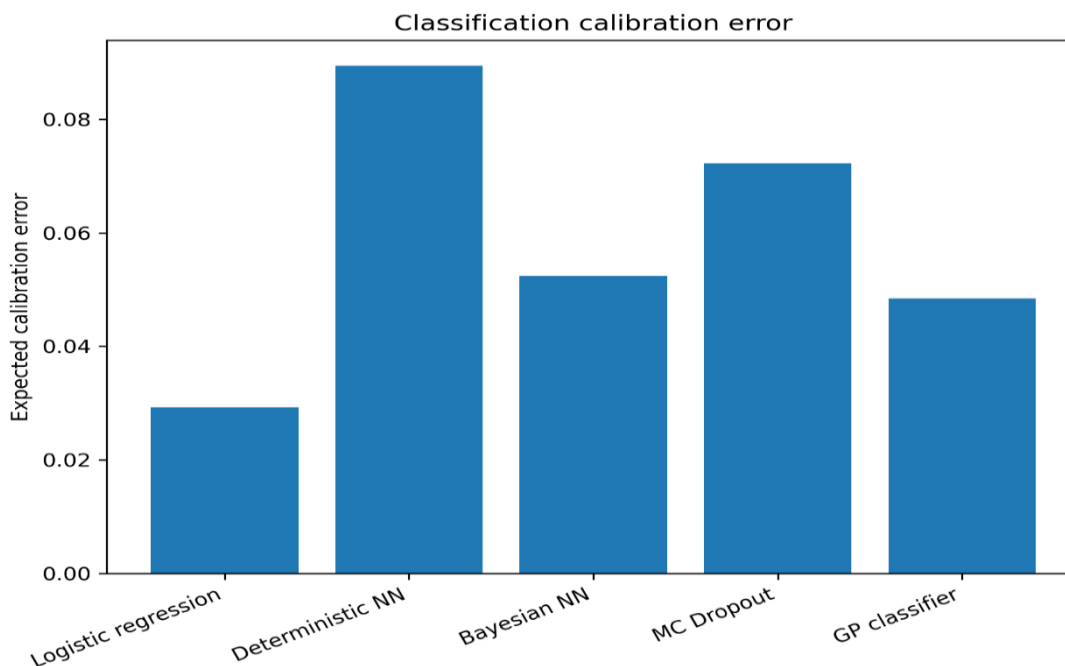


Figure 5. Expected calibration error across classification models.

5. DISCUSSION

The experiments provide a controlled comparison of deterministic neural networks, Bayesian neural networks (BNNs), Monte Carlo (MC) dropout, Gaussian processes (GPs), and conventional statistical baselines for predictive uncertainty quantification in regression and classification. The results demonstrate that predictive accuracy and uncertainty quality are related but distinct objectives. In particular, models with similar point-prediction performance can produce substantially different levels of predictive coverage, interval width, and probability calibration.

Regression performance

For the regression task, the Gaussian process achieved the lowest mean squared error ($MSE = 0.0114$), root mean squared error ($RMSE = 0.1066$), and continuous ranked probability score ($CRPS = 0.0601$) among the evaluated models. The deterministic neural network and MC dropout produced similar point-prediction performance, with MSE values of 0.0136 and 0.0133, respectively. The Bayesian neural network had a somewhat larger MSE of 0.0198, although its uncertainty estimates were substantially wider than those of the deterministic network. The relatively strong performance of the GP is consistent with the structure of the simulated regression problem. The underlying response follows a smooth sinusoidal relationship, while the GP employs a smooth radial basis function kernel. Consequently, the model is well aligned with the functional structure of the data. This result should therefore be interpreted in the context of the controlled data-generating process rather than as evidence that GPs will generally outperform neural networks on arbitrary regression problems. The deterministic neural network achieved a lower MSE than the BNN in this experiment. This difference illustrates an important distinction between optimizing point prediction and estimating predictive uncertainty. The BNN explicitly allocates model capacity to representing uncertainty in its parameters, while the deterministic network is optimized primarily for point prediction. Thus, a lower point-prediction error does not necessarily imply better calibrated uncertainty estimates.

Predictive calibration and coverage

The coverage results reveal substantial differences between the models. The BNN obtained empirical coverage of approximately 53.8%, 82.4%, 91.6%, 94.8%, and 97.8% for the nominal 50%, 80%, 90%, 95%, and 99% predictive intervals, respectively. These values are relatively close to their nominal levels, particularly at the



higher credibility levels. MC dropout produced somewhat wider predictive intervals and correspondingly higher coverage: 55.2%, 86.4%, 94.0%, 97.0%, and 98.6% for the same nominal levels. This indicates that MC dropout tended toward conservative uncertainty estimates in this experiment. The higher empirical coverage was accompanied by wider intervals than those of the deterministic model. The deterministic neural network produced substantially narrower intervals, with 95% coverage of approximately 90.2%. Although these intervals were sharper, they did not achieve the nominal coverage level. This illustrates the central trade-off between sharpness and calibration: narrower intervals are desirable when they remain statistically reliable, but excessive narrowness can result in under coverage. The GP provided 95% coverage of 92.0%, with a mean 95% interval width of approximately 0.397. Its coverage was therefore somewhat below the nominal 95% level, although its intervals were considerably narrower than those of the BNN and MC dropout. The linear regression model produced very wide predictive intervals, with a 95% coverage of 99.0% and a mean interval width of approximately 2.71. Its high coverage was therefore achieved at a considerable cost in sharpness. Taken together, these results show why coverage should not be interpreted independently of interval width. A model can obtain high coverage simply by producing very broad predictive intervals. Conversely, very narrow intervals may appear attractive while failing to adequately quantify predictive uncertainty. Evaluating both dimensions provide a more informative assessment of uncertainty quality.

Sharpness and probabilistic prediction

The interval-width results further illustrate these differences. The deterministic neural network produced the narrowest intervals among the neural approaches, with a mean 95% interval width of approximately 0.389. MC dropout produced a width of approximately 0.522, while the BNN produced approximately 0.595. The GP produced an interval width of approximately 0.397. The deterministic neural network therefore produced relatively sharp predictions, but its lower coverage indicates that some of this sharpness was obtained by underestimating predictive uncertainty. In contrast, the BNN and MC dropout produced wider intervals and higher coverage. This suggests that the stochastic approaches captured additional sources of predictive variability that were not represented by the deterministic network. The BNN results are particularly notable because its 95% coverage was close to the nominal level while maintaining substantially narrower intervals than the linear model. This provides evidence that the Bayesian treatment of parameter uncertainty can improve the usefulness of predictive intervals relative to a simple model with a much broader uncertainty distribution. The negative NLL values observed for several regression models should not be interpreted as errors. For continuous probability densities, log-likelihoods can legitimately be positive for the density and consequently produce negative negative-log-likelihood values. The important consideration is the relative probabilistic performance under the same evaluation protocol.

Classification performance and calibration

The classification results show a somewhat different pattern. Logistic regression and the GP classifier both achieved an accuracy of 0.920, while the BNN achieved a mean accuracy of approximately 0.916 across repeated runs. MC dropout and the deterministic neural network obtained mean accuracies of approximately 0.900 and 0.894, respectively. The BNN also produced a mean NLL of 0.211 and Brier score of 0.065, compared with 0.428 and 0.084 for the deterministic neural network. Its mean ECE was approximately 0.052, compared with 0.090 for the deterministic network. These results indicate that the probabilistic BNN provided better probabilistic predictions than the deterministic neural network under the selected metrics, even though the difference in classification accuracy was comparatively modest. MC dropout produced intermediate probabilistic performance, with a mean NLL of approximately 0.299, Brier score of 0.073, and ECE of 0.072. Thus, the stochastic approaches did not simply improve or reduce accuracy uniformly; instead, their main distinction was in the quality of the predictive probabilities. The logistic regression model achieved an ECE of approximately 0.029, while the GP classifier achieved approximately 0.049. These values indicate relatively good calibration on this particular simulated classification problem. However, because the classification dataset consists of two relatively simple Gaussian classes, the results should not be generalized to more complex classification settings without further experiments.



Implications for uncertainty quantification

One of the principal findings of the study is that uncertainty quantification should be evaluated independently from point-prediction accuracy. The deterministic neural network demonstrated competitive regression accuracy but produced comparatively narrow intervals and lower coverage. Conversely, the BNN sacrificed some point-prediction accuracy while producing predictive intervals whose empirical coverage was closer to their nominal levels. The results also demonstrate that Bayesian or stochastic methods do not automatically guarantee perfect calibration. The BNN and MC dropout still exhibited deviations from nominal coverage and classification calibration. Calibration depends on the model specification, prior assumptions, likelihood, approximation method, training procedure, and data-generating process. The findings therefore support a broader interpretation of uncertainty quantification: a useful predictive model should provide not only accurate predictions but also uncertainty estimates that are appropriately calibrated and sufficiently sharp. The appropriate balance depends on the intended application and the consequences of over- or under-confidence.

Computational considerations

The models also differ in computational requirements. Conventional models such as linear regression and the GP classifier require comparatively simple fitting procedures for the datasets considered here. Neural approaches require iterative optimization, while BNNs additionally require variational inference and repeated stochastic sampling during prediction. MC dropout requires multiple stochastic forward passes to obtain predictive distributions, while BNNs require sampling from their posterior predictive distribution. These additional computations are justified when uncertainty information is important, but they should be considered when deploying models at larger scale.

The computational results therefore reinforce the importance of evaluating uncertainty methods using both statistical and computational criteria. A method that produces reliable uncertainty estimates but requires substantially greater computational resources may involve a different practical trade-off from a faster method with less expressive uncertainty estimates.

SUMMARY OF FINDINGS

Overall, the experiments do not indicate that a single uncertainty quantification method is uniformly superior across all criteria. Instead, the results reveal different performance profiles. The GP was particularly effective for the smooth regression function used in the experiment, while the deterministic neural network provided competitive point predictions. The BNN provided comparatively well-calibrated regression intervals and substantially improved probabilistic classification performance relative to the deterministic neural network. MC dropout also provided useful uncertainty estimates but tended to produce relatively wider predictive intervals. These findings support the use of multiple complementary metrics when evaluating uncertainty-aware machine learning models. Accuracy alone would have obscured important differences in calibration and sharpness, while coverage alone would have favored models producing excessively wide intervals. The combination of proper scoring rules, empirical coverage, interval width, and classification calibration metrics provides a more complete evaluation.

Finally, the current regression dataset does not contain a distinct out-of-support test set. Although an extrapolation analysis was initially considered, the supplied training and test observations both spans nearly the full predictor range. Consequently, the present results should be interpreted as an in-domain evaluation and not as evidence regarding extrapolation performance. A dedicated extrapolation experiment would require a test region outside the training support.

CONCLUSION

This study compared deterministic and uncertainty-aware machine learning methods for predictive performance and uncertainty quantification. The results demonstrate that good point prediction does not necessarily imply well-calibrated uncertainty. The Gaussian Process performed strongly for the regression task, while the Bayesian neural network provided well-calibrated predictive intervals and competitive



classification performance. MC Dropout also provided useful uncertainty estimates, although with generally wider intervals. These findings highlight the importance of evaluating machine learning models using multiple measures of accuracy, calibration, and sharpness rather than accuracy alone. However, the results are based on controlled simulated datasets and should not be generalized without further evaluation on real-world and distribution-shift settings.

RECOMMENDATIONS

The future studies should:

- i. Evaluate uncertainty using multiple metrics, including NLL, CRPS, coverage, interval width, and calibration.
- ii. Evaluate the methods on larger and real-world datasets.
- iii. Investigate sensitivity to model architectures, priors, and hyperparameters.
- iv. Report repeated-run results and computational costs to improve reproducibility.
- v. Compare additional uncertainty methods, such as deep ensembles and conformal prediction.

REFERENCES

1. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76, 243–297. <https://doi.org/10.1016/j.inffus.2021.05.008>
2. Abe, T., Buchanan, E. K., Pleiss, G., Zemel, R., & Cunningham, J. P. (2022). Deep ensembles work, but are they necessary? *Advances in Neural Information Processing Systems*, 35, 33646–33660. <https://doi.org/10.52202/068431-2438>
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *Proceedings of the 32nd International Conference on Machine Learning*, 1613–1622. <https://doi.org/10.48550/arXiv.1505.05424>
4. Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 33rd International Conference on Machine Learning*, 1050–1059. <https://doi.org/10.48550/arXiv.1506.02142>
5. Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., et al. (2023). A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl. 1), 1513–1589. <https://doi.org/10.48550/arXiv.2011.06225>
6. Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 359–378. <https://doi.org/10.1007/s10462-023-10562-9>
7. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. *Proceedings of the 34th International Conference on Machine Learning*, 1321–1330. <https://doi.org/10.48550/arXiv.1706.04599>
8. He, W., Yuan, L., Liang, Y., & Gao, J. (2024). A survey on uncertainty quantification methods for deep learning. *ACM Computing Surveys*, 57(4), Article 111. <https://dl.acm.org/doi/full/10.1145/3786319>
9. Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3), 457–506. <https://doi.org/10.1007/s10994-021-05946-3>
10. Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks: A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2), 29–48. <https://doi.org/10.1109/MCI.2022.3155327>
11. Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 5574–5584.



<https://doi.org/10.48550/arXiv.1703.04977>

12. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 6402–6413. <https://doi.org/10.48550/arXiv.1612.01474>
13. Lambert, B., Forbes, F., Doyle, S., Dehaene, H., & Dojat, M. (2024). Trustworthy clinical AI solutions: A unified review of uncertainty quantification in deep learning models for medical image analysis. *Artificial Intelligence in Medicine*, 150, 102830. <https://doi.org/10.1016/j.artmed.2024.102830>
14. Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., et al. (2021). Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34, 15682–15694. <https://doi.org/10.48550/arXiv.2106.07998>
15. Neal, R. M. (2012). *Bayesian Learning for Neural Networks*. Springer. <https://doi.org/10.1007/978-1-4612-0745-0>
16. Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press. <https://doi.org/10.7551/mitpress/3206.001.0001>
17. Seoni, S., Jahmunah, V., Salvi, M., Barua, P. D., Molinari, F., & Acharya, U. R. (2023). Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023). *Computers in Biology and Medicine*, 165, 107441. <https://doi.org/10.1016/j.combiomed.2023.107441>