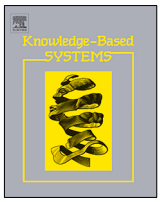
Knowledge-Based Systems 243 (2022) 108418

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



# Explainable machine learning in image classification models: An uncertainty quantification perspective

Xiaoge Zhang [a], Felix T.S. Chan [b,*], Sankaran Mahadevan [c,*]

[a] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[b] Department of Decision Sciences, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macao
[c] Department of Civil and Environmental Engineering, School of Engineering, Vanderbilt University, Nashville, TN 37235, USA

ARTICLE INFO

*Article history:*
Received 20 July 2021
Received in revised form 5 February 2022
Accepted 8 February 2022
Available online 17 February 2022

*Keywords:*
Deep learning
Model explainability
Uncertainty quantification
Bayesian neural network
Prediction difference analysis

ABSTRACT

The poor explainability of deep learning models has hindered their adoption in safety and quality-critical applications. This paper focuses on image classification models and aims to enhance the explainability of deep learning models through the development of an uncertainty quantification-based framework. The proposed methodology consists of three major steps. In the first step, we adopt dropout-based Bayesian neural network to characterize the structure and parameter uncertainty inherent in deep learning models, propagate and represent such uncertainties to the model prediction as a distribution. Next, we employ entropy as a quantitative indicator to measure the uncertainty in model prediction, and develop an Empirical Cumulative Distribution Function (ECDF)-based approach to determine an appropriate threshold value for the purpose of deciding when to accept or reject the model prediction. Secondly, in the cases with high model prediction uncertainty, we combine the prediction difference analysis (PDA) approach with dropout-based Bayesian neural network to quantify the uncertainty in pixel-wise feature importance, and identify the locations in the input image that highly correlate with the model prediction uncertainty. In the third step, we develop a robustness-based design optimization formulation to enhance the relevance between input features and model prediction, and leverage a differential evolution approach to optimize the pixels in the input image with high uncertainty in feature importance. Experimental studies in MNIST and CIFAR-10 image classifications are included to demonstrate the effectiveness of the proposed approach in increasing the explainability of deep learning models.

© 2022 Elsevier B.V. All rights reserved.

## 1. Introduction

Artificial intelligence (AI) applications in recent years increasingly use deep learning models. Different from conventional machine learning algorithms (e.g., support vector machine, decision tree, logistic regression), deep learning models deal directly with natural data in their raw form (e.g., image, voice, video), and learn representations in high-dimensional space from the raw data through multiple neuron layers and activation functions in an automatic fashion [1–5]. Such features have made deep learning more appealing than other conventional machine learning algorithms since it does not require carefully handcrafted (often manual) feature extraction or considerable domain expertise to extract informative patterns that can be detected or classified by the learning algorithm.

Although deep learning is being widely adopted in many domains, there have been growing concerns regarding its black-box nature, and end-users are hesitant to adopt deep learning models in high-stakes decision-making environments [6–8], such as healthcare, self-driving cars, and criminal justice. Since representation learning is too complex in the high-dimensional space and it is difficult to explain the learning process in a way that humans can comprehend, deep learning models are often treated as black-box functions in this regard. The growing prevalence of deep learning models in safety-critical domains has resulted in increasing demands for transparent, reproducible, and certifiable artificial intelligence (AI). Besides, when deep learning models make wrong predictions, there are no mature approaches to enable backward reasoning and diagnosis of the internal structure of deep learning models for the sake of correcting the model structure and parameters in order to avoid making similar mistakes in the future. The integration of explainability into deep learning models will help form a feedback loop that integrates model diagnosis and model refinement in the model development process. Model explainability is an essential component for the

* Corresponding authors.
*E-mail addresses:* xiaoge.zhang@polyu.edu.hk (X. Zhang), felix1202.chan@connect.polyu.hk (F.T.S. Chan), sankaran.mahadevan@vanderbilt.edu (S. Mahadevan).

https://doi.org/10.1016/j.knosys.2022.108418
0950-7051/© 2022 Elsevier B.V. All rights reserved.

development of such a closed-loop machine learning system. In view of all these factors, there is an increasing need for explainable deep learning, and for the learning mechanism to be presented in such a way that is easy to understand, where domain experts can leverage their expertise to verify and validate the predictions as well as the representation learning in deep learning models to ensure compliance with regulation standards.

Over the past few years, several types of solutions have been proposed to address the lack of transparency in deep learning models; these are often referred to as explainable artificial intelligence (XAI) solutions [9–12]. For example, Ribeiro et al. [13] developed Local explainable Model-Agnostic Explanation (LIME), where the classifier or regressor was approximated locally with an explainable model to interpret the model's prediction. Zintgraf et al. [14] used the prediction difference analysis approach to explain the decisions made by deep learning classifiers, where pixels supporting or not supporting a certain class in the input image were identified. Lundberg and Lee [15] combined optimal credit allocation with local explanation to interpret the prediction of any machine learning model. Besides, Lundberg et al. [16] presented TreeExplainer that combined many local explanations of each prediction to provide optimal local explanations for model predictions from a game theory point of view. Kim et al. [17] proposed an Explaining and Visualizing Convolutional neural network for Text information (EVCT) framework to provide human-explainable explanations and visualizations of text classification results, where sparse principal component analysis (SPCA) was developed to maximize human-explainability. Yang et al. [18] focused on explainable recommendation systems, and developed a deep learning-based method named Hierarchical Attention Network Oriented Towards Crowd Intelligence (HANCI) to extract important words and useful reviews to provide word-level and review-level explanations to achieve more accurate recommendation. Spinner et al. [19] developed explAIner — a framework for visible, interactive, and interpretable machine learning consisting of three stages: model understanding, model diagnosis, and model refinement. Leibig et al. [20] evaluated dropout-based Bayesian uncertainty measures for deep learning in diagnosing diabetic retinopathy (DR) from fundus images and showed that uncertainty-informed decision referral could improve diagnostic performance, where binary entropy was used to compare the performance of a Bayesian neural network with alternative model uncertainty quantification measures (e.g., Gaussian Process). Kwon et al. [21] quantified the uncertainty of predicted outcomes for classification via Bayesian neural networks, decomposed the predictive uncertainty measure into aleatory and epistemic uncertainties, and demonstrated interpretability of uncertainty maps in biomedical image segmentation. Jungo et al. [22] computed the uncertainty of a fully-connected convolutional neural network using Monte Carlo dropout, and employed entropy as a quantitative measure to identify segmentation results necessitating expert review for a clinical brain tumor dataset. Several quantitative explanation techniques have been developed to measure the robustness and fairness of machine learning algorithms. For example, Sharma et al. [23] proposed to use CERScore and NCERScore to quantify and explain the robustness and fairness of artificial intelligence systems by drawing a relation between counterfactuals and adversarial examples. Weng et al. [24] developed a generic metric based on extreme value theory to efficiently assess and interpret the robustness of neural network classifiers. However, these metrics are not suitable to be used in attribution-based methods, which seek to determine the contribution of each input feature to the target output. Recently, Burkart and Huber [25] conducted a state-of-the-art review of XAI methodologies in the literature.

Despite the tremendous progress made over the past few years, several crucial issues remain to be addressed along with the development of XAI. First, despite the active research on explainable machine learning, there is no unified or widely used definition of explainability among researchers. The literature thus far has pursued two major directions towards analyzing and enhancing the explainability of deep learning models: the first one is to uncover the connections between input features and intermediate layers as well as model prediction, and eventually provide explainability through sensitivity analysis (e.g., salient map [14]). The other direction is to train a post-hoc model to explain the black box model, or train a model with built-in interpretability (e.g., decision trees [26]). Secondly, a large amount of effort has focused on explaining the learning mechanisms of deterministic deep learning models. However, in practice, a deterministic prediction might be insufficient for decision-making activities in safety-critical environments. From this standpoint, there is an increasing demand for probabilistic deep learning models. Thirdly, if we use probabilistic deep learning models, in the case of high model prediction uncertainty, end-users are interested in learning which factors contribute to the model prediction uncertainty the most, e.g., whether noisy input, or the model itself (e.g., model structure and parameter uncertainty). Proper explanations need to be provided to address these issues.

Motivated by these needs, we analyze the explainability of deep learning models from an uncertainty quantification perspective. In terms of scope, we consider enhancing model explainability by analyzing (i) the contribution of each pixel to model prediction uncertainty, and (ii) the reduction in model prediction uncertainty after a noisy pixel is removed from the input image. We use these analyses to develop an optimization approach to maximize the explainability of the model, i.e., maximize the uncertainty reduction. Compared to previous studies, we make the following new contributions in this paper:

1. We propose to use entropy as a quantitative indicator to characterize the uncertainty in model prediction, thus deciding when to accept or reject predictions made by deep learning models. The introduction of entropy makes the neural network less prone to mistakes when the input data comes from outside the domain that the model is trained for. An Empirical Cumulative Distribution Function (ECDF)-based approach is developed to determine the threshold value of the entropy-based indicator. We demonstrate how entropy can be used as an effective measure to equip the model with the ability to say "I do not know". Experimental studies on the "out-of-domain" cases demonstrate that entropy acts as an effective safety net to guard the application domain of the trained model.
2. In cases with high model prediction uncertainty, we combine the prediction difference analysis (PDA) approach (explained in Section 4.2) with Bayesian neural network to characterize the uncertainty in feature importance of each input pixel. Note that feature importance is measured by sensitivity that is quantified by the change in model output corresponding to a small perturbation in model input [14]. Since PDA replaces the removed input feature using samples from a multivariate Gaussian distribution, we have multiple PDA runs; this helps to quantify feature importance uncertainty by measuring the variation of feature importance over the multiple PDA runs. Further, we identify the areas of input images that contribute to the uncertainty in input–output importance. The identification of such areas helps to identify the sources in the input data that dominantly contribute to the uncertainty in model prediction.
3. After pixels with high feature importance uncertainty in the input image are located, we formulate a robustness-based design optimization problem and take a differential

evolution approach to optimize the pixels with high feature importance uncertainty in the input image to reduce model prediction uncertainty, thereby enhancing the explainability of machine learning models. Reducing the noise in the input image strengthens the importance of the input image to model prediction, thus increasing the explainability of model prediction w.r.t. the instance.

The rest of the paper is structured as follows. Section 2 reviews research developments along novelty detection, abstention studies, and uncertainty quantification techniques. Section 3 introduces the basic concept of Bayesian deep learning. Section 4 introduces the proposed XAI approach consisting of three major steps: entropy-based uncertainty measure, prediction difference analysis to quantify the importance of input pixels to model prediction and the quantification of importance uncertainty, and robustness-based design optimization. Section 5 evaluates the performance of the proposed approach on several datasets, and demonstrates its effectiveness in interpreting deep learning models. Section 6 discusses the benefits brought by the developed approach. Section 7 provides concluding remarks and discusses future research directions.

## 2. Literature review

Deep learning has shown remarkable performance in tackling problems where the problem domain (e.g., image classification, object recognition, machine translation) is prohibitively challenging to be defined in an analytical form. The difficulty in formalizing the problem domain makes it a necessity to detect the scenarios when it becomes unreliable to count on the predictions made by the deep learning model. Towards this end, multiple approaches and techniques have been developed to address this issue, such as novelty detection [27], anomaly detection [28], concept shift [29], and abstention studies [30].

Among them, novelty detection aims to identify test data that differs in some features from the data the model is trained with [31]. It is also referred to as outlier detection [32], anomaly detection [33], one-class classification (OCC) [34] in the literature. The primary goal of novelty detection is to build a novelty detector to identify any data divergence from normality. Recently, Pimentel et al. [35] provided a comprehensive review on the methods developed for novelty detection. Roughly speaking, the methods for novelty detection can be categorized into two groups: semi-supervised training and unsupervised training techniques. For example, Wu and Ye [36] used normal and a small set of abnormal examples to construct a small hypersphere in the feature space to separate the normal data from abnormal data, while maximizing the margin between the hypersphere and the abnormal data points. Chen et al. [37] developed a semi-supervised kernelized spatial depth approach, and they labeled an observation with a depth value less than a predefined threshold as an outlier. In terms of unsupervised training techniques, Zhang et al. [38] proposed a Local Distance-based Outlier Factor (LDOF) to quantify the degree of deviation of a data point from its neighborhoods. Masud et al. [39] leveraged K-means clustering to detect drift in concept-evolving data streams.

Another perspective to safeguard the usage of deep learning models is to introduce the option of abstention when deep learning models are used for making predictions. Abstention refers to reject or refuse a prediction made by deep learning models. In high-stakes environments, when there is a large amount of uncertainty, abstaining from providing a prediction is a much better option than taking the consequence of a wrong decision. Over the past few years, a variety of abstention-based machine learning approaches have been developed, and these methods generally rely on two principal ideas: they either introduce abstention-specific cost into the loss function, or learn to abstain so that the performance of the model reaches a specific target. For example, Thulasidasan et al. [40] proposed abstention-based training to combat label noise, where a customized loss function was defined to incorporate the cost of abstention. Garcia et al. [41] developed a Structured Output Learning with Abstention (SOLA) framework to allow abstaining from predicting parts of the structure for the sake of avoiding providing erroneous insights as well as increasing the reliability of model predictions. As mentioned by Kompa et al. [42], uncertainty quantification is an essential component in the abstention of individual predictions made by deep learning algorithms. The quantification of aleatory and epistemic uncertainty allows models to abstain from making predictions for cases involving a large amount of uncertainty.

Besides, a large body of studies have attempted to unveil the intricate decision-making process of deep learning models by developing quantitative metrics to measure the robustness and faithfulness of neural networks. Typically, robustness (also called stability or sensitivity) is concerned with how a well-trained machine learning model changes due to a small perturbation of the input. For example, Alvarez-Melis and Jaakkola [43] designed self-explaining models by progressively generalizing linear classifiers to complex yet architecturally explicit models, and they investigated the robustness of the classifier by evaluating the relative change of the explanation vector within the $\epsilon$-sized neighborhood of the input. Guidotti and Ruggieri [44] performed comprehensive experimental studies to quantify the stability (or sensitivity) of interpretable models with respect to feature selection, instance selection, and model selection. In addition to robustness, another set of metrics that have been actively used is faithfulness, which is also referred to as infidelity or truthfulness [45]. This set of metric aims to check if the provided feature importance from an interpretability technique is faithful by changing the input and monitoring the output of the predictive model accordingly. Yeh et al. [46] proposed two objective explanation metrics, namely infidelity and sensitivity, to measure the feature importance by quantifying how the predictor function itself changed in response to random perturbations. Du et al. [47] developed a REcurrent ATtribution Method (REAT) to provide interpretation for predictions made by a RNN model, where they examined the faithfulness of the attribution method by observing the prediction change of the RNN after the sentence with the highest contribution was deleted. Mollas et al. [48] studied the faithfulness of feature importance-based interpretation using an Altruist Truthfulness Investigator (ATI) approach by examining the relative change of a model's prediction due to the variation of each feature's value.

In recent years, rapid progress has been made along the development of uncertainty quantification techniques in deep learning. For example, Blundell et al. [49] proposed a backpropagation-compatible algorithm called "Bayes by Backprop" to learn the probability distributions of weights in a neural network. Ritter et al. [50] constructed a Kronecker-factored Laplace function to approximate the posterior over the weights of a trained neural network, and they compared its performance with dropout-based Bayesian neural network. Louizos and Welling [51] adopted a stochastic gradient variational inference to estimate the posterior distribution over the weights of neural networks. Gal and Ghahramani [52] proposed that dropout in deep neural networks (NNs) could be used for approximating Bayesian inference in deep Gaussian processes. Later, Gal and Ghahramani [53] extended the idea of Monte Carlo dropout to convolutional neural networks and approximated the intractable posterior in Bayesian convolutional neural networks with Bernoulli variational distributions. In a similar way, Gal and Ghahramani [54] applied variational inference-based dropout technique in the long-short term memory (LSTM) and gated recurrent unit (GRU) models and assessed

the performance of Bayesian recurrent neural network in language modeling and sentiment analysis. Recently, Abdar et al. [55] performed a comprehensive review to examine a broad array of uncertainty quantification techniques in the field of deep learning.

In addition to various techniques for Bayesian deep learning, a few research studies have investigated explaining deep learning models from the standpoint of uncertainty quantification. For example, Kendall and Gal [56] developed a Bayesian deep learning-based framework to learn a mapping to aleatory uncertainty from input data. Hepp et al. [57] demonstrated how uncertainty-based estimation and explainability can be used for deep learning-based analysis of brain aging. Wickstrøm et al. [58] trained an ensemble of deep neural networks, and the standard deviation across the relevance scores was used to explain the decisions made by the deep learning ensemble.

## 3. Bayesian deep learning

In the context of Bayesian deep learning, a prior distribution (e.g., a Gaussian prior distribution: $\boldsymbol{W} \sim \mathcal{N}(0, I)$) is placed on the weights of the neural networks. Differing from deterministic neural networks, we optimize the loss function that is averaged over the possible weight combinations (referred to as marginalizations in Bayesian inference), and the posterior distributions of weights are inferred by combining the weights drawn from prior distributions and the observed data.

Suppose we have a neural network denoted as $f^{\boldsymbol{W}}(\cdot)$, where $f$ represents the structure of the neural network (e.g., number of layers and hidden units, choice of activation functions), and $\boldsymbol{W}$ is the collection of model parameters (neuron weights) to be estimated. Given a dataset $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\}$ and $\boldsymbol{Y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_n}\}$, the likelihood function of the model is defined as $p\left(\boldsymbol{y} | f^{\boldsymbol{W}}(\boldsymbol{x})\right)$ in the Bayesian context, then Bayesian inference aims to infer the posterior distribution $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y})$ over the model parameters $\boldsymbol{W}$; this computation can be mathematically formulated as below:

$$\overbrace{p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y})}^{\text{Posterior}} = \frac{\overbrace{p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W})}^{\text{Likelihood}} \overbrace{p(\boldsymbol{W})}^{\text{Prior}}}{p(\boldsymbol{Y}|\boldsymbol{X})} \tag{1}$$

Considering the highly non-linear nature of neural networks resulting from numerous neuron layers and different activation functions, analytical solution of the posterior probability $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y})$ is nearly impossible. Dropout variational inference is an approach that leverages variational inference to approximate Bayesian inference in deep neural networks [52–54], and it can be interpreted as a variational Bayesian approximation, where the approximating distribution is a mixture of Gaussian distributions with very low variances, and the mean of one Gaussian distribution is fixed at zero. More formally, the objective function of dropout variational inference in classification problems is shown in Eq. (2).

$$\min \quad \mathcal{L}(\vartheta, p) = -\frac{1}{n}\sum_{i=1}^{n} \log p\left(\boldsymbol{y_i} | f^{\widehat{\boldsymbol{W}}}(\boldsymbol{x_i})\right) + \frac{1-p}{2n}\|\vartheta\|^2 \tag{2}$$

where $\widehat{\boldsymbol{W}}$ is sampled from the approximated posterior distribution $q_{\vartheta^*}(\boldsymbol{W})$, $p$ is the dropout probability, $n$ is the number of data points in the dataset, and $\vartheta$ is the parameters to be optimized in the simple distribution $q_{\vartheta}(\boldsymbol{W})$.

In terms of implementation, Bayesian inference is approximated by training a model with dropout applied before each layer. The choice of which unit to drop in a neural network is randomly decided. At test time, we sample the weight from the approximated posterior distribution, apply dropout, and pass the sampled weights forward through the trained neural network multiple times; that is often referred to as Monte Carlo dropout. For classification problems, the epistemic uncertainty in model prediction can be modeled as:

$$p(y = c|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{Y}) \approx \frac{1}{M}\sum_{i=1}^{M} \text{Softmax}\left(f^{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x})\right) \tag{3}$$

where $c$ is the class label ($c = [1, 2, \ldots, n]$), $M$ represents the number of samples of the masked weights $\widehat{\boldsymbol{W}}_t$ drawn from the distribution $q_{\vartheta^*}(\boldsymbol{W})$, Softmax is a normalized exponential function commonly used as an activation function in the last layer of the neural network to get the probability of assigning input data to a given class.

## 4. Proposed methodology

In this paper, we aim to develop an end-to-end framework to help explain classification decisions made by deep learning models from an uncertainty quantification perspective. The framework of the proposed approach is illustrated in Fig. 1. As can be seen, the proposed approach consists of three steps. In the first step, we use entropy as a quantitative indicator to characterize the uncertainty in model prediction. Next, we compare the maximum entropy of a dropout-based Bayesian neural network over $n$ classes against a threshold $\gamma$ to decide whether to pass an instance to a human for further judgment. In the second step, we combine the prediction difference analysis (PDA) with Bayesian neural network to generate feature importance map and the uncertainty map. The feature importance map highlights the evidences supporting or not supporting the predicted class, thus explaining the decision made by the classifier; while the map of uncertainty in feature importance explains the sources of uncertainty that highly correlate with the uncertainty in model prediction. In the third step, we formulate a robustness-based design optimization problem to optimize the pixels with high feature importance uncertainty with the goal of strengthening the connection between input image and model prediction, thus enhancing the explainability of the machine learning model.

### 4.1. Entropy-based uncertainty quantification

Data collected by sensors is often corrupted by some noise; and this can occur in various domains [59]. In certain cases, due to the high noise in the collected data, it is difficult for humans to recognize objects in the image. Thus, blindly accepting the predictions of a deep learning model incurs high risk in undetected failures, e.g., out-of-the-domain cases, intentional attacks. As a result, it is important to develop a risk management tool to quantify the confidence in the prediction of the deep learning model and also have a deep learning model with the ability to say “I do not know" or “I am not sure". To address this issue, we propose to use entropy as a quantitative risk indicator to characterize the uncertainty in the model prediction. Suppose $\mathcal{X}$ is a discrete random variable with distribution given by:

$$\Pr(\mathcal{X} = x_i) = p_i \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{4}$$

The amount of information in the probability distribution $\mathcal{X}$ can be quantified using Shannon’s entropy, which is widely used in information theory [60–62]. Shannon’s entropy is mathematically defined as follows:

$$H(\mathcal{X}) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{5}$$

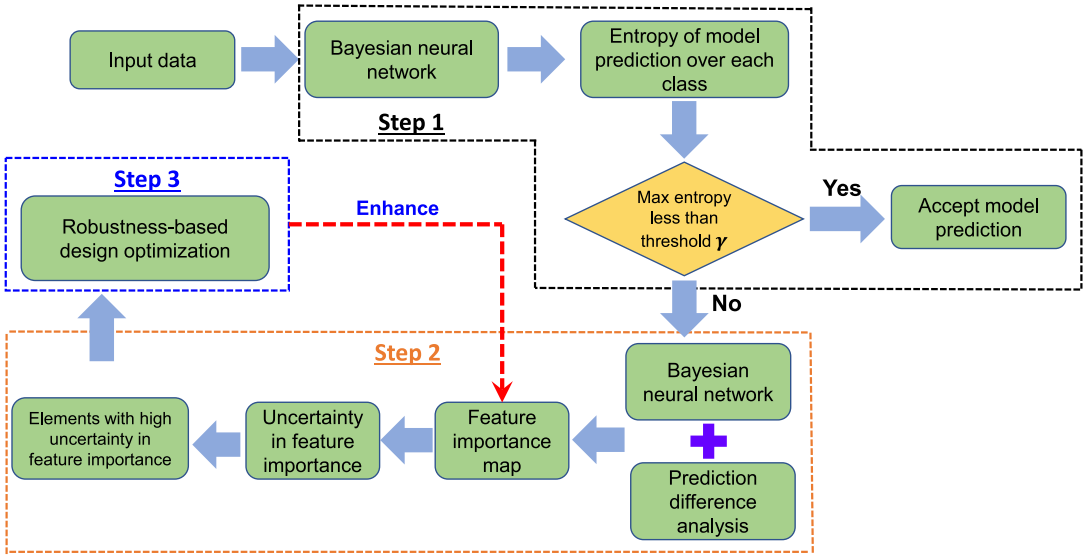



**Fig. 1.** Framework of the proposed methodology.

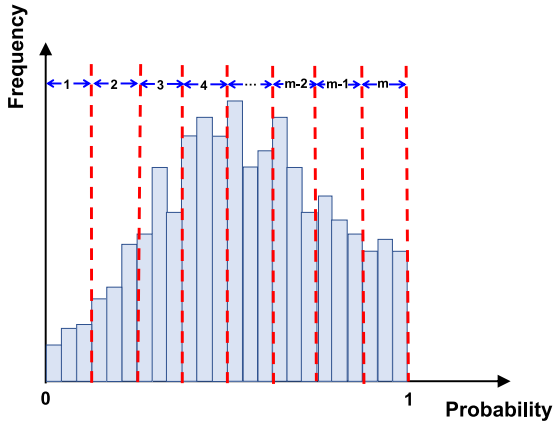



**Fig. 2.** The discretization of samples drawn from the approximated posterior distribution.

In the case of $\mathcal{X}$ being a continuous variable with probability density $p(x)$, the Shannon's entropy is defined as:

$$H(\mathcal{X}) = -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx. \tag{6}$$

In the context of a classification problem with $n$ classes $(1, 2, \ldots, n)$, as implied by the Shannon's entropy defined in Eq. (5), if the probability of labeling an input image as a specific class $c$ $(c = 1, 2, \ldots, n)$ is 1, then $H(\mathcal{X})$ reduces to the minimum value of zero, this denotes that there is no uncertainty in the model prediction. Whereas, when the probability of assigning the input image to any one of the $n$ classes is equal to $\frac{1}{n}$, the entropy reaches the maximum value reflecting that the trained model has the highest uncertainty in classifying the image.

In the case of Bayesian neural network, the output is a distribution over the classes. Therefore, we partition the distribution from the Bayesian neural network w.r.t. class $c$ into $m$ equal intervals. As shown in Fig. 2, the range of probabilities $[0, 1]$ is divided into $m$ equal intervals, then we estimate the probability $p^c\left(\frac{j-1}{m} < x \leq \frac{j}{m}\right)$, $c = 1, 2, \ldots, n; j = 1, 2, \ldots, m$ corresponding to the class $c$ using the equation below:

$$p^c\left(\frac{j-1}{m} < x \leq \frac{j}{m}\right) = \frac{\sum_{k=1}^{T}\left[\frac{j-1}{m} < S_k \leq \frac{j}{m}\right]}{T},$$

$$c = 1, 2, \ldots, n; j = 1, 2, \ldots, m. \tag{7}$$

where $T$ denotes the number of samples we draw from the approximated posterior distribution in the Bayesian neural network, $S_k$ denotes the $k$th sample drawn from the posterior distribution, $p^c\left(\frac{j-1}{m} < x \leq \frac{j}{m}\right)$ indicates the proportion of samples with probabilities falling within the range $\left(\frac{j-1}{m}, \frac{j}{m}\right]$ when the class label as predicted by the model is $c$, and $[\bullet]$ is a count function: $[\bullet]$ is 1 if $\bullet$ is true, and 0 if it is false.

Note that the specification of $m$ depends on the number of samples we draw. The more samples we draw, the closer we are to its true entropy value. In this paper, we draw 10,000 samples from the Bayesian neural network for each test instance. The range of probabilities $[0, 1]$ is divided into 100 equal intervals. Next, the probabilities pertaining to the $m$ discrete bins are then substituted into Eq. (4) to calculate the entropy associated with the model prediction specific to each class $c$. Thus, we have:

$$H^c(\mathcal{X}) = -\sum_{j=1}^{n} p^c\left(\frac{j-1}{m} < x \leq \frac{j}{m}\right) \log p^c\left(\frac{j-1}{m} < x \leq \frac{j}{m}\right) \tag{8}$$

Next, we define the maximum entropy over the $n$ classes with the equation shown in Eq. (9); when the maximum entropy $H^*$ is larger than a threshold value $\gamma$, it indicates that the model prediction has high uncertainty; thus, there is high risk involved if we use the model prediction in high stake decision-making activities. In this case, it is safer to pass such cases to a human for further judgment.

$$H^* = \max_{i \in [1,2,\ldots,n]} H^i(\mathcal{X}) \tag{9}$$

One issue is how to choose an appropriate value for the parameter $\gamma$. If $\gamma$ is set too low, then a large number of cases will be passed to humans for judgment; in contrast, if $\gamma$ is set too high, then the trained model might misclassify a large number of cases. To determine a proper threshold value of $\gamma$ for detecting out-of-domain cases, we propose to use the Empirical Cumulative Distribution Function (ECDF) of the entropy corresponding to 10% of data randomly sampled from the within-domain cases as well as the out-of-domain cases. Next, the intersection of ECDF corresponding to the sampled within-domain cases and the reversed ECDF corresponding to the sampled out-of-domain cases is used

as the threshold value for $\gamma$. By doing this, we reach a balance between the cases that should be labeled as out-of-domain cases and the cases that should be labeled as within-domain cases.

### 4.2. Feature importance uncertainty assessment

A Bayesian neural network characterizes the overall uncertainty in the model prediction, but in reality we are more interested in the factors in the input features contributing to model uncertainty in addition to the overall model uncertainty itself. In this paper, we apply the prediction difference analysis (PDA) approach developed by Robnik-Šikonja and Kononenko [63] to map the uncertainty in model prediction to the input features. The key idea in the PDA approach is to measure how the model prediction changes when an input feature is unknown [63], i.e., the difference between $p(c|\boldsymbol{x}_i)$ and $p\left(c|\boldsymbol{x}_i^{-j}\right)$, where $p\left(c|\boldsymbol{x}_i^{-j}\right)$ denotes the set of all input features in $\boldsymbol{x}_i$ except $x_i^j$, and $x_i^j$ refers to the $j$th feature in the input $\boldsymbol{x}_i$.

To approximate $p\left(c|\boldsymbol{x}_i^{-j}\right)$, we can simulate the absence of feature $x_i^j$ by marginalizing the feature:

$$p\left(c|\boldsymbol{x}_i^{-j}\right) = \sum_{x_i^j} p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right) p\left(c|x_i^j, \boldsymbol{x}_i^{-j}\right) \tag{10}$$

However, it is computationally expensive to approximate $p\left(c|\boldsymbol{x}_i^{-j}\right)$ with Eq. (10) because calculating $p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right)$ easily becomes infeasible in consideration of the large number of features in images. Following the heuristic developed by Zintgraf et al. [14], since the value of a pixel $x_i^j$ is highly dependent on the pixels of a local neighborhood around it, and the remaining pixels in an image are conditionally independent of $x_i^j$ given its local neighborhood, we mitigate the high computational burden of computing $p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right)$ by restricting the dependencies of pixel $x_i^j$ to only a small neighborhood around it. To be specific, for a given pixel $x_i^j$, we can approximate $p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right)$ by identifying a small patch $\widehat{\boldsymbol{x}}_i^j$ with a size of $l \times l$ that contains the pixel $x_i^j$. Thus, we have:

$$p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right) \approx p\left(x_i^j \middle| \widehat{\boldsymbol{x}}_i^{-j}\right) \tag{11}$$

where $\widehat{\boldsymbol{x}}_i^{-j}$ refers to the remaining pixels in the $l \times l$ patch $\widehat{\boldsymbol{x}}_i^j$ except the pixel $x_i^j$ rather than all the remaining pixels $\boldsymbol{x}_i^{-j}$, and $p\left(x_i^j \middle| \widehat{\boldsymbol{x}}_i^{-j}\right)$ conveys the idea of approximating the value of pixel $x_i^j$ by conditioning on the values of the surrounding pixels $\widehat{\boldsymbol{x}}_i^{-j}$.

In terms of implementation, we approximate the joint distribution of $x_i^j$ and its small local neighborhood patch $\widehat{\boldsymbol{x}}_i^{-j}$ as a multivariate Gaussian distribution.

$$\begin{aligned} p\left(x_i^j, \widehat{\boldsymbol{x}}_i^{-j}\right) &= \mathcal{N}\left(\begin{bmatrix} x_i^j \\ \widehat{\boldsymbol{x}}_i^{-j} \end{bmatrix}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \\ &= \mathcal{N}\left(\begin{bmatrix} x_i^j \\ \widehat{\boldsymbol{x}}_i^{-j} \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \end{aligned} \tag{12}$$

where $\mu$ is a vector with $l \times l$ elements, and $\sum$ is matrix with $(l \times l)^2$ elements for the $l \times l$-sized patch; their values can be estimated from image patches in the training data. Once the parameters of the multivariate Gaussian distribution are estimated, for a given neighborhood pixels $\widehat{\boldsymbol{x}}_i^{-j}$, the value of pixel $x_i^j$ can be estimated using the conditional Gaussian distribution below:

$$\begin{aligned} &p\left(x_i^j \middle| \widehat{\boldsymbol{x}}_i^{-j}\right) \\ &= \mathcal{N}\left(x_i^j; \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\left(\widehat{\boldsymbol{x}}_i^{-j} - \mu_2\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right) \end{aligned} \tag{13}$$

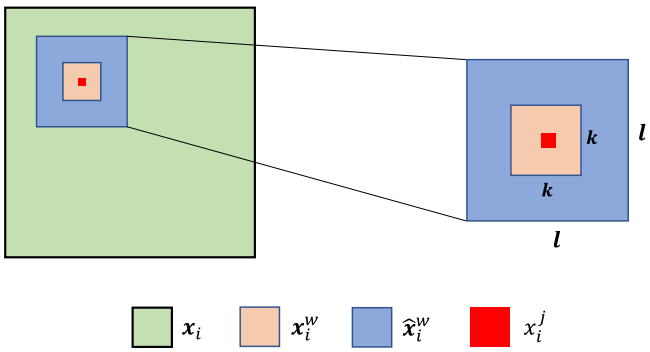



**Fig. 3.** Illustration of conditional sampling to approximate $p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right)$ in the image $\boldsymbol{x}_i$ (light green square), where $x_i^j$ refers to the pixel to be approximated (red solid square). In the figure, $\boldsymbol{x}_i$ refers to the input image, $\boldsymbol{x}_i^w$ (light yellow square) denotes the $k \times k$ patch surrounding the pixel $x_i^j$, $\widehat{\boldsymbol{x}}_i^w$ (light blue square) is a larger patch with size $l \times l$ containing $\boldsymbol{x}_i^w$, $\widehat{\boldsymbol{x}}_i^{-w}$ represents the $l \times l$-sized patch surrounding $\boldsymbol{x}_i^w$ but not including $\boldsymbol{x}_i^w$. **Note:** the figure is colored. For better view experience, visit the colored figure online.

Once we have $x_i^j$ drawn from the conditional Gaussian distribution shown above, we can measure the difference in model prediction between $p(c|\boldsymbol{x}_i)$ and $p\left(c|\boldsymbol{x}_i^{-j}\right)$ when the input feature $x_i^j$ is unknown following the weight of evidence method as proposed by Robnik-Šikonja and Kononenko [63]:

$$\mathrm{WE}_j(c|\boldsymbol{x}_i) = \log_2(\mathrm{odds}(c|\boldsymbol{x}_i)) - \log_2\left(\mathrm{odds}\left(c|\boldsymbol{x}_i^{-j}\right)\right) \tag{14}$$

where $\mathrm{odds}(z) = \frac{p(z)}{1-p(z)}$, and $\mathrm{WE}_j$ measures the importance of the input feature $x_i^j$ with respect to class $c$. The higher the value $\mathrm{WE}_j$, the more important pixel $x_i^j$ is to the class $c$.

Since the effect of perturbing a single pixel in an image is trivial on the prediction of the neural network models, we follow the second heuristic proposed by Zintgraf et al. [14] to conduct multivariate analysis instead of univariate analysis. Specifically, rather than perturb a single pixel $x_i^j$, we remove several features at once by strategically choosing a $k \times k$ adjacent pixels around $x_i^j$ in a sliding window fashion. As the sliding window can be overlapping, the importance of an individual pixel $x_i^j$ is obtained by averaging the importance values over the different patches it is in. Fig. 3 demonstrates the major concept of approximating $p\left(x_i^j \middle| \boldsymbol{x}_i^{-j}\right)$ using this heuristic. For a given input image $\boldsymbol{x}_i$ as represented by the light green block in Fig. 3, we select a patch $\boldsymbol{x}_i^w$ with size $k \times k$ containing $x_i^j$ in a sliding window fashion, and we place another patch $\widehat{\boldsymbol{x}}_i^w$ with a larger size $l \times l$ around $x_i^w$. The pixels in $x_w^i$ can be sampled conditional on the surrounding patch $\widehat{\boldsymbol{x}}_i^{-w}$. Once the pixels in $\boldsymbol{x}_i^w$ are sampled, we can use the weight of evidence as defined in Eq. (14) to compute the importance of each input feature $x_i^j$ with respect to class $c$.

The aforementioned procedure measures the importance of an input feature $x_i^j$ with respect to a given class $c$ in a deterministic model. In the context of a Bayesian neural network, we can extend the PDA approach to also quantify the uncertainty in feature importance in the input features by making multiple PDA runs, where dropout is applied in each run. We measure the importance of a given input feature with respect to a specific class in each run after dropout. With multiple runs, we quantify the uncertainty in feature importance as reflected in the model inputs. Quantification of uncertainty in feature importance helps to locate the sources of uncertainty that have significant impact on model prediction. Algorithm 4.2 demonstrates the detailed procedures in evaluating the uncertainty in feature importance with the PDA approach in a Bayesian neural network.

**Algorithm 1:** Evaluating feature importance and the uncertainty in feature importance with the prediction difference analysis (PDA) approach in Bayesian neural networks.

**Input:** an input image $\boldsymbol{x}_i$ with size $n \times n$, a classifier $p(c|\boldsymbol{x}_i)$, dropout rate $p$, dropout size $T$, an inner patch size $k$, an outer patch size $l$, and a sample size $S$

**Initialization:** WE = zeros(T, n, n), Count = zeros(T, n, n), $\mu^{\text{WE}}$ = zeros(n, n), $\sigma^{\text{WE}}$ = zeros(n, n)

**Process:**

**for** $t = 1$ to $T$ **do**
  Apply dropout to the classifier $p(c|\boldsymbol{x}_i)$ with dropout rate $p$
  **for** every patch $\boldsymbol{x}_i^w$ of size $k \times k$ in $\boldsymbol{x}_i$ **do**
    $\boldsymbol{x}_i^{'} = \text{copy}(\boldsymbol{x}_i)$
    $R^w = 0$
    Define the patch $\widehat{\boldsymbol{x}}_i^w$ of size $l \times l$ containing $\boldsymbol{x}_i^w$
    **for** $s = 1$ to $S$ **do**
      $\boldsymbol{x}_i^{w'} \leftarrow \boldsymbol{x}_i^w$ sampled from conditional multivariate distribution $p\left(\boldsymbol{x}_i^w \middle| \widehat{\boldsymbol{x}}_i^{-w}\right)$
      $R^w += p\left(c|\boldsymbol{x}_i^{'}\right)$, where $\boldsymbol{x}_i^{w'}$ replaces $\boldsymbol{x}_i^w$ in $\boldsymbol{x}_i^{'}$
    **end for**
    $p\left(c|\boldsymbol{x}_i^{-w}\right) = \frac{R^w}{S}$
    WE[t, coordinates of $\boldsymbol{x}_i^w$] += $\log_2(\text{odds}(c|\boldsymbol{x}_i)) - \log_2\left(\text{odds}\left(c|\boldsymbol{x}_i^{-w}\right)\right)$
    Count[t, coordinates of $\boldsymbol{x}_i^w$] += 1
  **end for**
  WE[t, coordinates of $\boldsymbol{x}_i^w$] = WE[t, coordinates of $\boldsymbol{x}_i^w$] /Count
**end for**
Average the feature importance: $\mu^{\text{WE}} = \frac{1}{T}\sum_{t=1}^{T} \text{WE}\left(t, \text{coordinates of } \boldsymbol{x}_i^w\right)$
Calculate the standard deviation of feature importance: $\sigma^{\text{WE}} = \sqrt{\frac{\sum_{t=1}^{T}\left(\text{WE}(\text{ t, coordinates of } \boldsymbol{x}_i^w) - \mu^{WE}\right)^2}{T}}$

**Output:** $\mu^{\text{WE}}, \sigma^{\text{WE}}$

### 4.3. Robustness-based design optimization for input image quality enhancement

In Section 4.2, we leverage the PDA approach to map the uncertainty in model prediction to model inputs. In this section, we formulate an uncertainty-driven optimization framework to enhance the quality of input images by optimizing the values of pixels that have high feature importance uncertainty. If an input image has a lot of noise, then the feature importance will be weak w.r.t. the predicted class, which results in poor explainability of model prediction. By reducing the noise in the input image through optimization properly, the connection between input features and model prediction will be strengthened, thus providing improved explainability of model prediction w.r.t. the predicted class. Suppose $\boldsymbol{S}_c(\boldsymbol{x}_i)$ denotes the set of samples that are drawn from the Bayesian neural network for the input image $\boldsymbol{x}_i$ regarding a given class $c$, $\boldsymbol{\theta}_c(\boldsymbol{x}_i)$ denotes the set of pixels in image $\boldsymbol{x}_i$ that dominantly contribute to the uncertainty in model prediction w.r.t. the predicted class $c$, our goal is formulate a robustness-based design optimization problem to optimize the pixels $\boldsymbol{\theta}_c(\boldsymbol{x}_i)$ to maximize the mean value of the samples $\boldsymbol{S}_c(\boldsymbol{x}_i)$ drawn from the Bayesian neural network while minimizing the uncertainty in model prediction for the class $c$. Mathematically, the problem can be formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{\theta}_c(\boldsymbol{x}_i)}{\text{Min}} \quad & (1 - \mu(\boldsymbol{S}_c(\boldsymbol{x}_i))) + \sigma(\boldsymbol{S}_c(\boldsymbol{x}_i)) \\ s.t. \quad & \min(\boldsymbol{x}_i) \le v_k \le \max(\boldsymbol{x}_i), \quad \forall k \in \boldsymbol{\theta}_c(\boldsymbol{x}_i). \end{aligned} \tag{15}$$

where $\mu(\boldsymbol{S}_c(\boldsymbol{x}_i))$ and $\sigma(\boldsymbol{S}_c(\boldsymbol{x}_i))$ denote the mean and standard deviation of the samples $\boldsymbol{S}_c(\boldsymbol{x}_i)$ drawn from the posterior distribution in the Bayesian neural network, respectively; $v_k$ denotes the value of the $k$th pixel in the set $\boldsymbol{\theta}_c(\boldsymbol{x}_i)$, $\min(x_i)$ and $\max(x_i)$ represent the minimum and maximum pixel value in image $\boldsymbol{x}_i$. Note that each time when we change the pixel values in $\boldsymbol{\theta}_c(\boldsymbol{x}_i)$, the samples $\boldsymbol{S}_c(\boldsymbol{x}_i)$ from the Bayesian neural network are updated accordingly.

As shown in Eq. (15), we formulate a robustness-based design optimization problem with the objective of minimizing a function combining the mean and standard deviation of the samples $\boldsymbol{S}_c(\boldsymbol{x}_i)$ for the sake of maximizing the Bayesian neural network's judgment towards labeling the input image $\boldsymbol{x}_i$ as the class $c$. Since both $\mu(\boldsymbol{S}_c(\boldsymbol{x}_i))$ and $\sigma(\boldsymbol{S}_c(\boldsymbol{x}_i))$ are non-negative, the function defined in Eq. (15) reaches the minimum value of 0 when $\boldsymbol{S}_c(\boldsymbol{x}_i)$ has the value of 1 and $\sigma(\boldsymbol{S}_c(\boldsymbol{x}_i))$ takes the value of 0. As the Bayesian neural network model is a black box to the optimizer, we take a meta-heuristic approach – differential evolution – to optimize the pixel values in this paper. Differential evolution (DE) is a well-known population-based optimization algorithm for solving complex multi-modal optimization problems [64–66]. DE does not use gradient information for optimization, thus it does not require the analytical form of the objective function, which makes it appropriate for solving the robustness-based design optimization formulated in this paper.

## 5. Numerical evaluations

In this section, we use two numerical examples to illustrate the insights gained by implementing uncertainty quantification through Bayesian neural network on two individual datasets: MNIST and CIFAR-10 [67]. In particular, we show how to set the value of the threshold parameter $\gamma$, and evaluate the performance of this parameter in helping recognize out-of-domain cases in deep learning tasks. More importantly, we quantify pixel-wise feature importance uncertainty with respect to the predicted class, and identify the sources of uncertainty in input data that contribute to uncertainty in model prediction. The two numerical examples follow the same structure summarized as below:

1. We briefly describe the architecture of convolutional neural network used for MINIST and CIFAR-10 image classification in this paper, and summarize how the out-of-domain data is prepared for each numerical example;
2. We derive entropy for each image in the normal and out-of-domain dataset. An empirical cumulative density functions (ECDF)-based approach is developed to determine the value of the threshold parameter $\gamma$. Using $\gamma$, we compile a report to summarize the performance of the entropy-based approach in identifying out-of-domain data.
3. We visualize and discuss the feature importance map as well as the uncertainty map of several instances, and demonstrate how the robustness-based design optimization can enhance the relevance between input features and model prediction through optimizing pixels with high uncertainty in feature importance.

### 5.1. MNIST example

We train the LeNet convolutional neural network using the 60,000 images in the MNIST training data [68] for $4 \times 10^5$ iterations with a learning rate of $1 \times 10^{-4}$, where a dropout rate

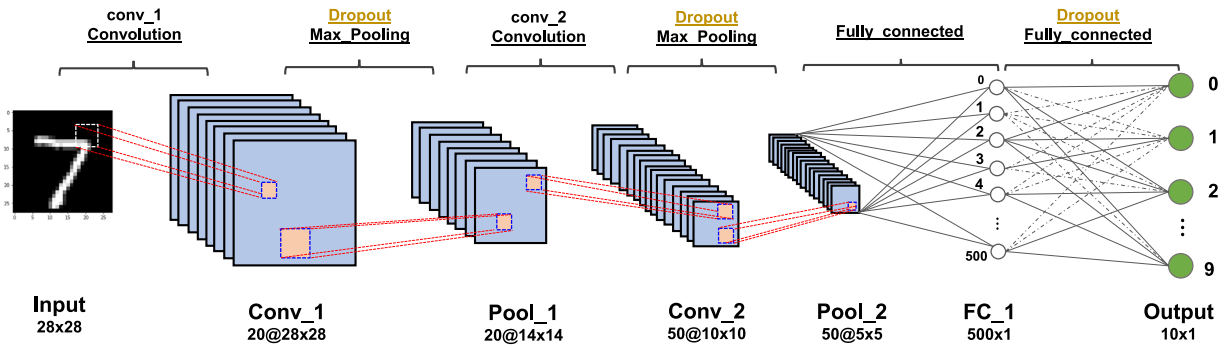



**Fig. 4.** The architecture of LeNet convolutional neural network, where three layers of dropout are applied.

**Table 1**
Summary of datasets used in the MNIST classification problem.

| Dataset name | Number of images |
|---|---|
| MNIST training data | 60,000 |
| MNIST test data | 10,000 |
| Randomly generated data | 1,000 |
| Not-MNIST data | 1,754 |

**Table 2**
Performance of probabilistic LeNet convolutional neural network in classification of within-domain and out-of-domain images in the MNIST sample.

| Dataset | Percentage of cases with $H^* > \gamma$ | Percentage of cases with $H^* < \gamma$ |
|---|---|---|
| MNIST test data | 19.86% | 80.14% |
| Randomly generated data | 78.56% | 21.44% |
| Not-MNIST data | 86.89% | 13.11% |
| Out-of-domain cases | 83.86% | 16.14% |

of 0.5 is applied before each weight layer. Fig. 4 summarizes the architecture of the LeNet convolutional neural network. At a high level, the LeNet convolutional neural network consists of 2 convolutional layers, 2 max pooling layers, 1 fully connected layer, and three dropout layers following the two convolutional layers and the fully connected layer.

In addition to the original MNIST dataset, we manually generate a set of images by randomly mixing two images from the MNIST training dataset via weighted sum as instances of images with high uncertainty. To make sure that the generated image does have features from each individual MNIST image, we generate the weight for each image within the range $[0.4, 0.8]$ randomly. The normalized weights are then used to fuse the two MINIST images. Besides, an open dataset available at http://yaroslavvb.com/upload/notMNIST/ consisting of English letters is used as an additional out-of-domain dataset to check whether the trained network has the ability to say "I do not know". The randomly generated mixed images and the images in the non-MNIST dataset are referred to as out-of-the-domain cases in the following discussion. Table 1 summarizes the types of images in each dataset used in this numerical example. In total, 2,754 out-of-domain cases are used in the MNIST study.

Following the method described in Section 4.1, we randomly take 10% of instances from the test data in the MNIST dataset as well as 10% of instances from the out-of-domain cases, and construct empirical cumulative density functions (ECDF) for the two datasets in order to determine $\gamma$'s value. Fig. 5 shows the ECDF of the maximum entropy corresponding to the 10% within-domain and the reversed ECDF for 10% out-of-domain cases, where X axis indicates the maximum entropy over the ten classes, and Y axis indicates the ECDF of the maximum entropy. In particular, when $\gamma$'s value is set at 0.92, which is the intersection point of the two ECDFs, then in the 10% within-domain cases, 78.6% of cases have entropy values less than $\gamma$; whereas, in the 10% out-of-domain cases, 78.6% of instances have entropy values larger than $\gamma$.

Table 2 summarizes the performance of the probabilistic LeNet convolutional neural network on the remaining 90% dataset of within-domain and out-of-domain cases when $\gamma$ is set at 0.92. From the table, we can see that the selected threshold value 0.92 for the parameter $\gamma$ has worked as an effective safety barrier that helps to identify a large proportion of images falling outside the domain of the dataset that the LeNet convolutional neural network is trained for. In particular, in the randomly generated data, 78.56% of images have entropy larger than 0.92; whereas the proportion of images with entropy value larger than $\gamma$ is 86.89% for the non-MNIST dataset, which is even higher than that of the randomly generated dataset. While for the MNIST test data, only 19.86% of images have entropy value larger than $\gamma$. Even though 19.86% of test data in the MNIST data are mislabeled as out-of-domain cases, in safety-critical applications, it is affordable to scrutinize these cases with high model prediction uncertainty by human means. Note that the LeNet model has not been trained with any out-of-domain case; the measure of model prediction uncertainty based on entropy has demonstrated a robust performance in recognizing out-of-domain images in the probabilistic LeNet convolutional neural network. In contrast, if a deterministic LeNet convolutional neural network is used to classify these images, none of the image will be labeled as out-of-domain cases since there is no concept of uncertainty in deterministic models, and the trained model is constrained in the domain of predicting integer numbers as outputs even if the input image is not an integer number.

Fig. 6 illustrates several images sampled from the three datasets listed in Table 2, the maximum entropy associated with each image, and the distributions of the predicted class. As can be seen, for the images in Fig. 6(a), although they are sampled from the MNIST test data, it is challenging to label them as digits with high confidence even by human means. The unclear traits in these images pose high risk if we move forward with the predictions from the LetNet model blindly without looking into the images ourselves. Likewise, it is more challenging to classify the images in Figs. 6(b) and 6(c) as integer numbers because they are either generated by blending images consisting of different integer numbers or the object in the image is English letter rather than integer number. Naturally, these images have even higher entropy values. In practice, the cases with entropy higher than $\gamma$ can be passed over to decision makers for further examination, thus increasing the safety of the applications against intentional attacks.

Following the methodology described in Section 4.2, we use an inner patch with a size of 3 ($k = 3$) and an outer patch with a size of 7 ($l = 7$) to simulate the absence of $3 \times 3$

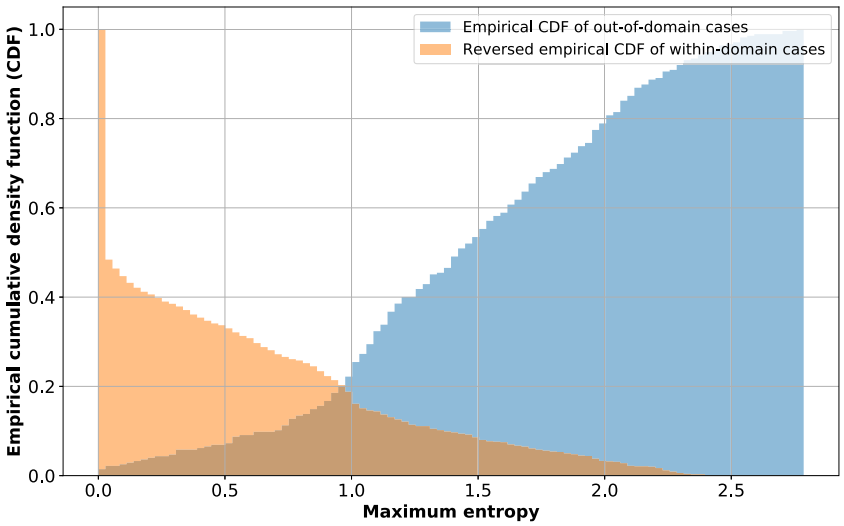
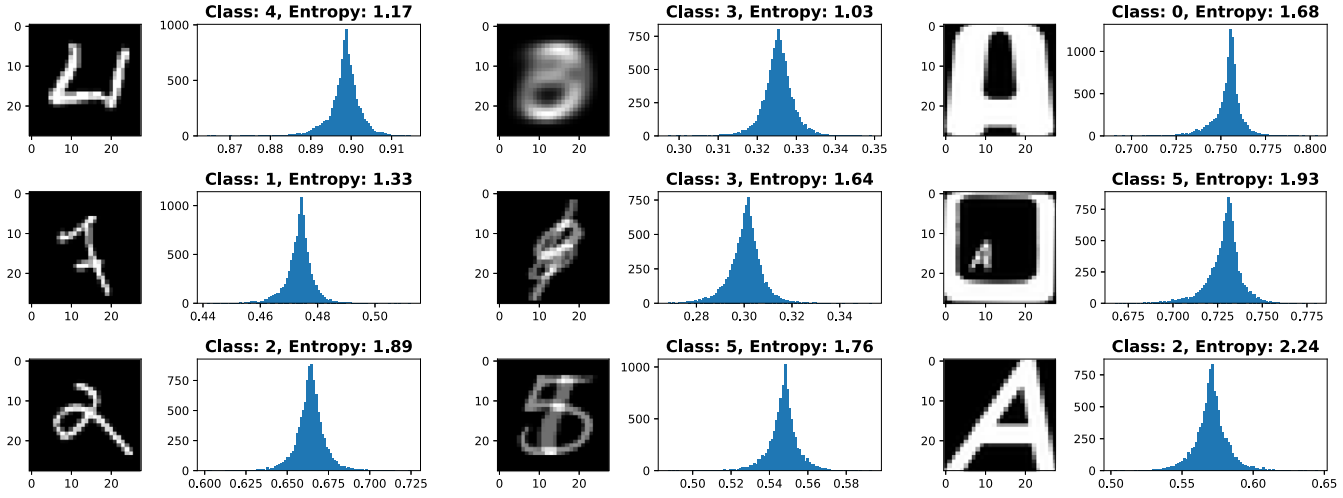



**Fig. 5.** The determination of entropy threshold value in the MNIST example.




(a) Entropy values for images sampled from the MNIST test data

(b) Entropy values for images sampled from the randomly generated data

(c) Entropy values for images sampled from the not-MNIST data

**Fig. 6.** Maximum entropy values for several images sampled from the three datasets and the corresponding distributions of the class with maximum entropy as predicted by the Bayesian LeNet convolutional neural network.

patch in the input image following a sliding window fashion. With respect to each inner patch, 20 samples ($S = 20$) are generated from the conditional multivariate distribution. In the Bayesian LeNet convolutional neural network, we generate 100 Monte Carlo samples ($T = 100$) of dropout in the LeNet model to calculate the uncertainty in feature importance. Note that PDA requires the specification of a target class, thus the evidence supporting or not supporting is relative to the target class. Fig. 7 visualizes the feature importance maps of several images randomly sampled from the MNIST test dataset w.r.t. the predicted class. Differing from Fig. 7, Fig. 8 visualizes feature importance on several instances randomly sampled from the non-MNIST dataset, and the last column of Fig. 8 highlights pixels in input images that highly correlate with the model prediction uncertainty. As shown in the second column of Fig. 8, the key parts constituting each digit number (e.g., loops, vertical stroke, curves) are shown in red pixels, and these red pixels provide supportive evidence for the trained model to label them as the predicted classes. In contrast, the blue pixels are evidence of not supporting the predicted class.

Another interesting observation is that uncertainty is low in the regions where the red pixels are located, while uncertainty is high in the regions where the input pixels confusing the model's judgment w.r.t the predicted class are located. For example, in Example A of Fig. 8, the input image is a mix of digit numbers 3 and 9. When the LeNet model labels the input image as number 3, there is high uncertainty in the areas where the salient features for the number 9 are located (see the salience map in the last column of Example A). In particular, the left-falling stroke in the input image significantly affects the model's prediction. Likewise, in Example B of Fig. 8, the vertical stroke and the large dot next to the loop at the bottom greatly contributes to the uncertainty of the model prediction when the input image is classified as number 6. These pixels makes the image different from the training examples, thereby decreasing the model's confidence when making predictions w.r.t. the predicted class.

In Example C of Fig. 8, the trained CNN model classifies the English letter "D" as number 5. As can be seen in the feature importance map of Example C, the shape formed by the red pixels is similar to the structure of a rotated number 5, which explains

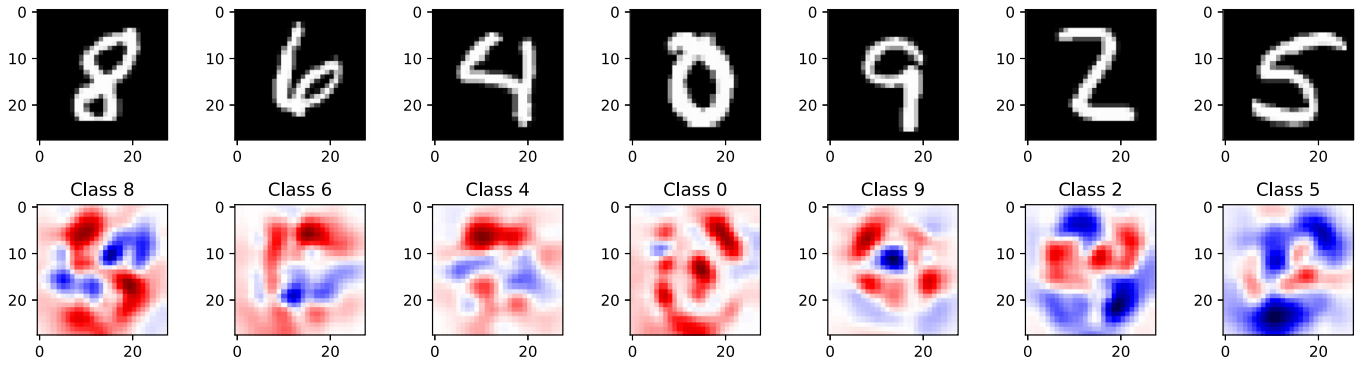



**Fig. 7.** Visualization of feature importance map on several images randomly sampled from the MNIST test data. Note that red pixels are evidences in support of the predicted class, and blue pixels are against the predicted class.

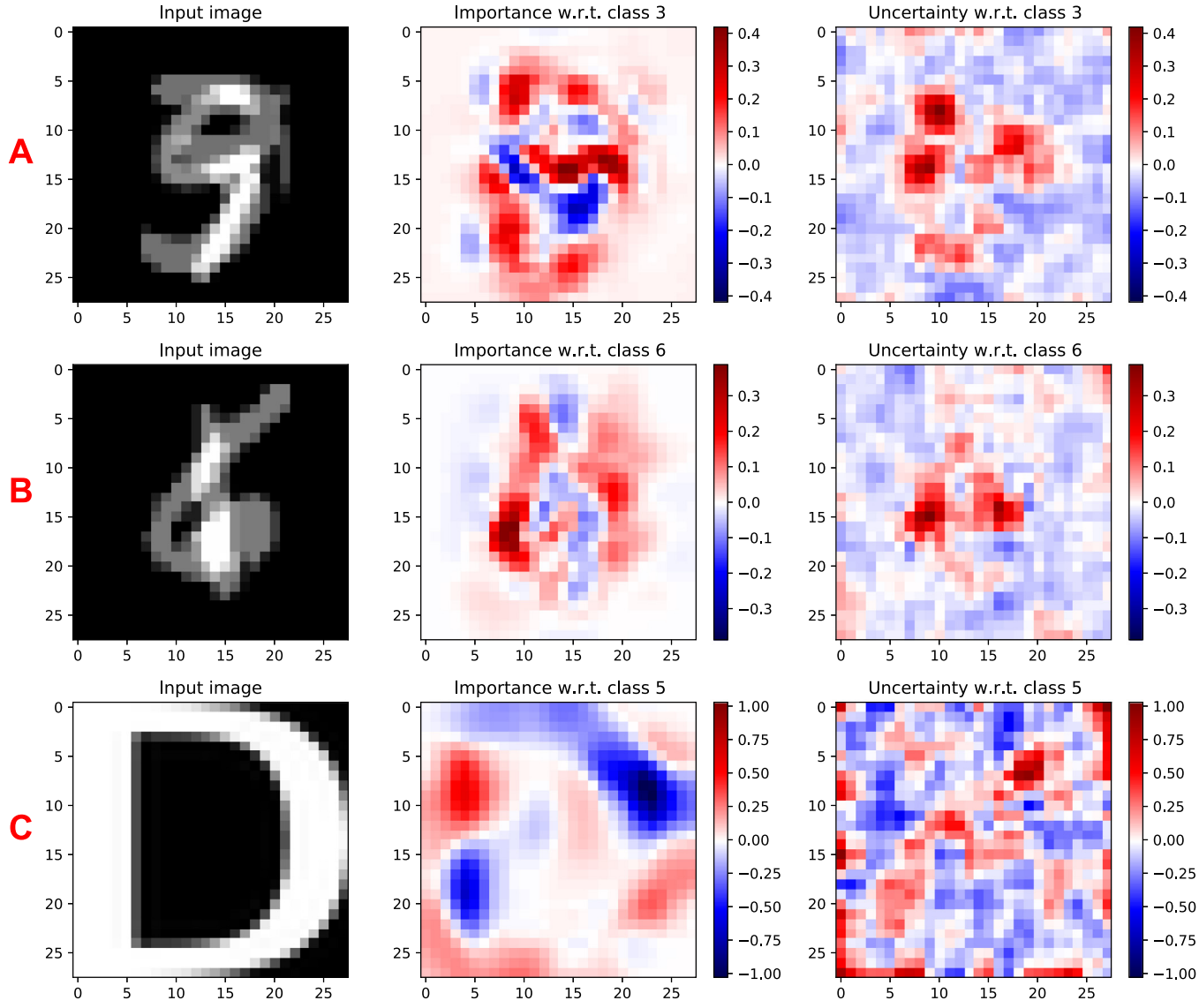



**Fig. 8.** Visualization of feature importance of each individual pixel and the contribution of each individual pixel to model prediction uncertainty w.r.t. the predicted class in the MNIST example. The first column shows the original input images, the second column indicates the feature importance of each input pixel w.r.t. the predicted class. The third column indicates the contributions of individual pixels to model prediction uncertainty. A positive change in red pixels further increases the uncertainty of the trained model w.r.t. the predicted class.

the rationality of labeling the English letter "D" as digit number 5. Whereas, the blue pixels in the feature importance map of Example C compose an area that interferes with the judgment of the trained CNN model, which thereby increases the uncertainty of the model when classifying the input image as number 5. As a result, a lot of uncertainty emerges from the areas where blue

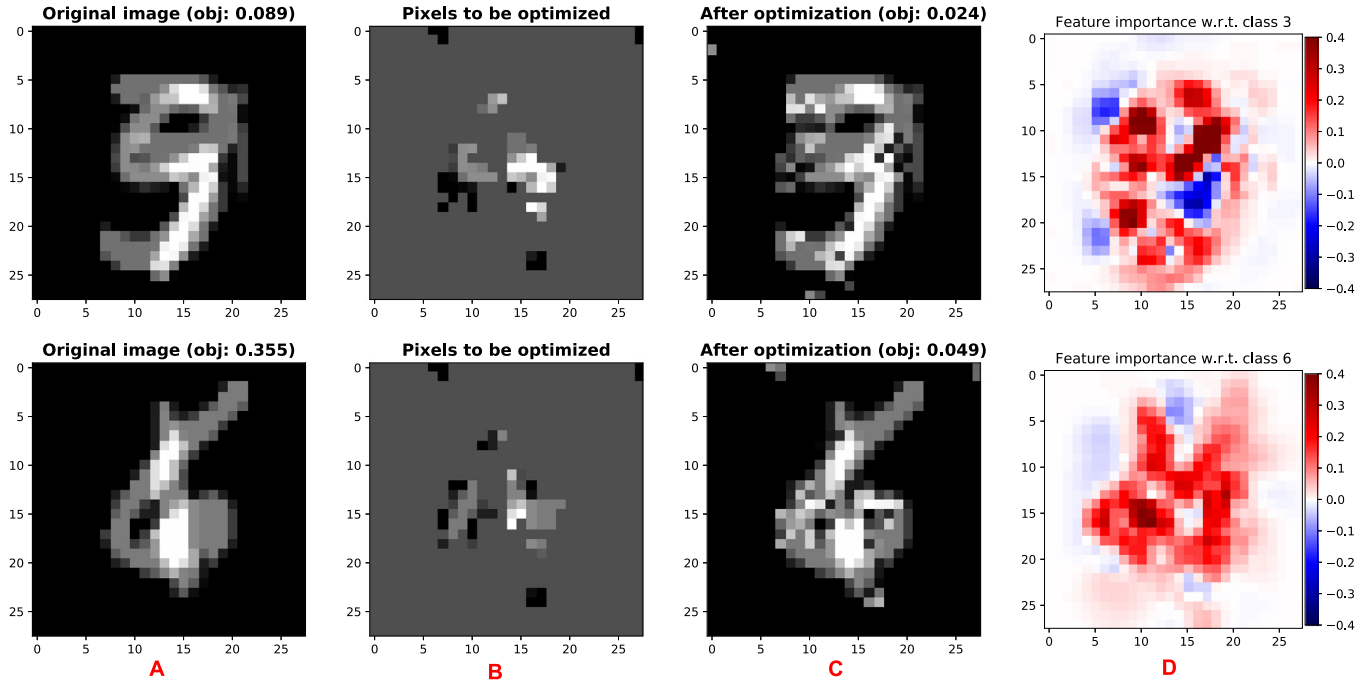



**Fig. 9.** The optimization results for sample images in the MNIST example.

pixels are located. The pixels with high uncertainty are shown in the feature uncertainty map of Example C.

Interestingly, since predictive difference analysis is relative to a target class, when the classifier makes wrong predictions, it still shows the evidence supporting or not supporting the wrong decision. Such information itself facilitates the understanding of why the classifier makes such wrong decisions. Another insight in the MNIST example is that although all the input images are outside the domain of the model, they are composed of low-level features with different degrees of outside-of-the-domain flavors. Specifically, in the first set of examples, input images are mixtures of integer numbers. As a result, the uncertainty in feature importance is highly concentrated on the areas having features that confuses model's judgment. While in Example C, the uncertainty in feature importance is scattered. The dispersed uncertainty in feature importance also explains the high entropy values for the out-of-domain examples.

Next, we perform robustness-based design optimization for the examples shown in Fig. 8 using the differential evolution approach. In the differential evolution method, we set the population size as 200, and the maximum iteration number as 50. Fig. 9 compares the original image and the image after optimization. Fig. 9(B) shows the pixels that are to be optimized by the differential evolution algorithm. In this study, we determine the pixels to be optimized by picking up the pixels that have feature importance uncertainty higher than the 90% percentile out of all the pixels. As shown in Fig. 9(C), after optimization, the objective function defined in Eq. (15) has reduced significantly for both cases. Fig. 9(D) visualizes feature importance for the two images after the optimization. As can be observed, the red areas supporting labeling the image as the target class has increased significantly, while blue areas decreasing the probability of that class has reduced accordingly. In terms of explainability, the vertical stroke in the original image of number 3 has faded out after the optimization. Besides, key features that play a decisive role in helping to identify these numbers has become salient in both cases. For example, regarding the number 6, the color of the pixels constituting the structure of number 6 has changed from gray to white. Such a significant change makes key features that help to identify the digit number stand out in contrast to the background color in the image. In summary, the differential evolution approach has successfully enhanced the explainability of the images through the optimization of pixels with high feature importance uncertainty.

### 5.2. CIFAR-10

CIFAR-10 is a collection of computer-vision datasets used for object recognition, and it is widely used for training machine learning and computer vision algorithms [69]. The CIFAR-10 dataset consists of 60,000 32 × 32 color images in 10 classes with 6000 images per class. The training dataset has 50,000 images and the test dataset has 10,000 images. Whereas, the CIFAR-100 dataset has 100 classes with 600 images per class. Each class has 500 training images and 100 testing images. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. In this paper, we use the 10,000 test images in CIFAR-100 dataset as out-of-domain cases to verify the performance of a CNN model trained with CIFAR-10 dataset in detecting the change of the domain of the data. CIFAR-100 and CIFAR-10 have low overlap in class labels except the label pickup truck in CIFAR-100. In this paper, we treat pickup truck in CIFAR-100 as a subcategory of the label truck in CIFAR-10.

In this paper, we trained a convolutional neural network with 50,000 images in the CIFAR-10 dataset. The architecture of the convolutional neural network is shown in Fig. 10. As can be seen, the network consists of two convolutional layers, three dropout layers, two max-pooling layers, and two fully-connected layers. Each convolutional layer is followed by a dropout layer with a dropout rate of 0.5, and a 2 × 2 max-pooling layer in order. The convolutional layers have the same 5 × 5 sliding window with a padding size of 2, and they generate 192 feature maps. The first fully-connected layer follows right after the second max-pooling layer and generates 1,000 outputs. The first fully-connected layer is followed by a dropout layer, where a dropout rate of 0.5 is applied. The last fully-connected layer transforms the 1,000 outputs into the 10 class labels that are used in the CIFAR-10 dataset. The convolutional neural network is trained

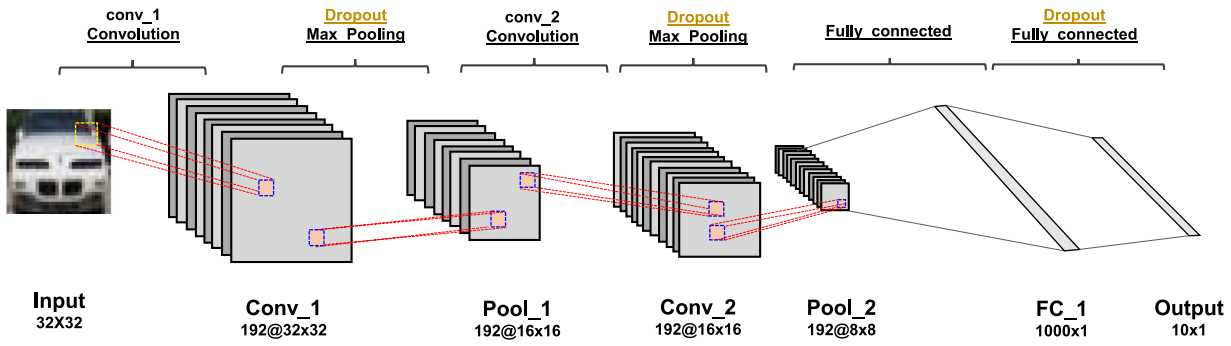



**Fig. 10.** The architecture of convolutional neural network trained for the CIFAR-10 dataset.

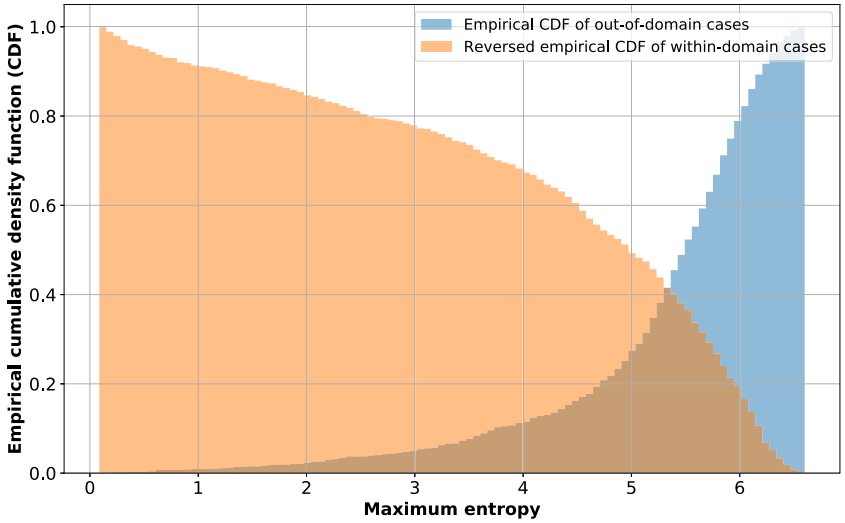



**Fig. 11.** The determination of entropy threshold value in the CIFAR-10 example.

for 100,000 iterations with a base learning rate of $1 \times 10^{-3}$, a momentum of 0.9 and a weight decay of $4 \times 10^{-3}$. Following a similar approach, we draw 10,000 samples for each test case in the Bayesian neural network, compute the entropy for each test instance, and use the maximum entropy of each case across the ten classes as a criterion to determine whether a case is within the domain or outside the domain. To determine the threshold value of the parameter $\gamma$, 10% of instances in the test dataset are randomly chosen. Fig. 11 shows the empirical CDFs of the maximum entropy corresponding to the 10% CIFAR-10 and CIFAR-100 test instances. Note that since we treat pickup truck as a subcategory of truck, we exclude 100 cases with labels pickup truck from the CIFAR-100 dataset when we select the 10% out-of-domain cases for deciding the value of $\gamma$. From Fig. 11, we observe that when $\gamma$ is set at 5.325, a trade-off that balances the labeling of out-of-domain and within-domain cases correctly is reached. In particular, when $\gamma$ is 5.325, in the 10% within-domain cases, 60.4% of cases have entropy values less than $\gamma$; whereas, in the 10% out-of-domain cases, 60.5% of instances have entropy values larger than $\gamma$.

Table 3 summarizes the performance of the trained Bayesian convolutional neural network in classifying the remaining 90% within-domain and out-of-domain images in CIFAR-10 and CIFAR-100. From the statistics shown in Table 3, we observe that the threshold value 5.325 has worked effectively when classifying the different categories of images. Specifically, only 42.59% of images out of the 9,000 cases in the CIFAR-10 test data have maximum entropy values exceeding $\gamma$, while nearly thirds of the images are correctly labeled as cases with maximum entropy less than the threshold value. In contrast, with respect to the out-of-domain test dataset CIFAR-100, more than thirds of images are correctly identified as high-risk cases due to the high entropy values exceeding $\gamma$; thus, only 38.97% of images are labeled as within-domain cases. Similarly, regarding the class "truck" in the CIFAR-10 dataset, only 20.60% of the cases have entropy larger than $\gamma$, whereas 79.40% of images are recognized as within-domain cases correctly. Similarly, out of the 100 CIFAR-100 pickup truck images, only 43% of instances have maximum entropy exceeding the threshold value, and the maximum entropy of the remaining 57% dataset is less than the threshold. Based on the data summarized in Table 3, we conclude that although the domain of the data has changed significantly, the proposed entropy-based measure works robustly in labeling a large portion of images correctly, which proves that the entropy-based uncertainty measure can be used as an effective approach to detect the change of data domain in machine learning.

**Table 3**
Performance of probabilistic convolutional neural network in classifying with-domain and out-of-domain images in the CIFAR-10 example.

| Dataset | Percentage of cases with $H^* > \gamma$ | Percentage of cases with $H^* < \gamma$ | No of cases |
|---|---|---|---|
| CIFAR-10 test data | 42.59% | 57.41% | 9,000 |
| CIFAR-100 test data | 61.03% | 38.97% | 8,910 |
| CIFAR-10 truck | 20.60% | 79.40% | 1,000 |
| CIFAR-100 pickup truck | 43.00% | 57.00% | 100 |

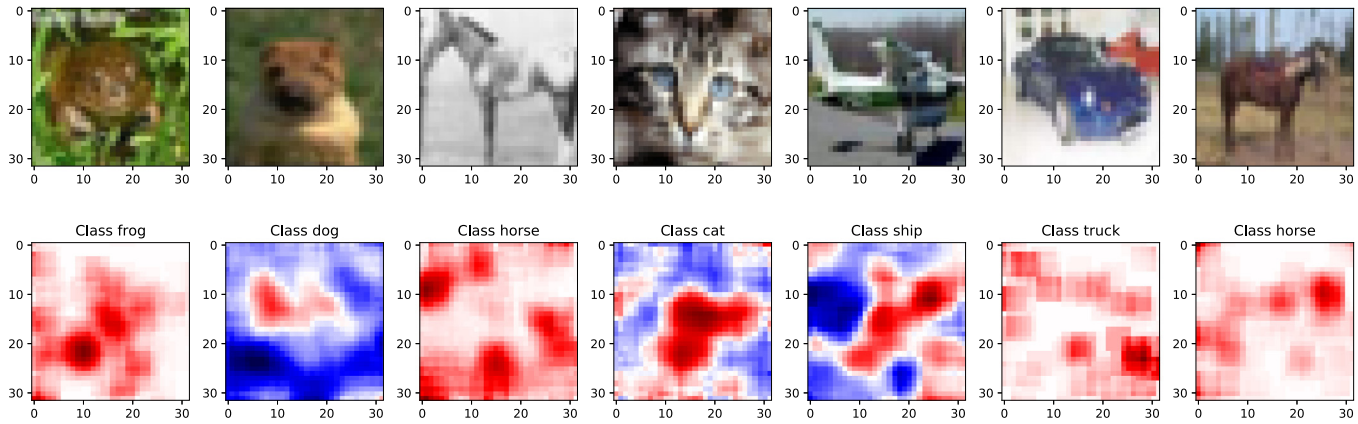



**Fig. 12.** Visualization of feature importance map on several images randomly sampled from the CIFAR-10 test data. Note that red pixels are evidences in support of the predicted class, and blue pixels are against the predicted class.

One fact worthy of mention is that the trained convolutional neural network has an overall accuracy of 61.72% on the CIFAR-10 test dataset, which partially helps explain the relatively low values of the trained model in identifying out-of-domain cases in the CIFAR-10 example in comparison with the MNIST example. In other words, the convolutional neural network misclassifies 38.28% of images in the training data. The deficiency in the trained model propagates to the Bayesian neural network when we use it to make predictions on the within-domain and out-of-domain images as well as to calculate the entropy for each test instance. As a result, the accuracy of labeling within-domain and out-of-domain images is significantly limited by the quality of the trained model. If we could improve the resolution of the trained model in recognizing a variety of images, then the accuracy of the Bayesian neural network in detecting the change of the domain of the images through the entropy-based measure will improve accordingly.

Similarly, we demonstrate the methodology described in Section 4.2; specifically, we use a patch with an inner size of $5 \times 5$ ($k = 5$) and another patch with an outer size of $7 \times 7$ ($l = 7$) to mimic the absence of $5 \times 5$ patch in the input image following a sliding window fashion. Regarding each $5 \times 5$ patch, we generate 40 Monte Carlo samples using a multivariate distribution conditional on the $2 \times 2$ local neighborhood pixels surrounding the $5 \times 5$ patch to simulate the effect of removal of the $5 \times 5$ patch from the input image. Fig. 12 illustrates the feature importance map w.r.t. the predicted class for several images sampled from the test data in CIFAR-10. As can be observed, the feature importance maps successfully identify the major evidences supporting or not supporting each classification decision. In some cases, as the object has very similar features with the background in terms of color and texture, certain parts of background is also labeled as supporting evidence.

To characterize the uncertainty in feature importance, we perform 100 Monte Carlo runs with dropout in the Bayesian neural network. Fig. 13 illustrates the visualization of evidence in the input images labeled by the model as supporting and not supporting the class, and identifies the pixels in the input images contributing to the uncertainty in model prediction. The first set of images (Examples A and B) is randomly selected from the CIFAR-10 test dataset, while Example C is randomly picked from the test dataset in CIFAR-100. As can be seen from Fig. 13, the developed approach correctly identifies features that are closely relevant to model's judgment w.r.t. each input images in Examples A and B. Even though the model misclassifies a bird as a deer in Example A, the red pixels constitute an object with structure very similar to the skeleton of the deer. In Example B, the model recognizes important features in the input image and correctly labels it as a dog. In both Examples A and B, the uncertainty concentrates on the four edges of the image.

In contrast, since Example C is outside the domain what the model is trained for, the object composed by the features consisting of red pixels is far away from what the model labels the images as. Specifically, Example C shows an image of a kangaroo, since the model is constrained in the domain of the 10 predefined classes, it labels the input image as a ship, which is the object with the shape most similar to the kangaroo from the eye of the trained model. Besides, the uncertainty in feature importance highly overlaps with the areas of evidences consisting of red pixels and supporting the model's judgment. As mentioned before, since the identified features as represented by red pixels are yet far away from what the labeled object is supposed be like, the red pixels are the minimum set of features that enable the model to make such a prediction out of the 10 predefined classes. Nonetheless, the model still lacks some informative features to have adequate confidence to label the input image as the corresponding object. In other words, only when the features are redundant for a given object, in this case, even if a small portion of the input image is altered, it does not impact the overall model judgment in classification. In contrast, in the case of minimum set of input features, the removal or alteration of any single pixel has a significant impact on the prediction of the model because the evidence is already insufficient for the model to make decisions, let alone more evidence to be removed. Even though the classification on Example C is wrong, the salient feature importance is still valuable to facilitate the understanding of model predictions. Another observation in Example C is that the background in the input image has extremely similar features to the traits in the objects, thus the background has high uncertainty in the evidence against the predicted class, which are represented as blue pixels in Fig. 13.

Similar to the first numerical example, we perform robustness-based design optimization to enhance the explainability of the deep learning model with the differential evolution approach. To do that, we set the population as 1,000, and we run the differential evolution algorithm for 100 iterations. Fig. 14 illustrates the computational results of the algorithm. The subfigure in the middle shows the pixels to be optimized, and we select these pixels as their feature importance uncertainty ranks 95% out of all the pixels. In total, there are 52 pixels. Considering the image has three channels (red, green, blue), there are 52*3 = 156 decision variables to be optimized. The subfigure in the

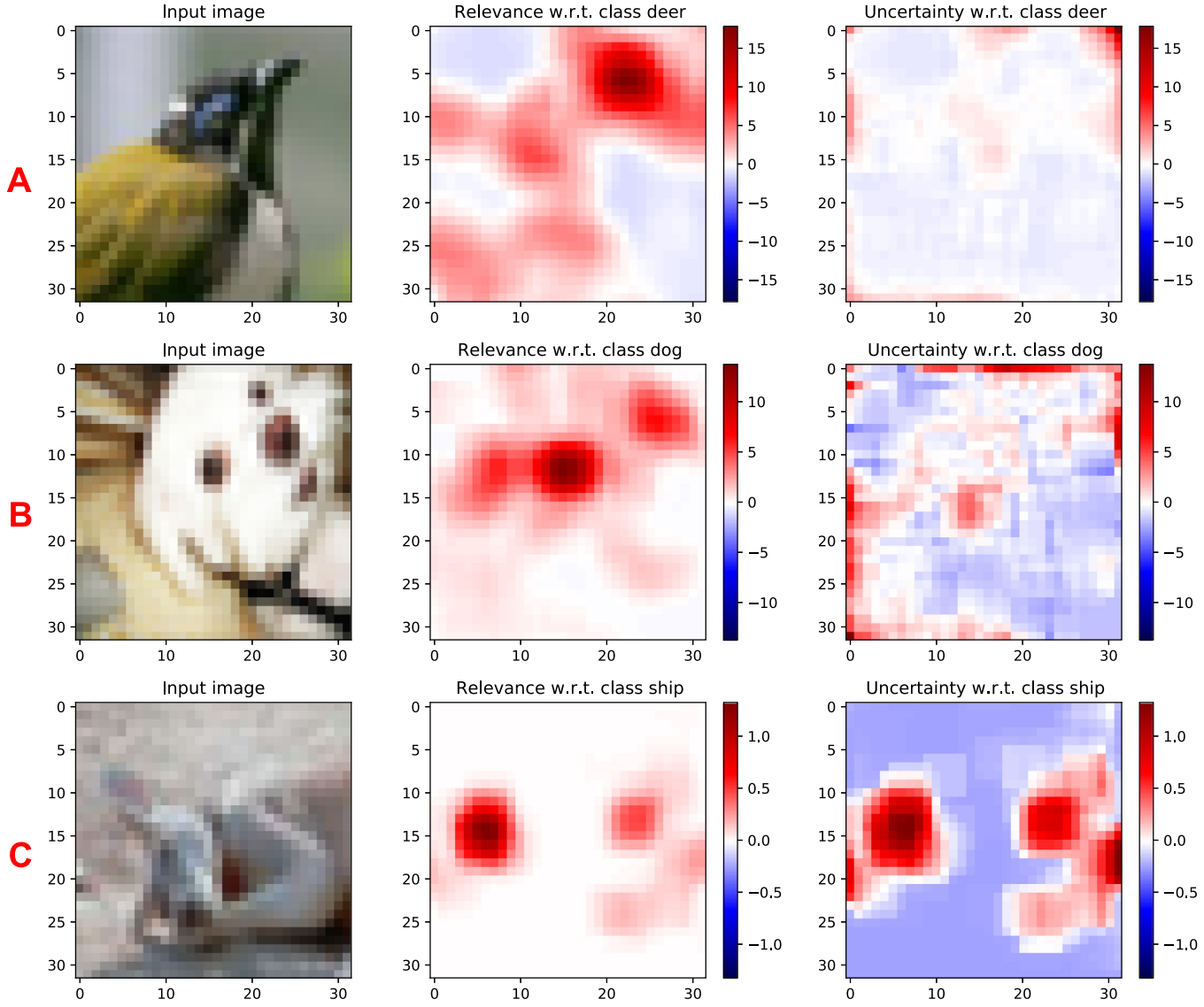



**Fig. 13.** Visualization of feature importance of each individual pixel and the contribution of each individual pixel to model prediction uncertainty w.r.t. the predicted class in the CIFAR-10 example. The first column shows the original input images, the second column indicates the feature importance of each input pixel w.r.t. the predicted class. Note that red pixels are the evidence in support of predicted class, and blue pixels are against the predicted class. The third column indicates the contributions of individual pixels to model prediction uncertainty. A positive change in red pixels further increases the uncertainty of the trained model w.r.t. the predicted class.

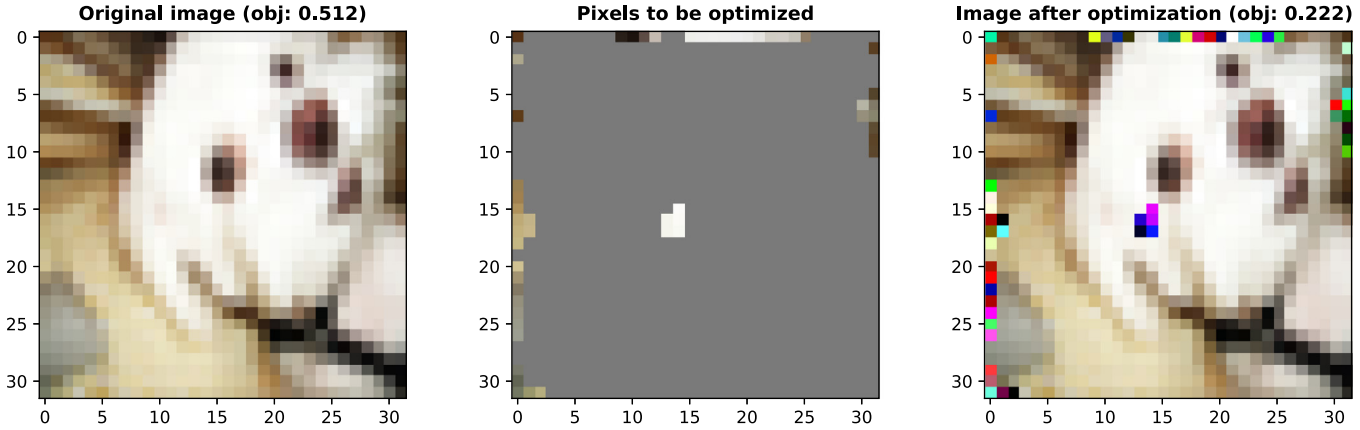



**Fig. 14.** The optimization result for the sample image in the CIFAR10 example.

right shows the image after optimization. Although the value of the objective function has reduced from 0.512 to 0.222, the optimized image is difficult to comprehend. As mentioned earlier, the trained convolutional neural network has an overall accuracy of 61.72% on the CIFAR-10 test dataset, which is relatively low. The inadequate training and relatively low resolution in the trained CNN model might propagate to model prediction, which might eventually lead to inaccurate representation of feature importance uncertainty. From this perspective, the uncertainty arising from model and the data needs to be separated to help further investigate this issue.

## 6. Discussion

This article presents an uncertainty quantification (UQ)-based framework to explain decisions made by deep learning models in image classifications. The proposed framework explains classification decisions made by deep neural networks from three perspectives. First, the entropy-based uncertainty measure explains the significant difference in neural network responses when dealing with within-domain and out-of-domain images. The parameter $\gamma$ works as an effective safety metric to identify cases that are either too noisy or outside the domain of the trained model. The introduction of entropy-based threshold parameter $\gamma$ safeguards the usage of deep learning models in an open-world environment.

Secondly, we combine the Bayesian neural network with predictive difference analysis (PDA) to quantify feature importance uncertainty, from which we identify pixels that are primarily responsible for the uncertainty in model prediction. In this context, the feature importance-based techniques reviewed in Section 2 are unsuitable to be adopted in this paper due to two major reasons. (I) As mentioned earlier, we are interested in the uncertainty pertaining to the feature importance of each pixel in the input image. To achieve this goal, multiple runs are needed using a probabilistic machine learning framework while the techniques reviewed in Section 2 are only applicable in the deterministic context. (II) Most of the feature importance-based techniques reviewed in Section 2 extract the importance of each feature by calculating the difference in feature importance before and after adding random perturbation (e.g., Gaussian noise), where Gaussian noise is typically used to simulate the random perturbation. Hence, the local neighborhood around a given pixel is not considered; whereas in images, the value of a given pixel is highly dependent on the pixels of its local neighborhood.

In this paper, we combine the Bayesian neural network with PDA. By doing this, we highlight pixels in the input image that highly correlate with model prediction uncertainty. Identifying the locations of sources of uncertainty in the input image helps to interpret varied amounts of uncertainty as reflected in model prediction. Moreover, locating pixels with high feature importance uncertainty also provides valuable information if an instance is passed to a human for further judgment.

Thirdly, after pixels with high feature importance uncertainty are located, we formulate a robustness-based design optimization problem to increase feature importance by optimizing the pixels having high variability in feature importance in the input image. A differential evolution approach is adopted to perform the optimization. The optimized pixels significantly reduce model prediction uncertainty, and enhance the relevance between input features and model prediction. The increased feature importance helps the human decision maker to better comprehend the model's prediction w.r.t. the predicted class because there is more evidence supporting (or less evidence against) the predicted class.

## 7. Conclusion

In this paper, an uncertainty quantification-based framework is formulated to interpret decisions made by deep neural networks. The developed framework consists of three steps. In the first step, we leverage an entropy-based indicator to measure the uncertainty in model prediction and define an entropy-based threshold parameter to decide when to accept or reject predictions made by the deep learning model. Next, we integrate Bayesian neural network with PDA to quantify pixel-wise feature importance uncertainty w.r.t. the predicted class. Thirdly, we formulate a robustness-based design optimization problem by optimizing image pixels with high feature importance uncertainty for the purpose of increasing the relevance between input features and model prediction. Two datasets for image classification are used to demonstrate the performance of the developed methodology in explaining decisions made by deep learning models.

Future work can be carried out in several directions. First, it is computationally expensive to draw a large number of Monte Carlo samples from dropout-based Bayesian neural network to identify the pixels responsible for model prediction uncertainty; thus it is important to develop a fast and scalable approach (e.g., surrogate model) to approximate Bayesian inference in neural networks for large-scale applications. Improvement in computational speed will enable Bayesian neural networks to be adopted in time-sensitive applications that require near real-time decision making. Secondly, explainability of the neural network models is characterized here in a qualitative manner; it is desirable to develop metrics to measure and quantitatively examine the consistency of explainability in various scenarios. The development of appropriate evaluation metrics will also significantly facilitate the performance comparison of different models in explaining classifier decisions. Along this front, there are two major challenges to be addressed in order to facilitate comparison of explainability as provided by different methods. (I) In the current literature, several techniques are available to measure the explainability of machine learning models in an overall sense, but are not suitable in the context of attribute and uncertainty quantification as considered in this paper; future work can address this need and develop suitable quantitative metrics. (II) the way we define explanation is through the visualization of feature importance map and uncertainty map with respect to each predicted class. How to judge one feature importance map (or uncertainty map) is better (or worse) than another one is another challenge because it is not clear what the desired feature importance map (or the baseline) is. Last but not least, it is worth investigating the applications of the proposed methodology to other domains, such as natural language processing (NLP), sentiment analysis, medical images, etc., where different types of data are being processed to acquire actionable information (such as language translation, estimating distance from sensor data in autonomous vehicles, learning from customer feedback, etc.). These efforts will greatly deepen our comprehension in the decision-making process of various AI systems.

## CRediT authorship contribution statement

**Xiaoge Zhang:** Conceptualization, Methodology, Experimental validation, Writing – original draft. **Felix T.S. Chan:** Writing – review & editing. **Sankaran Mahadevan:** Writing – review & editing, Methodology, Discussion.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The work presented in this paper is supported by Centre for Advances in Reliability and Safety (CAiRS), an InnoHK Research Cluster of HKSAR Government, and the Research Committee of The Hong Kong Polytechnic University under project code 1-BE6V.

## References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[2] X. Zhang, S. Mahadevan, Bayesian neural networks for flight trajectory prediction and safety assessment, Decis. Support Syst. 131 (2020) 113246.
[3] A. Aldweesh, A. Derhab, A.Z. Emam, Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, Knowl.-Based Syst. 189 (2020) 105124.
[4] X. Li, Z. Du, Y. Huang, Z. Tan, A deep translation (GAN) based change detection network for optical and SAR remote sensing images, ISPRS J. Photogramm. Remote Sens. 179 (2021) 14–34.
[5] X. Zhang, S. Mahadevan, Ensemble machine learning models for aviation incident risk prediction, Decis. Support Syst. 116 (2019) 48–63.
[6] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (2019) 206–215.
[7] L.R. Chai, Uncertainty Estimation in Bayesian Neural Networks and Links to Interpretability, (Master's thesis), University of Cambridge, 2018.
[8] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (2019) 828–841.
[9] A.V. Konstantinov, L.V. Utkin, Interpretable machine learning with an ensemble of gradient boosting machines, Knowl.-Based Syst. 222 (2021) 106993.
[10] D. Gunning, Explainable artificial intelligence (XAI), Def. Adv. Res. Proj. Agency 2 (2017) 2.
[11] Z. Yang, S. Dong, HAGERec: hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation, Knowl.-Based Syst. 204 (2020) 106194.
[12] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PloS one 10 (2015) e0130140.
[13] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[14] L.M. Zintgraf, T.S. Cohen, T. Adel, M. Welling, Visualizing deep neural network decisions: Prediction difference analysis, in: The Fifth International Conference on Learning Representations (ICLR 2017), 2017.
[15] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017) 4765–4774.
[16] S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (2020) 56–67.
[17] B. Kim, J. Park, J. Suh, Transparency and accountability in AI decision support: Explaining and visualizing convolutional neural networks for text information, Decis. Support Syst. (2020) 113302.
[18] C. Yang, W. Zhou, Z. Wang, B. Jiang, D. Li, H. Shen, Accurate and explainable recommendation via hierarchical attention network oriented towards crowd intelligence, Knowl.-Based Syst. 213 (2021) 106687.
[19] T. Spinner, U. Schlegel, H. Schäfer, M. El-Assady, explAIner: A visual analytics framework for interactive and explainable machine learning, IEEE Trans. Vis. Comput. Graphics 26 (2019) 1064–1074.
[20] C. Leibig, V. Allken, M.S. Ayhan, P. Berens, S. Wahl, Leveraging uncertainty information from deep neural networks for disease detection, Sci. Rep. 7 (2017) 1–14.
[21] Y. Kwon, J.-H. Won, B.J. Kim, M.C. Paik, Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation, Comput. Statist. Data Anal. 142 (2020) 106816.
[22] A. Jungo, R. Meier, E. Ermis, E. Herrmann, M. Reyes, Uncertainty-driven sanity check: Application to postoperative brain tumor cavity segmentation, in: International Conference on Medical Imaging with Deep Learning, 2018.
[23] S. Sharma, J. Henderson, J. Ghosh, CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 166–172.
[24] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, L. Daniel, Evaluating the robustness of neural networks: An extreme value theory approach, in: International Conference on Learning Representations, 2018.
[25] N. Burkart, M.F. Huber, A survey on the explainability of supervised machine learning, J. Artif. Intell. Res. 70 (2021) 245–317.
[26] M. Wu, M. Hughes, S. Parbhoo, M. Zazzi, V. Roth, F. Doshi-Velez, Beyond sparsity: Tree regularization of deep models for interpretability, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.
[27] S. Ntalampiras, I. Potamitis, N. Fakotakis, Probabilistic novelty detection for acoustic surveillance under real-world conditions, IEEE Trans. Multimed. 13 (2011) 713–719.
[28] M. Ahmed, A.N. Mahmood, J. Hu, A survey of network anomaly detection techniques, J. Netw. Comput. Appl. 60 (2016) 19–31.
[29] P. Vorburger, A. Bernstein, Entropy-based concept shift detection, in: Sixth International Conference on Data Mining, IEEE, 2006, pp. 1113–1118.
[30] W. Cheng, E. Hüllermeier, W. Waegeman, V. Welker, Label ranking with partial abstention based on thresholded probabilistic models, in: NIPS 2012, vol. 25, Neural Information Processing Systems Foundation (NIPS), 2012, pp. 2510–2518.
[31] X. Ding, Y. Li, A. Belatreche, L.P. Maguire, An experimental evaluation of novelty detection methods, Neurocomputing 135 (2014) 313–327.
[32] R. Domingues, M. Filippone, P. Michiardi, J. Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses, Pattern Recognit. 74 (2018) 406–421.
[33] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data 6 (2012) 1–39.
[34] N. Görnitz, M. Kloft, K. Rieck, U. Brefeld, Toward supervised anomaly detection, J. Artif. Intell. Res. 46 (2013) 235–262.
[35] M.A. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Process. 99 (2014) 215–249.
[36] M. Wu, J. Ye, A small sphere and large margin approach for novelty detection using training data with outliers, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 2088–2092.
[37] Y. Chen, X. Dang, H. Peng, H.L. Bart, Outlier detection with the kernelized spatial depth function, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2008) 288–305.
[38] K. Zhang, M. Hutter, H. Jin, A new local distance-based outlier detection approach for scattered real-world data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2009, pp. 813–822.
[39] M. Masud, J. Gao, L. Khan, J. Han, B.M. Thuraisingham, Classification and novel class detection in concept-drifting data streams under time constraints, IEEE Trans. Knowl. Data Eng. 23 (2010) 859–874.
[40] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, J. Mohd-Yusof, Combating label noise in deep learning using abstention, in: International Conference on Machine Learning, PMLR, 2019, pp. 6234–6243.
[41] A. Garcia, C. Clavel, S. Essid, F. d'Alché Buc, Structured output learning with abstention: Application to accurate opinion prediction, in: International Conference on Machine Learning, PMLR, 2018, pp. 1695–1703.
[42] B. Kompa, J. Snoek, A.L. Beam, Second opinion needed: communicating uncertainty in medical machine learning, NPJ Digit. Med. 4 (2021) 1–6.
[43] D. Alvarez-Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[44] R. Guidotti, S. Ruggieri, On the stability of interpretable models, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.
[45] J. DeYoung, S. Jain, N.F. Rajani, E. Lehman, C. Xiong, R. Socher, B.C. Wallace, ERASER: A benchmark to evaluate rationalized NLP models, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4443–4458.
[46] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D.I. Inouye, P.K. Ravikumar, On the (in) fidelity and sensitivity of explanations, Adv. Neural Inf. Process. Syst. 32 (2019).
[47] M. Du, N. Liu, F. Yang, S. Ji, X. Hu, On attribution of recurrent neural network predictions via additive decomposition, in: The World Wide Web Conference, 2019, pp. 383–393.
[48] I. Mollas, N. Bassiliades, G. Tsoumakas, Altruist: argumentative explanations through local interpretations of predictive models, 2020, arXiv preprint arXiv:2010.07650.
[49] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International Conference on Machine Learning, PMLR, 2015, pp. 1613–1622.
[50] H. Ritter, A. Botev, D. Barber, A scalable laplace approximation for neural networks, in: 6th International Conference on Learning Representations, vol. 6, International Conference on Representation Learning, 2018.
[51] C. Louizos, M. Welling, Multiplicative normalizing flows for variational Bayesian neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2218–2227.
[52] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, 2016, pp. 1050–1059.
[53] Y. Gal, Z. Ghahramani, Bayesian convolutional neural networks with Bernoulli approximate variational inference, 2015, arXiv preprint arXiv:1506.02158.

[54] Y. Gal, Z. Ghahramani, A theoretically grounded application of dropout in recurrent neural networks, in: Adv. Neural Inf. Process. Syst., 2016, pp. 1019–1027.
[55] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion 76 (2021) 243–297.
[56] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision? in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 5580–5590.
[57] T. Hepp, D. Blum, K. Armanious, B. Schölkopf, D. Stern, B. Yang, S. Gatidis, Uncertainty estimation and explainability in deep learning-based age estimation of the human brain: Results from the German National Cohort MRI study, Comput. Med. Imaging Graph. 92 (2021) 101967.
[58] K.K. Wickstrøm, K. ØyvindMikalsen, M. Kampffmeyer, A. Revhaug, R. Jenssen, Uncertainty-aware deep ensembles for reliable and explainable predictions of clinical time series, IEEE J. Biomed. Health Inf. (2020).
[59] G. Algan, I. Ulusoy, Image classification with deep learning in the presence of noisy labels: A survey, Knowl.-Based Syst. 215 (2021) 106771.
[60] C.E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Comput. Commun. Rev. 5 (2001) 3–55.
[61] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theory 37 (1991) 145–151.
[62] J. Wu, J. Sun, L. Liang, DEA cross-efficiency aggregation method based upon Shannon entropy, Int. J. Product. Res. 50 (2012) 6726–6736.
[63] M. Robnik-Šikonja, I. Kononenko, Explaining classifications for individual instances, IEEE Trans. Knowl. Data Eng. 20 (2008) 589–600.
[64] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2010) 4–31.
[65] Y.O. Serrano-Silva, Y. Villuendas-Rey, C. Yáñez-Márquez, Automatic feature weighting for improving financial Decision Support Systems, Decis. Support Syst. 107 (2018) 78–87.
[66] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.
[67] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Technical Report, University of Toronto, 2009.
[68] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Process. Mag. 29 (2012) 141–142.
[69] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90.