An Optimized Uncertainty-Aware Training Framework for Neural Networks

An Optimized Uncertainty-Aware Training Framework for Neural Networks

Pegah Tabarisaadi, Abbas Khosravi, *Senior Member, IEEE*, Saeid Nahavandi, *Fellow, IEEE*, Miadreza Shafie-Khah, *Senior Member, IEEE*, and João P. S. Catalão, *Fellow, IEEE*

***Abstract*— Uncertainty quantification (UQ) for predictions generated by neural networks (NNs) is of vital importance in safety-critical applications. An ideal model is supposed to generate low uncertainty for correct predictions and high uncertainty for incorrect predictions. The main focus of state-of-the-art training algorithms is to optimize the NN parameters to improve the accuracy-related metrics. Training based on uncertainty metrics has been fully ignored or overlooked in the literature. This article introduces a novel uncertainty-aware training algorithm for classification tasks. A novel predictive uncertainty estimate-based objective function is defined and optimized using the stochastic gradient descent method. This new multiobjective loss function covers both accuracy and uncertainty accuracy (UA) simultaneously during training. The performance of the proposed training framework is compared from different aspects with other UQ techniques for different benchmarks. The obtained results demonstrate the effectiveness of the proposed framework for developing the NN models capable of generating reliable uncertainty estimates.**

***Index Terms*— Classification, deep neural network (NN), uncertainty accuracy (UA), uncertainty quantification (UQ).**

I. Introduction

THE advent of artificial intelligence (AI)-based systems has revolutionized the world in many aspects. Fields, such as healthcare, transportation, entertainment, cybersecurity, and education, have greatly benefited from recent advances in the field of AI [1], [2], [3], [4], [5]. The performance of neural networks (NNs) is often evaluated using error-based metrics. The frequently reported metrics in the literature for classification are accuracy, sensitivity, specificity, and cross entropy [6]. Increasing the accuracy of NN predictions is considered one of the great challenges in the AI community [7], [8], [9], [10], [11]. Traditionally, NNs are forced to make decisions even if they do not have sufficient information. On the other hand, as uncertainty is an inseparable part of all real-world applications, the reliability of predictions generated by NNs is always questionable. In order to make NN predictions more reliable and trustworthy, novel metrics are required to evaluate the NN performance, especially for high-risk (edge) samples. Quantifying the uncertainty level of NN predictions is of vital importance for many safety-critical applications, such as medical diagnosis and autonomous driving. If an automated AI-based medical diagnosis system is not confident about its decision, it can ask for a second opinion. Communication of uncertainties can effectively and efficiently minimize the number of fatal mistakes, reduce costs, and save many lives. In the end, it will also contribute to developing trustworthy AI systems.

Uncertainty in NN predictions mainly originates from two factors: the model and the data [12]. The epistemic uncertainty represents what the system does not know due to incomplete data. It determines how much the user can trust predictions in regions with no training data. The epistemic uncertainty is relatively higher in regions where there is little or no training data available in comparison with where there are sufficient training data available. Gathering more quality data will reduce epistemic uncertainty [2]. On the other hand, the aleatoric uncertainty represents the structural uncertainty in the data [13]. For instance, the inherent noise or mislabeled samples in the training data will cause aleatoric uncertainty. This type of uncertainty cannot be reduced by including more samples. While it is of vital importance to determine how accurate the NN predictions are, it is highly desirable to know how trustworthy those predictions are.

Uncertainty quantification (UQ) methods are tools to check and examine the trustworthiness of predictions [12]. Using them, an NN can communicate its lack of confidence in its predictions (*knowing when it does not know*). As a principled way for UQ, Bayesian neural networks (BNNs) bridge deep learning and Bayesian probability theory and have been extensively used to generate reliability scores for NNs [14], [15], [16], [17], [18], [19], [20]. Bayesian methods are known to be computationally demanding and not perfectly suitable for large DNNs. To address this issue, the Monte Carlo dropout (MC-dropout) algorithm [17] was proposed by adding dropout [21] for scoring. Instead of a single value prediction, a distribution of predictions is obtained that can be interpreted as a Bayesian

approximation to measure uncertainties. The main advantage of the MC-dropout is that it does not require any change applied to common practice for NN development. This feature and its ease of use are the main reasons for its popularity in the last few years for UQ studies [22], [23], [24], [25].

McClure and Kriegeskorte [26] claim that using Bernoulli or Gaussian dropout improves the classification accuracy and propose a new model based on this conclusion. The batch normalization is used in [27] as the Bayesian model. Markov Chain Monte Carlo (MCMC) [28] and stochastic gradient MCMC (SG-MCMC) [29], [30] are also popular UQ methods. In variational inference (VI)-based methods, the Bayesian inference is considered an optimization problem. This family of UQ methods aims at approximating the posterior distribution with the minimum computational burden [31], [32], [33], [34]. The Bayes-by-Backprop (BbB) algorithm is another effective UQ method that has been widely used in the literature. It applies a probability distribution for NN weights [16]. In [35], a simple method that is computationally cost-effective and requires few hyperparameter tuning that is proposed as an alternative to BNNs. In [36], an effective and simple method for detecting out of distributions and adversarial samples is proposed. A comprehensive review of UQ methods for NNs could be found in [12].

The current UQ methods for NNs generate probabilistic estimates covering both epistemic and aleatoric uncertainties at the final step. The NN development process is purely completed based on error-based performance metrics, such as accuracy and cross entropy. Confusion matrix, accuracy, specificity, and sensitivity metrics are widely used in the literature to quantitatively check the performance of a model. Unfortunately, there are not easy to understand and use evaluation metrics for uncertainty estimates. In some cases, the UQ algorithms are compared based on the predicted epistemic and aleatoric uncertainties for a specific dataset. The algorithm generating higher uncertainties is often considered to be more conservative and most of the time better. On the other hand, the output of a UQ method is often a probability distribution. This is then used to generate point estimates, such as mean, median, and mode. So, the evaluation is finally done based on point estimates. Some useful information is overlooked or lost within this process.

In short, the UQ methods estimate the uncertainties of the predictions, but the training procedure only aims to increase the accuracy, and calibrating the uncertainty is not considered in training. The uncertainty estimation performance cannot be compared precisely, and the final evaluation is based on point estimate.

A perfect uncertainty-aware algorithm is supposed to generate low uncertainty for correct predictions and high uncertainty for incorrect predictions. This is the key characteristic of a reliable UQ method. This can be well captured using a new performance metric called uncertainty accuracy (UA).

In this article, we propose a novel uncertainty-aware training framework for NNs. The proposed framework is built around a novel multiobjective loss function aiming at increasing the accuracy and UA simultaneously. To the best of our knowledge, this is the first work trying to improve the quality of uncertainty estimates by including their quality metrics in the process of NN training. The usefulness of the proposed training framework is comprehensively examined for four case studies (two synthetic and two real datasets) and compared with the common techniques used in the literature for uncertainty estimate generation. These methods include backpropagation (BP), MC-dropout, Bayesian ensembling (BE), and BbB.

The key contributions of this article are as follows.

1) Introducing novel metrics for quantitative evaluation of uncertainty estimates.
2) Designing a novel multiobjective loss function covering both the accuracy and the UA simultaneously.
3) Developing NNs using the proposed uncertainty-aware loss function.
4) Confusion matrix, confusion uncertainty matrix, and uncertainty density plots for correctly classified and incorrectly classified samples are reported and compared in all cases.

The rest of this article is organized as follows. Section II illustrates the predictive uncertainty evaluation and introduces some metrics for evaluating the UQ methods. The proposed algorithm is described in Section III in detail. Simulation results and experiments for four benchmark case studies are reported and discussed in Section IV. Section V concludes this article with some guidelines for future work.

II. Predictive Uncertainty Evaluation

Predictions generated by an NN are either *correct* or *incorrect* in a typical binary classification problem. The entropy of these predictions is interpreted as the predictive uncertainty estimate ($\mathbb{H}_i$) for each sample. This value is compared with a chosen threshold value (τ), and the predicted samples are categorized into two groups: ones that the model is certain about their predictions ($\mathbb{H}_i < \tau$), and ones that the model is uncertain ($\mathbb{H}_i > \tau$). So, the test samples are now in four different groups:

1) certain and correct (CC);
2) incorrect and uncertain (IU);
3) correct and uncertain (CU);
4) incorrect and certain (IC).

In an ideal scenario, we would like the model to predict all labels as CC. This means that the model has predicted all labels correctly, and it is certain about its predictions. The second preference is IU, which means that the model flags incorrect predictions as being uncertain. The CU refers to correct predictions that the model has not been confident about them. This is the third preference (correctly classified by chance). The IC is the worst scenario meaning that the model confidently makes incorrect predictions.

Consider a classification problem where an uncertainty-aware network trained on the training set $D_{\text{train}} = (X_{\text{train}}, Y_{\text{train}})$. Receiving the test dataset $D_{\text{test}} = (X_{\text{test}}, Y_{\text{test}})$, where $X_{\text{test}} = \{x_1, x_2, \ldots, x_n\}$ and $Y_{\text{test}} = \{y_1, y_2, \ldots, y_n\}$ represent the inputs and test labels, respectively, and the total number of test samples is n. The NN output in an uncertainty-aware classification setup is a

TABLE I

UNCERTAINTY CONFUSION MATRIX

		Correctness	
		Correct	Incorrect
Confidence	Certain	CC	IC
	Uncertain	CU	IU

probability distribution. $p_i(\hat{y}_i|x_i, w_i)$ can be considered as the NN output for the ith input. The entropy of predicted probabilities ($\mathbb{H}_i$) is defined as the predictive uncertainty estimate

$$\mathbb{H}_i(\hat{y}_i|x_i, \Theta) = -\sum_c p(\hat{y}_i = c|x_i, \Theta)\log(\hat{y}_i = c|x_i, \Theta) \quad (1)$$

where $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ indicates the NN's predictions, and c indicates all classes that y can take. Θ indicates the model parameters adjusted based on the training set D_{train}, and $\mathbb{H}_i(\cdot)$ is the predictive uncertainty estimate for y_i.

The uncertainty estimate for y_i ($\mathbb{H}_i(\hat{y}_i|X_i, \Theta)$) is compared with a threshold value $\tau \in [0, 1]$. Each prediction $\hat{y}_i$ can be then put into two groups:

1) certain when $\mathbb{H}_i(\hat{y}_i|x_i, \Theta) < \tau$;
2) uncertain when $\mathbb{H}_i(\hat{y}_i|x_i, \Theta) \geq \tau$.

Based on these, the quantitative values for CC, CU, IC, and IU are calculated as follows:

$$\text{CC} = \sum_i \mathbb{1}(\hat{y}_i = y_i \;\&\; \mathbb{H}_i(\hat{y}_i|x_i, \Theta) < \tau) \quad (2)$$

$$\text{CU} = \sum_i \mathbb{1}(\hat{y}_i = y_i \;\&\; \mathbb{H}_i(\hat{y}_i|x_i, \Theta) \geq \tau) \quad (3)$$

$$\text{IC} = \sum_i \mathbb{1}(\hat{y}_i \neq y_i \;\&\; \mathbb{H}_i(\hat{y}_i|x_i, \Theta) < \tau) \quad (4)$$

$$\text{IU} = \sum_i \mathbb{1}(\hat{y}_i \neq y_i \;\&\; \mathbb{H}_i(\hat{y}_i|x_i, \Theta) \geq \tau) \quad (5)$$

where $i = 1, \ldots, N$. It is important to note that CC, CU, IC, and IU are all smaller than N.

A. Uncertainty Confusion Matrix

Similar to the concept of confusion matrix, CC, IC, CU, and IU can be used to build the uncertainty confusion matrix. This is shown in Table I. It is desired to have all samples in CC and IU cells [37], [38], [39].

B. Correct-Certain Ratio

The correct-certain ratio (R_{CC}) is defined as the ratio of CC predictions to all certain predictions

$$R_{\text{CC}} = P(\text{Correct}|\text{Certain}) = \frac{P(\text{Correct, Certain})}{P(\text{Certain})} = \frac{\text{CC}}{\text{CC} + \text{IC}}.$$

The best value of R_{CC} is one.

C. Incorrect-Uncertain Ratio

The incorrect-uncertain ratio (R_{IU}) is defined as the ratio of IU predictions to all uncertain predictions

$$R_{\text{IU}} = P(\text{Incorrect}|\text{Uncertain}) = \frac{P(\text{Incorrect, Uncertain})}{P(\text{Uncertain})} = \frac{\text{IU}}{\text{IU} + \text{CU}}.$$

R_{IU} should be ideally one.

D. Uncertainty Accuracy

According to Table I, the accuracy of predictive uncertainty estimates can be defined as follows [37], [38], [39]:

$$\text{UA} = \frac{\text{CC} + \text{IU}}{\text{CC} + \text{IU} + \text{CU} + \text{IC}}. \quad (6)$$

It is worth mentioning that the UA can be interpreted as the ratio of favorable results (CC and IU) to all possible results. For all the above metrics, higher values indicate better performance.

III. PROPOSED ALGORITHM

In this section, the details of the proposed algorithm are provided. We first define a multiobjective loss function covering both the accuracy and the UA simultaneously. The main challenge is finding a proper term for UA in the loss function. The uncertainty density plots of correctly and incorrectly classified samples indicate that the UA increases, as the overlap between these density plots decreases (strong negative correlation). So, we could potentially consider the distance of these density plots as a UA representative.

The Kullback–Leibler (KL) divergence is applied here to measure the distance between uncertainty density estimates for correct and incorrect predictions

$$\text{KL}(p||q) = \Sigma_{i=1}^{N} p(x_i)\log\left(\frac{p(x_i)}{q(x_i)}\right) \quad (7)$$

where $p(x_i)$ and $q(x_i)$ are related to correctly and incorrectly classified samples. So, the proposed loss function is defined as follows:

$$\text{Loss}_{\text{UA}} = \mathcal{L} + \beta\ \text{KL}(\Phi_C, \Phi_I) \quad (8)$$

where β is a constant coefficient. Also, $\mathcal{L}$ is a conventional loss function

$$\mathcal{L} = \alpha\ C(y_i, \hat{y}_i). \quad (9)$$

This could be the cross entropy, or it is defined based on the prediction error (difference between the prediction $\hat{y}_i$ and the target y_i). Φ_C and Φ_I represent correctly and incorrectly classified samples, respectively.

$\text{KL}(\Phi_C, \Phi_I)$ is the KL divergence of uncertainty density estimates of correct and incorrect predictions. It is the UA representative in the proposed loss function. The details of the proposed algorithm are described in Algorithm 1.

Alpha and beta can be considered to regularize the emphasis on the loss function terms. For instance, if it is desirable that the first term in the loss function is dominant (plays the main role), a very higher value can be chosen for alpha in

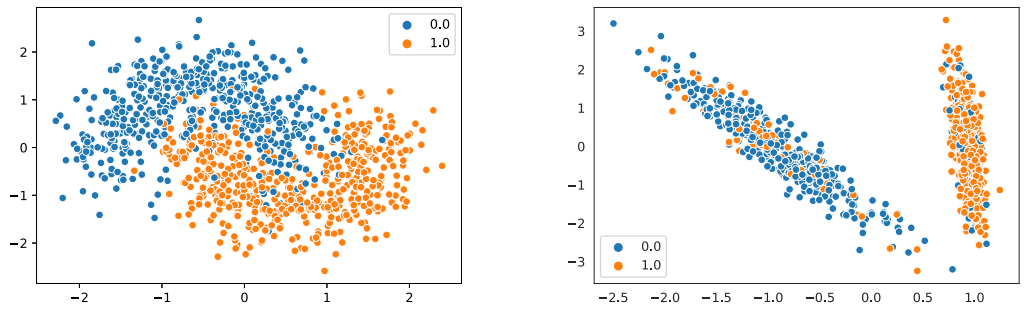


Fig. 1. 2-D synthetic dataset used in this study. (a) Two-moon dataset. (b) Random two-class classification dataset.

Algorithm 1 Uncertainty-Aware Training Algorithm (Framework) for NNs

Choose the dataset $D(X, Y)$
Split the data into D^{train} and D^{test}
$\varphi_C^{train} \leftarrow \{\}$
$\varphi_I^{train} \leftarrow \{\}$
Randomly initialize the network parameters (Θ_0).
while Training criteria not met **do**
  Generate predictions for D^{train}
  if $y_i^{train} == \hat{y}_i^{train}$ **then**
    $\varphi_C^{train}.append(D_i^{train})$
  end if
  if $y_i^{train} \neq \hat{y}_i^{Dtrain}$ **then**
    $\varphi_I^{train}.append(y_i^{train})$
  end if
  Calculate $\mathbb{H}_C^{train}$ for samples in φ_C^{train}
  Calculate $\mathbb{H}_I^{train}$ for samples in φ_I^{train}
  $Loss_{UA} \leftarrow \mathcal{L} + \beta\ KL(\mathbb{H}_C^{train}, \mathbb{H}_I^{train})$
  Update Θ through minimization of this loss
end while
Generate prediction for D^{test}
Evaluate predictive uncertainty estimates for D^{test}

comparison with beta. A higher value for alpha means that the main emphasis of the loss function is on optimizing the accuracy. In case the second term of loss function should play the main role, a higher value for beta is chosen in comparison with alpha. A higher value for beta means that the main emphasis is on optimizing the UA. However, in most cases, including our work, similar values are chosen for alpha and beta (equal importance).

IV. Experiments

In this section, the proposed algorithm is applied to different synthetic and real-world datasets. Considered datasets are as follows:

1) two-moon dataset [see Fig. 1(a)];
2) random two-class classification dataset [see Fig. 1(b)];
3) COVID-19 dataset [40];
4) breast cancer dataset [41].

The first two datasets are synthetic ones widely used in the relevant literature. The other two are real medical datasets used for analyzing the performance of deep NNs.

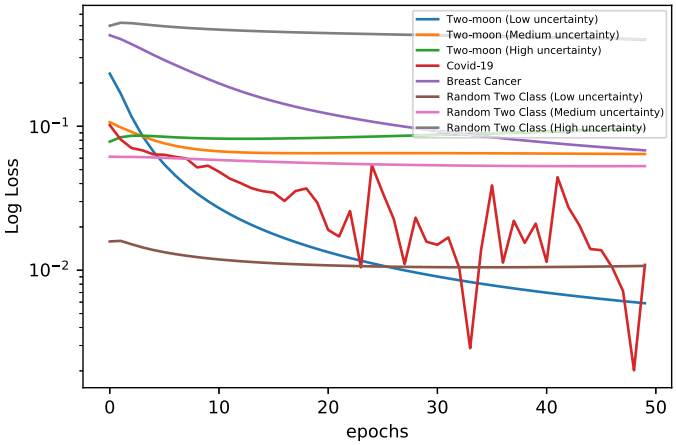


Fig. 2. Convergence plots for the validation set were obtained from NNs developed using the proposed uncertainty-aware training algorithm. For synthetic case studies, the more the uncertainty, the greater the validation loss.

The proposed algorithm is also compared with several benchmark algorithms, including BP, MC-dropout [22], [23], [24], [25], BE [42], and BbB [16]. The accuracy, UA, the confusion matrix, and the uncertainty confusion matrix are reported and compared in all cases to evaluate the performance of different algorithms. In addition, the uncertainty density plots for correctly and incorrectly classified samples are reported for all datasets and methods. The distance between these two uncertainty density plots has a positive correlation with the UA. The average predicted uncertainty for correctly and incorrectly classified samples is also reported for all cases. It is worth mentioning that choosing a suitable value for the threshold may improve the results. The comparative results for all datasets and models are shown in Table II. The convergence plots of the loss function for the validation are shown in Fig. 2. For two synthetic case studies, the higher the uncertainty level, the larger the validation loss. This indicates that the proposed training methods properly capture different sources of uncertainty in the data and assign more predictive uncertainty to less certain predictions. It is worth mentioning that in all scenarios, the τ value is selected at 0.6, and a fully connected NN is considered.

A. COVID-19 Dataset

The purpose is to automatically diagnose COVID-19 cases using chest computerized tomography (CT) images. The dataset contains only 275 positive cases out of 470 cases. The

TABLE II

COMPARISON OF FIVE UNCERTAINTY-AWARE ALGORITHMS FOR COVID-19, BREAST CANCER, TWO-MOON, AND RANDOM TWO-CLASS CLASSIFICATION DATASETS

Dataset	Metric	Proposed	Back-Propagation	MC	B-Ensembling	Bayes-by-Backprop
Covid-19	Accuracy (%)	79	68	**83**	81.9	59
	Uncertainty Accuracy (%)	**72**	**72**	64	57	40
Breast Cancer	Accuracy (%)	**94**	92	85	93	87
	Uncertainty Accuracy (%)	**93**	88	85	88	82
Two-moon (Low uncertainty)	Accuracy (%)	**100**	**100**	90	90	99
	Uncertainty Accuracy (%)	**99.6**	99	87	90	99
Two-moon (Medium uncertainty)	Accuracy (%)	**93**	90	86	85	89
	Uncertainty Accuracy (%)	**92**	72	84	84	87
Two-moon (High uncertainty)	Accuracy (%)	82	80	**83**	**83**	77
	Uncertainty Accuracy (%)	**80**	72	70	78	78
Random Two Class (Low uncertainty)	Accuracy (%)	95	93	**96**	92	**96**
	Uncertainty Accuracy (%)	95	94	93	92	**96**
Random Two Class (Medium uncertainty)	Accuracy (%)	**84**	77	82	82	**84**
	Uncertainty accuracy (%)	83	73	79	83	**84**
Random Two Class (High uncertainty)	Accuracy (%)	**78**	67	77	73	77
	Uncertainty accuracy (%)	**77**	67	29	72	**77**

VGG16 [43] is applied for extracting features from chest CT images. The extracted features are used as input for developing the classifier (an NN model in this case). Due to the limited number of training samples, the data augmentation is applied to increase the number and diversity of training samples [44]. The extracted features are then fed to five different UQ algorithms. The results of the proposed algorithm, BP, MC-dropout, BE, and BbB are compared in terms of accuracy and UA in Table II. The proposed method achieves an accuracy of 79%, which is much higher than those of the BP and BbB, but it is slightly lower than those of the MC-dropout (83%) and BE (81.9%). However, the latter two perform poorly based on UA metrics (64% and 57%, respectively). In contrast, the UA for the proposed method is 72%, which is the highest UA for this dataset. It can be concluded that for this dataset, the proposed method gains a better performance considering both accuracy and UA at the same time.

In an ideal case, the correct predictions must have high confidence (low uncertainty), and the incorrect predictions must have low confidence (high uncertainty). Fig. 3 displays the average uncertainty for correct and incorrect predictions of the test set. The uncertainty density plots of correctly classified and incorrectly classified test samples for all methods are also reported in Fig. 4. It is worth mentioning that there is a positive correlation between the distance of these density plots and UA. The proposed UQ algorithm generates more reliable and accurate uncertainty estimates on average in comparison with other UQ methods. Two density plots for MC-dropout, BE, and BbB algorithms have a high overlap, which means that the uncertainty estimates generated by these models are not indicative of the prediction correctness. For the proposed UQ method, the uncertainty estimates are also much higher for the incorrectly predicted samples. So, in general, the proposed method performs better in uncertainty estimation on average. Also, it is noted that the uncertainty for correct predictions is much lower for the proposed, BP, and MC-dropout algorithms than for other methods. The BE and BbB algorithms do not have a good performance in this regard, as they predict high uncertainty on average for correctly classified samples.

B. Breast Cancer Dataset

According to the results in Table II, the proposed algorithm achieves the highest accuracy (94%) and UA (93%) for the breast cancer dataset. The BP and BE also reach the competitive accuracy at 92% and 93%, respectively, but their UA values are much lower (88%).

According to Figs. 3 and 5, all algorithms predict relatively lower uncertainty for correctly classified samples in comparison with the incorrectly classified samples. The best performance in this regard is reported for the proposed algorithm. This is followed by the BP and BE methods. The BbB obtains the poorest results in terms of UA, indicating that it cannot well separate two classes in terms of uncertainty estimates.

C. Two-Moon Dataset

The first synthetic dataset is the well-known two-moon dataset [see Fig. 1(a)]. It consists of 1000 samples of two classes. To better examine the performance of the proposed method, the uncertainty level of this dataset is considered in three levels: low, medium, and high. The training and testing sets include 70% and 30% of samples, respectively. The performance results for the proposed method and benchmark methods are reported in Table II. For the case of low uncertainty, the proposed method achieves the highest accuracy (100%) and the UA (99.6%). The BP and BbB obtain similar results as well. Fig. 3 also shows that the proposed method and the BP are certain in their predictions and predict all the labels correctly. The lowest performance is related to MC-dropout whose accuracy and UA are 90% and 87%, respectively.

For medium uncertainty, the accuracy and the UA of the proposed algorithm are the highest (93% and 92%, respectively). The lowest UA is at 72%, which is related to BP. The average uncertainties reported in Fig. 3 show that the BbB is in the second rank.

For the case of high uncertainty, the best-reported accuracy (83%) belongs to MC-dropout and BE. However, these methods perform poorly based on UA metrics (70% and 78%,

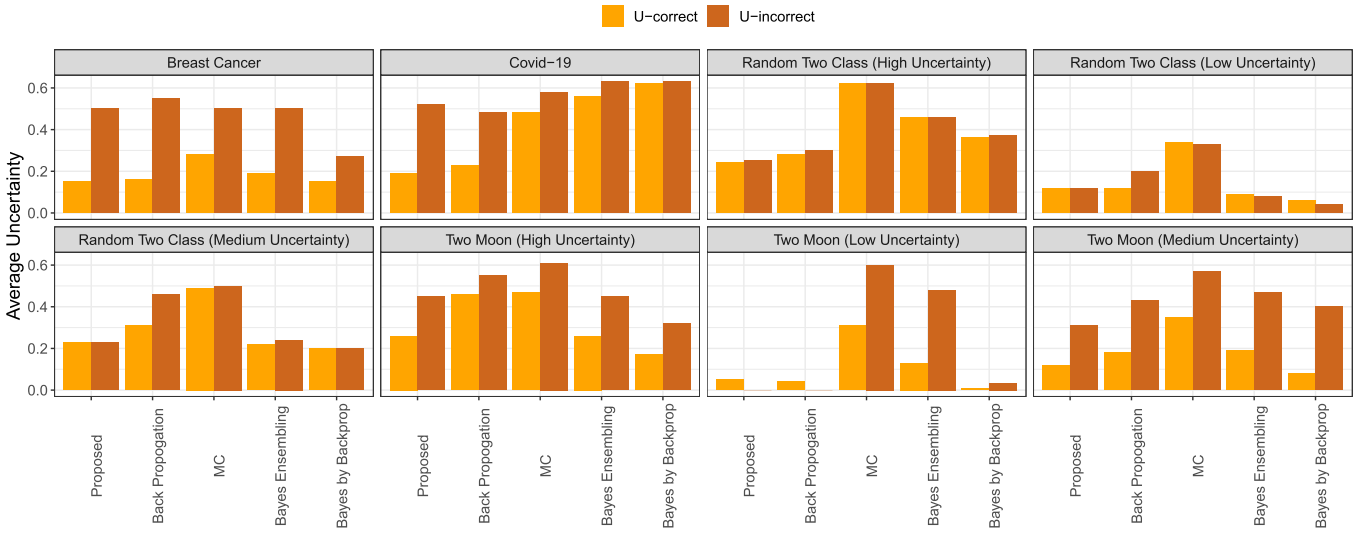


Fig. 3. Comparing the average uncertainty for correctly classified and misclassified samples for all datasets and methods.

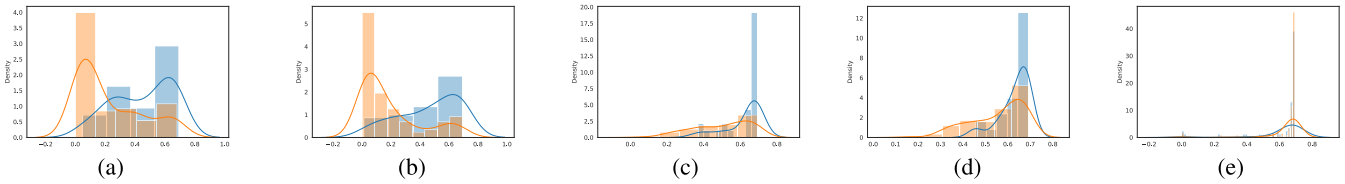


Fig. 4. Entropy density plots of correct and incorrect predictions for the COVID-19 dataset. (a) Proposed. (b) BP. (c) MC-dropout. (d) BE. (e) BbB.

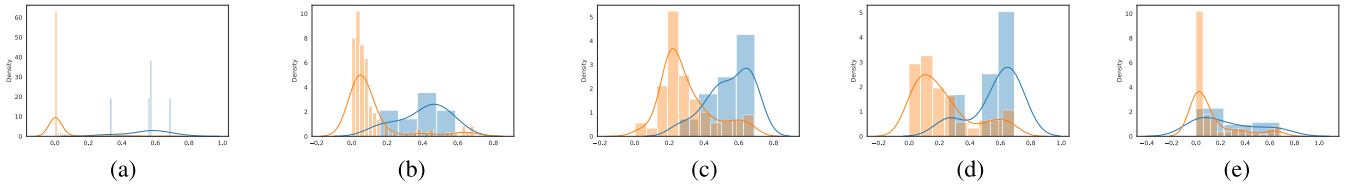


Fig. 5. Entropy density plots of correct and incorrect predictions for the breast cancer dataset. (a) Proposed. (b) BP. (c) MC-dropout. (d) BE. (e) Bayes-by-Backprop.

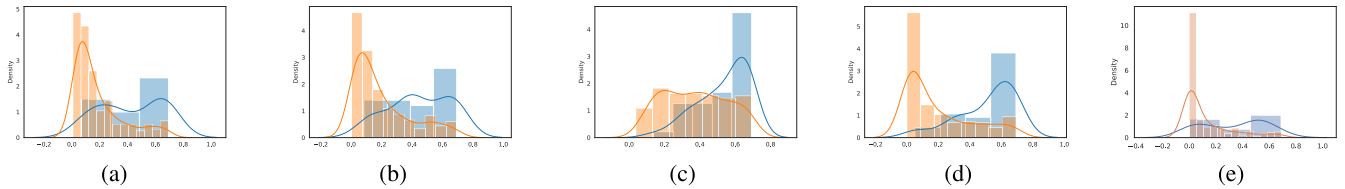


Fig. 6. Entropy density plots of correct and incorrect predictions for the two-moon dataset. (a) Proposed. (b) BP. (c) MC-dropout. (d) BE. (e) BbB.

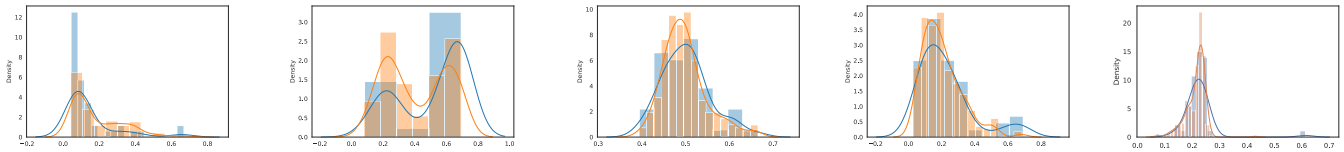
respectively). The proposed method achieves a very reasonable performance in terms of accuracy (82%) and UA (80%). Its UA is the best among all investigated methods. Considering Fig. 3, the proposed method generates smaller uncertainty estimates for correct predictions than incorrect predictions.

The uncertainty density plots of correctly classified and incorrectly classified test samples are shown in Fig. 6 for all methods. The MC-dropout is the weakest in terms of separation between two densities.

D. Random Two-Class Dataset

The random two-class dataset includes 1000 samples, as shown in Fig. 1(b). Again, three levels of uncertainty (low, medium, and high) are considered for this case study. For the case of low uncertainty, the best performance is reported for BbB with the accuracy and UA of 96%. The proposed method performs competitively well and achieves the accuracy and UA of 95%. Fig. 3 also shows that the proposed algorithm on average predicts lower uncertainty for correct predictions compared with incorrect predictions. It is also noted that the MC-dropout, BE, and BbB predict higher uncertainties for correct predictions in comparison with incorrect predictions. This simply means that they can confidently make wrong predictions.

For the case of medium uncertainty, the best accuracy results are reported for the proposed and BbB methods. The BbB



(a) (b) (c) (d) (e)

Fig. 7. Entropy density plots of correct and incorrect predictions for the random two-class dataset. (a) Proposed. (b) BP. (c) MC-dropout. (d) BE. (e) BbB.

achieves a UA of 84%, while it is 83% for the proposed method (second rank). The BP algorithm obtains the poorest results.

The proposed method achieves the best results for the case of high uncertainty. The accuracy and UA are 78% and 77%, respectively. The MC-dropout algorithm also reaches good accuracy of 77%, but its UA value is very low (29%). The bar plots in Fig. 3 and density plots in Fig. 7 carry the same information.

V. Conclusion

In this article, a novel uncertainty-aware training framework is proposed for developing NNs. The uncertainty confusion matrix and uncertainty-related metrics, such as UA, are defined for evaluating the quality of predictive uncertainty estimates. Based on these metrics, a novel multiobjective loss function is defined, covering both the prediction accuracy and the UA simultaneously. The NN parameters are then adjusted by minimizing the multiobjective loss function using the gradient descent method. The comprehensive and comparative studies demonstrate the competency of the proposed framework in generating quality uncertainty estimates.

The framework proposed in this article opens up many research opportunities. The multiobjective uncertainty-aware loss function and its variations could be used for redesigning and rebuilding traditional UQ techniques, such as density mixture networks. Also, it could be utilized for optimal neural architecture search.

References

[1] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3D vehicle detection," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3266–3273.

[2] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, "Sampling-free epistemic uncertainty estimation using approximated variance propagation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2931–2940.

[3] R. Alizadehsani et al., "Hybrid genetic-discretized algorithm to handle data uncertainty in diagnosing stenosis of coronary arteries," *Expert Syst.*, vol. 39, no. 7, 2020, Art. no. e12573.

[4] R. Alizadehsani et al., "Model uncertainty quantification for diagnosis of each main coronary artery stenosis," *Soft Comput.*, vol. 24, no. 13, pp. 10149–10160, 2019.

[5] P. Tabarisaadi, A. Khosravi, and S. Nahavandi, "A deep Bayesian ensembling framework for COVID-19 detection using chest CT images," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 1584–1589.

[6] J. M. Twomey and A. E. Smith, "Performance measures, consistency, and power for artificial neural network models," *Math. Comput. Model.*, vol. 21, nos. 1–2, pp. 243–258, Jan. 1995.

[7] G. M. Foody and M. K. Arora, "An evaluation of some factors affecting the accuracy of classification by an artificial neural network," *Int. J. Remote Sens.*, vol. 18, no. 4, pp. 799–810, Mar. 1997.

[8] T. Kavzoglu, "Increasing the accuracy of neural network classification using refined training data," *Environ. Model. Softw.*, vol. 24, no. 7, pp. 850–858, Jul. 2009.

[9] V. A. Maksimenko et al., "Artificial neural network classification of motor-related EEG: An increase in classification accuracy by reducing signal complexity," *Complexity*, vol. 2018, pp. 1–10, Aug. 2018.

[10] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogram. Remote Sens.*, vol. 145, pp. 120–147, Nov. 2018.

[11] R. M. Ramos, C. G. Ralha, T. M. Kurc, J. H. Saltz, and G. Teodoro, "Increasing accuracy of medical CNN applying optimization algorithms: An image classification case," in *Proc. 8th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2019, pp. 233–238.

[12] M. Abdar et al., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," 2020, *arXiv:2011.06225*.

[13] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," 2019, *arXiv:1910.09457*.

[14] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015.

[15] A. Graves, "Practical variational inference for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2348–2356.

[16] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015, *arXiv:1505.05424*.

[17] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[18] S. Farquhar, M. A. Osborne, and Y. Gal, "Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1352–1362.

[19] J. Maroñas, R. Paredes, and D. Ramos, "Calibration of deep probabilistic models with decoupled Bayesian neural networks," *Neurocomputing*, vol. 407, pp. 194–205, Sep. 2020.

[20] P. Izmailov, W. J. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, and A. G. Wilson, "Subspace inference for Bayesian deep learning," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 1169–1179.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

[22] K. Brach, B. Sick, and O. Dürr, "Single shot MC dropout approximation," 2020, *arXiv:2007.03293*.

[23] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, Apr. 2019.

[24] H. Liu et al., "Universal adversarial perturbation via prior driven uncertainty approximation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2941–2949.

[25] T. Nair, D. Precup, D. L. Arnold, and T. Arbel, "Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation," *Med. Image Anal.*, vol. 59, Jan. 2020, Art. no. 101557.

[26] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *J. Mach. Learn. Res.*, vol. 9, no. 3, 2008.

[27] M. Teye, H. Azizpour, and K. Smith, "Bayesian uncertainty estimation for batch normalized deep networks," 2018, *arXiv:1802.06455*.

[28] M. A. Kupinski, J. W. Hoppin, E. Clarkson, and H. H. Barrett, "Ideal-observer computation in medical imaging with use of Markov-chain Monte Carlo techniques," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 20, no. 3, pp. 430–438, 2003.

[29] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven, "Bayesian sampling using stochastic gradient thermostats," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3203–3211.

[30] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian Monte Carlo," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1683–1691.

[31] J. Swiatkowski et al., "The k-tied normal distribution: A compact parameterization of Gaussian mean field posteriors in Bayesian neural networks," 2020, *arXiv:2002.02655*.
[32] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.
[33] A. Mnih and D. Rezende, "Variational inference for Monte Carlo objectives," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2188–2196.
[34] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008–2026, 2018.
[35] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2016, *arXiv:1612.01474*.
[36] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," 2018, *arXiv:1807.03888*.
[37] P. Tabarisaadi, A. Khosravi, and S. Nahavandi, "Uncertainty-aware skin cancer detection: The element of doubt," *Comput. Biol. Med.*, vol. 144, May 2022, Art. no. 105357.
[38] A. S. Jokandan et al., "An uncertainty-aware transfer learning-based framework for COVID-19 diagnosis," 2020, *arXiv:2007.14846*.
[39] H. Asgharnezhad et al., "Objective evaluation of deep uncertainty predictions for COVID-19 detection," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, Dec. 2022.
[40] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "COVID-CT-dataset: A CT scan dataset about COVID-19," 2020, *arXiv:2003.13865*.
[41] W. H. Wolberg, W. N. Street, and O. L. Mangasarian. (1992). *Breast Cancer Wisconsin (Diagnostic) Data Set*. UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml/
[42] T. Pearce, F. Leibfried, and A. Brintrup, "Uncertainty in neural networks: Approximately Bayesian ensembling," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 234–244.
[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[44] F. Chollet, "Fundamentals of maching learning," in *Deep Learning With Python*. Shelter Island, NY, USA: Manning Publications Co, 2017.