remote sensing

MDPI

Article

Uncertainty Quantification in Data Fusion Classifier for Ship-Wake Detection

Maice Costa [1,†], Daniel Sobien [1,†], Ria Garg [2], Winnie Cheung [2], Justin Krometis [3] and Justin A. Kauffman [1,*]

[1] National Security Institute, Virginia Tech, Arlington, VA 22203, USA
[2] Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24061, USA
[3] National Security Institute, Virginia Tech, Blacksburg, VA 24060, USA
* Correspondence: jakauff@vt.edu
† These authors contributed equally to this work.

Abstract: Using deep learning model predictions requires not only understanding the model's confidence but also its uncertainty, so we know when to trust the prediction or require support from a human. In this study, we used Monte Carlo dropout (MCDO) to characterize the uncertainty of deep learning image classification algorithms, including feature fusion models, on simulated synthetic aperture radar (SAR) images of persistent ship wakes. Comparing to a baseline, we used the distribution of predictions from dropout with simple mean value ensembling and the Kolmogorov—Smirnov (KS) test to classify in-domain and out-of-domain (OOD) test samples, created by rotating images to angles not present in the training data. Our objective was to improve the classification robustness and identify OOD images during the test time. The mean value ensembling did not improve the performance over the baseline, in that there was a –1.05% difference in the Matthews correlation coefficient (MCC) from the baseline model averaged across all SAR bands. The KS test, by contrast, saw an improvement of +12.5% difference in MCC and was able to identify the majority of OOD samples. Leveraging the full distribution of predictions improved the classification robustness and allowed labeling test images as OOD. The feature fusion models, however, did not improve the performance over the single SAR-band models, demonstrating that it is best to rely on the highest quality data source available (in our case, C-band).

Keywords: data fusion; uncertainty quantification; simulated data; deep neural network; synthetic aperture radar

Citation: Costa, M.; Sobien, D.; Garg, R.; Cheung, W.; Krometis, J.; Kauffman, J.A. Uncertainty Quantification in Data Fusion Classifier for Ship-Wake Detection. *Remote Sens.* **2024**, *16*, 4669. https://doi.org/10.3390/rs16244669

Academic Editors: Dusan Gleich and Gong Cheng

Received: 4 October 2024
Revised: 17 November 2024
Accepted: 10 December 2024
Published: 14 December 2024

1. Introduction

Understanding a model's prediction confidence is essential to using the predictions in a decision process, knowing when to trust the automated process and when to request support from a human agent. We can achieve this by quantifying the uncertainty of a model's predictions, but this requires additional processing or features in the model. There are two main types of uncertainty, aleatoric and epistemic [1]. Aleatoric uncertainty, also known as data uncertainty or intrinsic uncertainty, stems from the inherent randomness and variability present in the data. Epistemic uncertainty, also known as model uncertainty or knowledge uncertainty, arises from the limitations of the model, and can be reduced through a better model architecture, additional training data, and optimization. Characterizing epistemic uncertainty is important when training a model on limited data, since gaps in the input space require extrapolation to make predictions, resulting in more uncertainty. Limited data also leads to underfitting and overfitting problems, where a model fails to generalize the predictions to unseen data. In this study, we used Monte Carlo dropout (MCDO) in a deep learning model trained on limited data to help assess the epistemic uncertainty and improve model performance on out-of-domain (OOD) test samples.

In addition to collecting or generating more training data, using regularization and ensemble methods, such as MCDO, can help capture different aspects of the data distribution and reduce uncertainty. MCDO is a regularization technique that consists of randomly setting to zero some fraction of a model's weights (associated with the neurons in a neural network) during each forward and backward pass of training. This regularization process prevents the network from depending too heavily on any single neuron and becoming too specialized. As a result, the network becomes more robust to changes in the input data, preventing overfitting and improving the generalizability of the model [2,3].

While dropout was originally designed as a regularization technique, it has also found applications in uncertainty quantification and network pruning at inference time. Yarin Gal and Zoubin Ghahramani offered a probabilistic interpretation of dropout that sparked more interest in techniques for uncertainty estimation and combinations of regularization and non-linearity techniques [4,5]. The use of dropout during inference creates different thinned networks, effectively training an ensemble of models. During testing, the predictions obtained from multiple forward passes are treated as samples from a distribution of predictions. Variance, or other statistical measures, of the predictions from multiple forward passes can estimate the uncertainty associated with the model's predictions. The higher the variance and/or shift in output distribution from the training data, the more uncertain the model is about its prediction for a particular input sample.

Uncertainty quantification in neural networks and the use of dropout-based Bayesian approximation are active areas of research. Extending the dropout method to uncertainty estimation aims to improve the accuracy, efficiency, and generalizability in various deep learning architectures and applications. Variational dropout [6] considers dropout rates as learned parameters with their own distributions. Dropout variational inference is implemented with a Monte Carlo simulation of stochastic forward passes at test time [5]. A further connection with Bayesian deep learning is exploited to develop an alternative to grid-search over dropout probabilities with concrete dropout [7], which introduces a continuous relaxation of the binary dropout mask used in traditional dropout, enabling automatic tuning of the dropout probability. Additional extensions have adapted dropout to serve recurrent networks and Transformer architectures, both as regularization and as a compression technique to reduce network sizes [8–10].

Most of the existing works using MCDO for uncertainty quantification relied on calculating the mean and standard deviation or variance for the final prediction and quantifying the uncertainty. Numerous previous studies have applied this technique for medical imagery from real clinical settings [11], agricultural images of plant leaves for disease detection [12], and physical systems to model and predict the physical parameters of those systems [13]. Relying on these types of uncertainty quantifications, however, makes the assumption that predictions should be normally distributed. Another common approach for uncertainty quantification with MCDO uses predictive entropy. Habibpour et al. [14] applied predictive entropy using the mean of class predictions with Shannon's entropy [15] to detect credit card fraud. Abdar et al. [16] used predictive uncertainty with MCDO, along with deep ensembling, in existing convolutional neural network (CNN) models in classifying medical images of skin cancer. Another possibility is to use the 95% confidence interval as an uncertainty metric, which defines the bounds of the mean of the distribution within 95% probability. Salari et al. [17] applied this with a transformer model that predicted anatomical landmarks in magnetic resonance imaging (MRI) images.

The motivation behind using MCDO is its simplicity and applicability to any task performed with the use of a deep learning model. In particular, we exploit the ease of implementation and the flexibility of MCDO to combine it with a statistical test, resulting in a new method for uncertainty quantification (UQ) that was more appropriate for our application. While this is an established method, we see less instances of it for UQ in the literature; in most instances, the uncertainty is calculated via standard deviation or variance, but, in this work, we demonstrate that this is not always the best way to use the MCDO results. This is not the only method for uncertainty, for instance Bayesian neural

networks (BNNs) can also be used for UQ, but we save that approach for future work. The research questions we asked included the following: *how can MCDO be used with a deep learning model to learn from limited data and improve model performance?*

Unlike other approaches, we used a statistical test, the Kolmogorov–Smirnov (KS) test, to measure the uncertainty as a distribution shift of the MCDO prediction distribution. To the best of our knowledge, this approach has not been used for uncertainty quantification with MCDO, or deep learning in general. By relying on a statistical test that compares the prediction distribution with a known distribution from the validation data, we can account for uncertainty in OOD samples where predictions are not expected to be Gaussian distributions, e.g., for distributions of predictions skewed near zero or one. We considered how the model performed on simulated synthetic aperture radar (SAR) images of ship wakes on the ocean surface. To use SAR data with deep learning typically requires a large dataset for training, to account for the variations in SAR parameters and configurations that effect the final image [18]. SAR data, however, are time-intensive and expensive to collect, so large datasets of SAR images are not very common [18,19]. To address this, we generated images in previous work [20]; however, that approach used physics-based simulations, which are also time-intensive to generate. Therefore, our dataset has only 224 images. The ship wakes were augmented to rotate them to specific angles for training, as this is a way to add the variety expected in the real world, but with a small dataset size, rotation augmentations cannot cover all possible angles from 0 to 360, leaving gaps in the coverage. We used a simple CNN because the dataset was small and as we did not want it to overfit the data, because that could have been a source of brittleness when testing on the OOD samples. The main contributions of this work are as follows:

- We demonstrate a use case of MCDO for uncertainty quantification with SAR imagery using CNN models, including some models with feature-level fusion of multiple SAR bands.
- We apply the KS statistical test for quantifying the uncertainty of the MCDO model which, to the best of our knowledge, has not previously been used in this application rather than measures of variance or entropy, and demonstrate that this is a better approach for our application of detecting OOD samples.

This work characterizes uncertainties for image classification algorithms the authors previously developed [20]. Once uncertainties had been characterized on each individual band, we implemented feature-level data fusion with model inputs and assessed whether this data fusion approach could limit the uncertainty in OOD conditions. The scope and focus of this work did not include uncertainty characterization of the physics-based model that generated the data nor the input data for the algorithms, as our goal was to assess the changes in uncertainty when using data fusion. We chose to focus the study on a previously developed model architecture for binary classification of surface ship wakes and the corresponding feature-level fusion models. To minimize changes to the model architecture, we chose the dropout approach over other approaches commonly used for uncertainty quantification of machine learning models, such as BNNs [21–23]. While dropout variational inference is a practical method with reduced computing costs, we continue to investigate other approaches for uncertainty quantification.

2. Previous Work

The simulated SAR images were generated via a series of physics-based models that simulated a surface ship wake. There were three distinct domains and computational models used to produce the synthetic SAR imagery: namely, a hydrodynamic domain that represented a portion of the ocean, the surface of that domain to account for the roughness and surfactant redistribution, and the SAR sensor domain that modeled the radar return based on the surface features produced by the two proceeding hydrodynamic models. The hydrodynamic models that included parabolized (2D + t) Navier–Stokes and surfactant redistributions were implemented in OpenFOAM v2012 (See http://www.openfoam.com (accessed on 18 July 2022) for more information about OpenFOAM). Then, the radar images

were generated via a modified ERIM Ocean Model (EOM) [24], which ports data into Python 3.8 to generate the synthetic SAR images.

Figure 1 shows the full simulation pipeline. More details about the specifics of data generation can be found in [20,25,26].

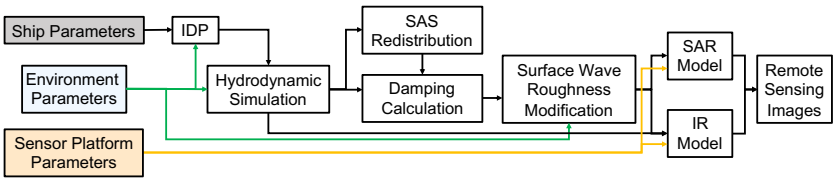

Figure 1. Flowchart outlining the simulation process that generated the simulated SAR data. Input parameters are on the left, green arrows indicate where the environmental parameters were injected into the pipeline, and the yellow arrows indicate where the sensor parameters were injected. IDP—initial data plane, SAS—surface active substance, IR—infrared [26]. Reprinted with permission from Ref. [26] 2023, Sobien.

A 2D + t computational domain was used for the hydrodynamic simulation, to increase the speed at which the simulations were performed. This was possible because the surface ship wakes of interest were far enough downstream from the ship that there was negligible impact from contributions in the streamwise direction. Periodic boundary conditions were used on the front and back faces of the 2D slice of ocean, in order to propagate the evolution of the wake in time. The front face was initialized on the initial data plane (IDP), which is a prescribed time downstream of the ship in which the streamwise variations are negligible and can either be extracted from a nearfield simulation of the ship or computed analytically [27]. The left and right sides were set to standard outlet boundary conditions with damping regions, to ensure there were no reflections off the side walls of the domain. The top and bottom boundaries were treated as free slip. A Boussinesq approximation accounted for the buoyant forces and a $k - \epsilon$ turbulence model included turbulent fluctuations in the flow [28]. The makeup of the ocean, involving the different stratified layers associated with temperature and salinity are beyond the scope of the work presented here, but details can be found in [29]. Once the 2D + t simulation had been completed, a reconstruction of surface was performed by stitching together the slices that were evolved in time. The reconstructed surface then served as the computational domain that redistributed the surfactant concentration and modified the surface wave roughness, both of which were then used in the SAR model. The SAR parameter list is shown in Table 1, which provides the sensor input parameters used to generate the synthetic images.

Table 1. Run matrix for SAR image generation.

Parameter	Possible Values
SAR band	C, S, X
Look angle	0°, 90°, 180°, 270°
Polarization	VV, HH
Inclination angle	30°, 40°, 50°, 60°

The different SAR configurations, documented in Table 1, impacted the resulting images, which were then used in the wake detection models. Since these configurations resulted in different images of the same scene, we could use them to study data fusion algorithms and determine if the aggregation of these images over the same scene resulted in a higher probability of detection of the present wake. We should also state that all of the hydrodynamic cases that were executed modeled a ship wake, but certain configurations of the SAR sensor were unable to pick up these wakes, those are labeled as 'no-wake' cases. Beyond the SAR configurations, the hydrodynamic parameters (and ship parameters) played a role in the generated wake images. Here, we maintained the same ship for all

cases, which was based on the ship used in [30]. The main hydrodynamic parameters that impacted the images generated in this study were the swell direction (head seas or following seas), swell height (0 m (calm seas), 0.5 m, 1.0 m, and 1.5 m), and stratification (yes or no).

3. Methodology

The goal of this experiment was to identify if data fusion machine learning models reduce uncertainty. The chosen classifier for the study used an MCDO technique. MCDO involves dropping nodes during training so the network can learn on a sparse representation of the data. The result is several output predictions for a single input, which can either be averaged or analyzed as a distribution.

To estimate the uncertainty, we utilized Monte Carlo integration of the model's likelihood $p(y|x, \Theta_t)$, where x is the model input and y is the target. The parameter (weights and biases) Θ_t is a sample from the approximate parametric distribution $q(\Theta_t|S)$, where S is the set of training samples, $S = \{(x_i, y_i)\}_{i=1}^{N}$, i is the sample index, N is the total number of samples, and t represents an individual stochastic forward pass (associated with an individual dropout model). The result is an approximation of the predictive distribution,

$$p(y|x) \overset{\text{MC}}{\approx} \frac{1}{T} \sum_{t=1}^{T} p(y|x, \Theta_t), \tag{1}$$

where T is the total number of stochastic forward passes, which we set as 100. We apply the dropout layer before the last fully connected layer of the proposed network architecture. For each neuron in this layer, we consider a Bernoulli random variable that takes a value of one with a fixed probability p_{MCDO}, which is common to all neurons. For each forward pass, if the corresponding Bernoulli variable takes the value one, the neuron is switched off, meaning its value is set to zero. The output of the Monte Carlo simulation includes the mean $\mu(x, \Theta_t)$ and the variance $\sigma^2(x, \Theta_t)$. By randomly switching off some of the neurons, we obtain different network configurations. Each forward pass produces a different output, and the multiple passes yield a distribution over the mean. The model uncertainty is quantified via analysis of the distribution of T stochastic forward passes. In [5], the authors also added the inverse model precision to the variance, but our results did not include this term. We acknowledge that the method is an approximation, but it provides a practical tool and useful insights to understand the effects of data transformation on model uncertainty.

3.1. Data Splits, Augmentations, and OOD Samples

The data were randomly split into 60/20/20 for training, validation, and testing, and we used the same images (i.e., the same set of environmental, ship, and SAR parameters, other the than band in Section 2) in each split for every band and model in the experiment, so there was consistency in comparing the results. The training and validation data were augmented via a random 45-degree interval rotation, θ between 0 and 315 degrees, so $\theta \in \{0, 45, 90, 135, 180, 225, 270, 315\}$. The original dataset consisted of SAR images with a surface ship wake generated via physics-based models, and the wake was horizontal right to left in every image of the original dataset [20]. Rotation augmentations add the variety that we expect to see in real-world data. Because the dataset size was small (224 images total with 134 images for training), rotation augmentations could not adequately cover all possible angles from 0 to 360, so there were gaps in the coverage, which was inevitable given the small size of the dataset.

During testing, each image was passed through the model multiple times to obtain a distribution for evaluation. One test set consisted of all the test images rotated to the same angle, so there was a test set for each of the eight angles listed above, as well as three out-of-domain (OOD) angles: $\{30, 60, 105\}$. For example, the baseline classifier trained on C-band images was then evaluated on the test images eleven times. For the first round, no augmentations were applied, for the second round, 30-degree rotations were applied, for the third, 45-degree rotations were applied, and so on until all angles had been used. This

gave us a clearer understanding of how the model performed inside and outside its training domain. The OOD angles (30, 60, 105) were in the coverage gaps between the in-domain angles, and while these did not represent a significant change in angle, the results below show that this was indeed a significant change for our models. The delineation of OOD and in-domain samples was only 15 degrees in some instances, but because the training dataset was so small, the gaps in coverage could have an impact on the model generalizability and performance on OOD samples.

3.2. Models

The models in this study were based on the CNN classifiers developed in our past work Higgins et al. [20] to test a foundational method for fusing C-, S-, and X-band SAR images to improve ship wake classification. SAR image data were fused using two methods: a feature-level approach, where multiple images were fed into a CNN classifier and features were extracted and fused at the same time; and a decision-level approach, where a pretrained classifier for each band was fused based on Bayesian inference of the models' past performances as priors [20]. The baseline classifier from the previous work was based on a simple three-convolutional-layer and two fully connected layer neural network. A small architecture was chosen to compliment the small dataset size and help prevent the model from overfitting our dataset. We used the baseline classifier and feature-level fusion algorithms from our previous work in this study.

All the deep learning models used were based on the same baseline CNN, comprising three convolutional block layers (with 10, 20, and 30 channels, respectively) and two fully connected linear layers (with 70 and 30 nodes, respectively). The output layer of the CNN was a single node for the predicted probability of the image containing a persistent wake. The convolutional block for each model comprised a 2D convolution with a kernel size of 3, a batch norm layer with no track running stats, the rectified linear unit (ReLU) activation function, and a max pooling layer of kernel size 2. The input to the network can be a single SAR-band image, (i.e., a single-channel image) or it can be concatenated multiple SAR-band images (i.e., a multi-channel image). The models with multi-band inputs use the CNN layers for feature extraction and fusion, as described in Higgins et al. [20].

In this study, we introduced MCDO into the fully connected layers of the baseline model using the PyTorch v0.17.1, with Cuda 12.1 dropout layer, with probability of 0.5 applied after each one. This randomly zeroed out half the parameters between the layers, forcing the model to generalize during training, but we also used this during inference to obtain multiple different outputs from the same input image. The objective of adding MCDO to this model, and the goal of this study, was to improve the performance of the model on OOD test samples.

During evaluation, the MCDO model inferenced on the same test image for 100 passes, providing a distribution of predicted probabilities per image. The objective of this study was to compare the MCDO classifier with the baseline classifier to see if the MCDO distribution of outputs could provide better performance, in conjunction with the feature data fusion, on the OOD images or at least indicate model uncertainty when the provided image was OOD and that the prediction should be regarded with skepticism.

Selecting Model Threshold

Before making predictions on the test data, we first determined the threshold that the predicted probabilities had to be above to classify that image as a wake. We automated the process to use the validation results to set the threshold separately for each model and band combination. We found the maximum predicted probability for the no-wake ground truth images and the minimum predicted probability for the wake ground truth images, then set the threshold (τ) at 25% of the distance, D between the max no-wake prediction ($\hat{y}_{nw,max}$) and min wake prediction ($\hat{y}_{w,min}$), where

$$D = \hat{y}_{w,min} - \hat{y}_{nw,max}. \quad (2)$$

The threshold τ can be calculated as

$$\tau = \hat{y}_{nw,max} + 0.25 * D, \tag{3}$$

and reduced to

$$\tau = 0.75\,\hat{y}_{nw,max} + 0.25\,\hat{y}_{w,min}. \tag{4}$$

We used the max no-wake prediction ($\hat{y}_{nw,max}$) and min wake prediction ($\hat{y}_{w,min}$) from the validation results of each trained model. This assumed the validation data represented the training data well (e.g., similar coverage of wake rotation angles), and that the inference data would match this coverage for in-domain samples. If the predictions on a validation set had $\hat{y}_{nw,max} = 0.2$ and $\hat{y}_{w,min} = 0.9$, then the threshold hold was calculated as $\tau = 0.75 * 0.2 + 0.25 * 0.9 = 0.375$. Any sample during inference that was predicted as wake with greater than 0.375 was labeled wake and any prediction less than that was labeled as non-wake.

3.3. Metrics

Three metrics were used for evaluation. We used the Matthews correlation coefficient (MCC) and average precision (AP) for the performance evaluation of the baseline and MCDO classifiers. The MCC is the measure of agreement between a model's class predictions and the ground truth. This metric was calculated using the MCC function from scikit-learn [31] and was defined as

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{5}$$

where TP is the count of true positives, TN is true negatives, FP is false positives, and FN is false negatives. We chose MCC because metrics like accuracy and F1 can misrepresent the results in an overoptimistic way, whereas MCC is more reliable, as it uses all four confusion matrix metrics (TP, FP, TN, FN) and can better handle imbalanced datasets [32]. The MCC has also been suggested as a standard metric for statistical and machine learning evaluation [33].

AP is a metric that evaluates prediction probabilities (the continuous output of models before thresholding) compared to the ground truth. This metric was calculated using the AP function from scikit-learn [31]. By varying the threshold and evaluating the precision and recall, the AP is a threshold-independent way of comparing different models,

$$AP = \sum_{n} (R_n - R_{n-1}) * P_n, \tag{6}$$

where precision, P, and recall, R, are calculated over a series of thresholds, n, (which the scikit-learn implementation performs automatically). This is the same as the area under the precision–recall curve. It provides a comprehensive analysis of theoretical performance—how well a model could perform if the threshold is tuned properly.

The third metric was the KS test, which calculates the max difference between two cumulative distribution functions (CDF). A CDF measures the proportion of a distribution that is equal to or less than the prediction probability, so the KS score is the max difference in proportion between two distributions at the same predicted probability value. We used the KS test to measure uncertainty as the distributional shift of the MCDO prediction distribution. This is achieved by comparing the MCDO distribution for a single image to the no-wake and wake distributions from the validation data. By relying on a statistical test that compared the inference distribution with a known distribution from the validation data, we could account for uncertainty in the OOD samples, where predictions are not expected to be Gaussian distributions. If the KS score is low (below 0.9 for our study), then it is considered in-domain, otherwise it is labeled as OOD.

$$KS = \max_{1 \le t \le T} |CDF_1(\hat{y}_t) - CDF_2(\hat{y}_t)|, \quad (7)$$

where, as above, t and $T = 100$ represent the individual MCDO model prediction and the total number of MCDO model predictions, respectively, CDF are the cumulative distribution functions for the validation results and a single test image distribution, respectively, and $\hat{y}_i$ is the predicted probability between 0 and 1 where the difference is measured. Figure 2 shows an example of a CDF, which measures the proportion (y-axis) of a distribution that is equal to or less than the prediction probability (x-axis) for the wake positive validation distribution (blue) compared to an in-domain wake image (orange) and an OOD wake image (green). The in-domain wake distribution had a KS score of 0.47 (meaning the max difference in proportion of the two distributions was 0.47) measured at the predicted probability value of 0.9998. The OOD wake distribution had a KS score of 1.0 measured at the predicted probability of 0.5940. The bi-directional arrows visually represent the measured KS score in Figure 2. The in- and out-of-domain results were from the same image, but the out-of-domain image was rotated 30-degrees.

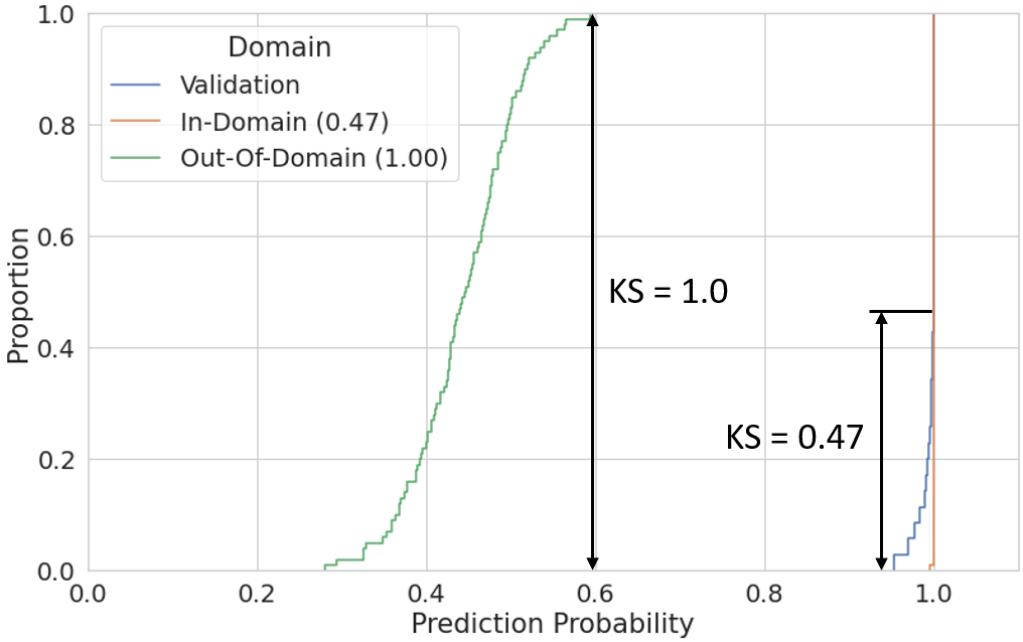

Figure 2. Example of two Kolmogorov–Smirnov (KS) test measurements relative to validation results for a wake positive case (blue). The plots are cumulative distribution functions (CDF), which measure the proportion (y-axis) of a distribution that is equal to or less than the prediction probability (x-axis). The in-domain wake distribution for a single image is shown in orange, with a measured KS of 0.47; and the out-of-domain (OOD) wake distribution for a single image is shown in green, with a measured KS of 1.0. The bi-directional arrows visually represent the measured KS score. The in- and out-of-domain results are from the same image, but the out-of-domain image has been rotated 30 degrees.

3.4. Experimental Setup

The SAR ship wake dataset was a binary classification problem, where the classes were either persistent ship wake present in the SAR return or no persistent wake present. The dataset has 224 images that were divided into a 60/20/20 train/val/test split (134/45/45 images), which were stratified by the target class. The dataset is imbalanced, with 32 no-wake images and 192 wake images. The MCC is better than accuracy or F1 score for imbalanced data and hence we used that metric for evaluation [32,33].

The training lasted 200 epochs, which is more than one might expect for such a small dataset, but the MCDO required additional training epochs for the model to converge. The batch size was 4 and the training was carried out on an NVIDIA T400, 4 GB GDDR6 GPU.

We used a stochastic gradient descent (SGD) optimizer with no scheduler, a learning rate of 0.001, a weight decay of 0.1, and a binary cross-entropy (BCE) loss function.

The training also included random image rotations in 45-degree intervals from [0, 315] for the in-domain angles. Model testing used the same parameters, where applicable, with the only exception being the image rotation angles fixed at a single angle. We repeated testing using every angle (in-domain and OOD) with the test set, one at a time. This repeated the same test set eleven times, but under the assumption that the rotation augmentation created a new independent and identically distributed (IID) sample image.

4. Results

4.1. Baseline and Monte Carlo Dropout Predictions

Augmenting the test data with rotations that were different from the training data shifted the resulting predictions for both the baseline and DO classifiers. Figure 3 shows strip plots of the baseline (top) and MCDO (bottom) classifiers, and each column represents a different SAR band. All of the in-domain angle results (see Section 3.1) are aggregated on the left-hand-side of each subplot (labeled "In") and all of the OOD angle results are aggregated on the right-hand side (labeled "Out"). The baseline classifier results (top) show the predicted probability per test image as a horizontal line, but the MCDO classifier (bottom) used the mean of the 100 passes predicted per image, so the number of results are the same for both.

Figure 3. The top row shows the baseline classifier results and the bottom row has the MCDO classifier results. Each column of subplots is for a different SAR band, meaning a model trained and evaluated on the corresponding band. The results in each subplot are grouped on the left-hand-side for in-domain angles, while results on the right-hand-side are OOD angles. Colors indicate the target or ground truth of the image, either orange for wake or blue for no-wake.

For all bands and models, we can see that the in-domain results tightly grouped near 1 for the wake ground truth images (orange) and 0 for the no-wake ground truth images (blue). There was a clear shift in predicted probabilities for both the baseline and MCDO classifier for the OOD images, as the wake predictions spanned the entire range of probabilities, indicating the difficulty the models had with the OOD images with a wake present. On the other hand, it seems that the no-wake images had relatively similar distributions of predicted probabilities whether in-domain or OOD. The exception to these findings were the S-band results, where

the no-wake results had probabilities that spanned much higher than all the other models, and the S-band OOD wake results did not drop as low as the other OOD results.

From Figure 3, we can see there was a clear distinction between the in-domain and OOD results for the test sets, but little difference between the baseline and MCDO classifier results on the OOD samples. Simply using the MCDO method without further interpreting or analyzing the results did not alter the predictions when the test samples shifted from in-domain to OOD angles. We need to apply statistical methods to determine if the probability distribution from the MCDO classifier for a test image aligned with in-domain results or not, which we carry out in the following section.

4.2. Kolmogorov–Smirnov Test

The validation results for the appropriate model, band, and target were used as the reference for the KS test in this section. We used matching validation results as the reference CDF of the KS test for each model, band, and target. For example, all the C-band MCDO classifier validation results for ground truth wake images were used as the reference CDF, then each distribution of 100 passes from the MCDO classifier test predictions was compared to the reference CDF. Figure 4 shows the C-band test predictions for a single in-domain set (0-degree rotation on the left-hand-side) and a single set (30-degree rotation on the right-hand-side); where the no-wake ground truth is in blue, the wake ground truth is in orange, and the reference validation CDF curves are in black. The no-wake reference CDF curve, generated from in-domain samples, was near 0 to 0.1, while the wake reference CDF curve was near 0.95 to 1. Each blue or orange line represents the outputs for one single image passing through the MCDO classifier 100 times. For the in-domain results, we can see the MCDO curves stayed very near the reference CDF curves, indicating those results were in-domain. In the right-hand subplot, we can see the OOD results for the wake ground truth images spread across the entire range of probabilities, with some close to one of the reference CDF curves, but most far from both.

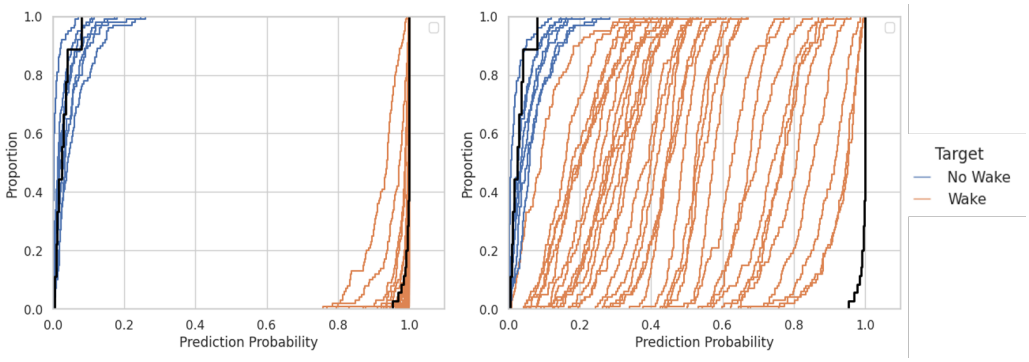

Figure 4. C-band test results for in-domain (0-degree rotation on left-hand-side) and OOD (30-degree rotation on right-hand-side) predictions for no-wake ground truth (blue), wake ground truth (orange), and the reference validation CDF curves (black). The reference curve near 0 is for the no-wake images, while the reference curve near 1 is for the wake images. Each blue or orange line represents the distribution of outputs for a single image passing through the MCDO classifier 100 times.

We set the KS threshold at 0.9, so if a MCDO Classifier's prediction KS score was below that value, it was predicted as in-domain to the reference CDF curve it was closest to. If the KS score was greater than 0.9 from both reference CDF curves, then we classified the prediction as OOD wake, because we know, from Figure 3, the OOD predicted probabilities of no-wake images varied little from the in-domain predicted probabilities (as we elaborate in Section 5). We performed this KS thresholding and show the KS labeled results in Figure 5. Similarly to in Figure 3, we show the mean probability for the MCDO classifier predictions, but use colors based on the KS thresholding label, where blue is no-wake, orange is wake, and green is OOD wake.

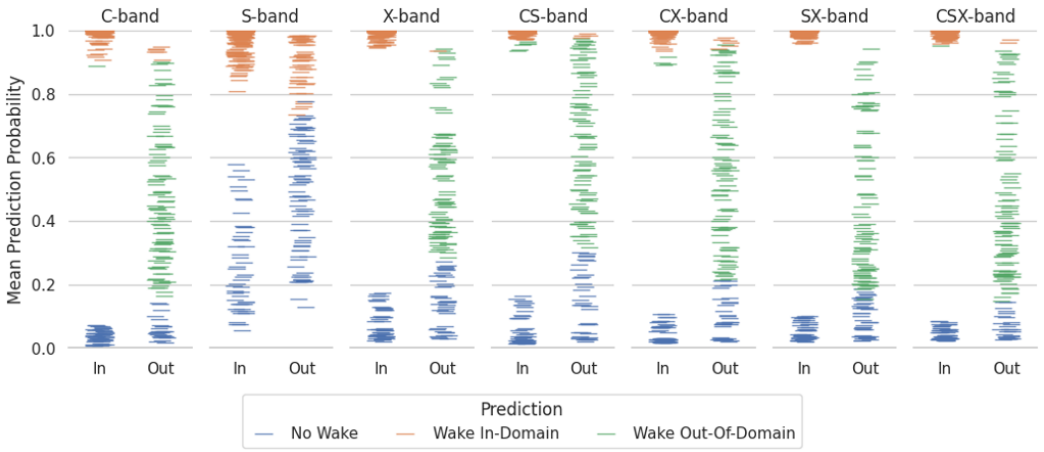

Figure 5. Strip plot showing the mean probability for the MCDO classifier prediction probabilities of each image. Color labels are based on the prediction from the KS value, where no-wake (blue) are CDF curves that are within a KS distance of 0.9 from the respective band's no-wake validation data, wake (orange) are curves within KS 0.9 of the wake validation data, and wake out-of-domain are those curves that are greater than KS 0.9 from either validation curve.

Most in-domain predictions were labeled properly, but some lower predicted probability wake cases were incorrectly labeled OOD (but were still labeled wake cases). There were several OOD wake images labeled incorrectly as no-wake as they approached the no-wake ground truth probability values (refer to Figure 3, which shows the data with ground truth labels). The S-band again stood out from the rest, and in fact there was only one OOD image that was labeled correctly, showing that the methods struggled with the classification task in S-band images. By using the distribution of the MCDO classifier predictions, we could better determine which samples may be OOD compared to using a single value like the mean prediction or standard deviation. By comparing the predictions to the distribution of validation results, we avoided the 'thresholding problem' for predicted probabilities, meaning that it is easier to measure a distribution shift than set a threshold for expected values when you have not seen the OOD data before and do not know what to expect regarding their predicted probabilities. We evaluate the KS labels in the next section with the MCC and AP metrics.

4.3. MCC Results

We calculated MCC for the baseline classifier, using the mean value of the predicted probabilities with the MCDO classifier (using the same thresholds as the baseline classifier, see Section 3.2), and using the KS labels (Figure 5) with the MCDO Classifier. Figure 6 shows the MCC results split by in-domain angles (left), OOD angles (middle), and both domains together (right). The baseline classifier results are in blue, the mean probability of the MCDO classifier is in orange, and the KS predictions from the MCDO classifier are in green. All models and bands performed well with in-domain data, but there was a drop in performance with OOD data. The results for both domains reflect the same relative performance of the models from the OOD data.

The KS predictions performed best over all the bands, with a significant improvement for the C-, CSX-, and SX-band models, while the improvements were marginal for the other models. The only band that performed worse was the S-band. Using the mean probability of the MCDO classifier results did not improve the performance for most bands; there were only marginal improvements for the SX- and CSX-bands. From these results, we can see that additional assessment (e.g., statistical KS test) is required to take advantage of the distribution from the MCDO classifier. Adding this step to the uncertainty quantification of outputs for the MCDO classifier helped make the model predictions more robust for the OOD samples, even though it was trained on a small training set. The individual results from Figure 6 are tabulated in Table 2.

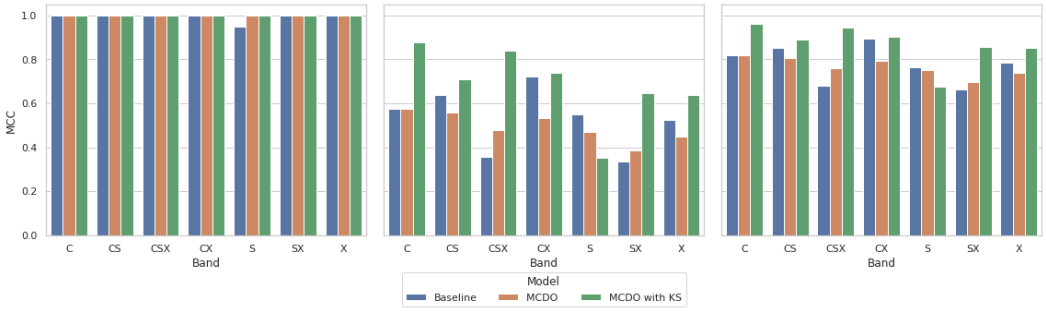

Figure 6. MCC results split by in-domain angles (**left**), OOD angles (**middle**), and all the image domains together (**right**). The baseline classifier results are in blue, the mean predicted probability of the MCDO classifier is in orange, and the KS predictions from the MCDO distributions are in green.

Table 2. Matthew's correlation coefficient (MCC) results for all models, bands, and datasets.

	In-Domain			Out-of-Domain			All-Domains		
Band	**Baseline**	**MCDO**	**KS**	**Baseline**	**MCDO**	**KS**	**Baseline**	**MCDO**	**KS**
C	1.000	1.000	1.000	0.575	0.575	0.878	0.818	0.818	**0.960**
S	0.951	1.000	1.000	0.548	0.469	0.351	**0.763**	0.753	0.675
X	1.000	1.000	1.000	0.523	0.448	0.637	0.787	0.740	**0.852**
CS	1.000	1.000	1.000	0.637	0.557	0.710	0.852	0.807	**0.889**
CX	1.000	1.000	1.000	0.724	0.531	0.738	0.895	0.792	**0.902**
SX	1.000	1.000	1.000	0.334	0.385	0.648	0.664	0.698	**0.858**
CSX	1.000	1.000	1.000	0.356	0.476	0.838	0.679	0.758	**0.944**

4.4. AP Results

We calculated AP results on the baseline and the mean probability MCDO classifier results. We had to omit the KS labels, because they are not a continuous probability but a categorical label. The AP scores for all bands and models were between 0.998 and 1.0. They were effectively the same and told us little about the performance between the models. The AP metric, however, uses a range of thresholds during calculation, so the results indicated that for each model, whether predicting in-domain or OOD, there is a theoretical threshold that gives ideal or near-ideal performance. However, given that the nature of OOD data means we have never seen or trained on it before, there is no *a priori* way to adjust the threshold for OOD data and still call the results OOD.

The AP metric indicates that the predictions for OOD wake samples shifted below what we expected and set our threshold at for the baseline and MCDO classifiers, because if we could change the threshold our precision and recall would increase. We do need to caveat that the AP metric does suffer on some imbalanced datasets and can report optimistic results [33]; however, because it is threshold-agnostic, it can help compare different models without needing to optimize the threshold. The similarity in performance between the baseline and the mean predictions of the MCDO classifier indicates that there was not much change in performance between the two when not using the outputs of the MCDO without a statistical analysis.

5. Discussion

5.1. Comparison of Baseline and Monte Carlo Dropout for OOD Data

Both the baseline and the MCDO classifiers were impacted by test images rotated to OOD angles. The main benefit of the MCDO model is that we can assess a distribution of outputs to understand if an image is in fact OOD, rather than just relying on a single datapoint. In theory, the MCDO distribution should act as an ensemble and we could use the mean to determine the predicted class, but the mean MCDO results in Figure 6 and in Table 2 show that this was not the case for our experiments. The percent change in MCC of MCDO using the mean probability (Table 3) was +11.62% for the CSX-band but

−11.58% for the CX-band from the baseline, with a −1.05% change averaged across all the bands. Using the entire distribution with the KS test, however, provided a significant boost in the performance of the model. The percent change in MCC from the baseline to the MCDO with KS test was +39.07% for the CSX-band, and while the rest were positive, only the S-band had a negative change of −11.51%. The averaged percent change for the KS test was +12.48% across all the bands, demonstrating that making full use of the MCDO distribution with the KS test benefited the model performance over taking the mean of the dropout distribution.

Table 3. Percentage change in classifier MCC performance from the baseline using the all-domain results from Table 2.

Band	MCDO	KS
C	0.00%	17.32%
S	−1.29%	−11.51%
X	−5.95%	8.27%
CS	−5.23%	4.35%
CX	−11.58%	0.74%
SX	5.10%	29.10%
CSX	11.62%	39.07%
Average	−1.05%	12.48%

The results for the AP metric were less informative, because the AP was at or near 1 for every model and data subset. AP is calculated from the area under the precision–recall curve, so a high AP means the precision remains high for all or most thresholds. Given this, the threshold for the MCC could likely be recalculated to improve the results, but that would require an *a priori* understanding of model performance when it comes to OOD data, which would break the assumption that the test data are truly OOD. We prefer to use the assumption that the inference data are truly unknown to the model and the user, and therefore we could not reconfigure the threshold for any unknown domain changes.

5.2. Limitations

There are some limitations in generalizing this approach to other scenarios. Most notably, the no-wake images experienced no distribution shift when the images were augmented via rotation, because they did not have any important features for the model to learn and were mostly background ocean noise. Coupled with our problem being binary classification, it was simple to infer that any images with a shifted distribution were the wake OOD images. If, for example, noise was instead applied as the augmentation, the no-wake OOD images could have shifted distributions higher and overlapped with the wake OOD distributions. This is merely a hypothetical, and future work would need to explore this approach with other augmentations.

We arbitrarily chose the KS threshold for in- vs. out-of-domain, but generalizing this approach for other problems or datasets would require understanding what the distribution of in-domain data typically looks like, in order to properly choose a threshold.

5.3. Why Not Other Quantitative Metrics?

In Section 4.2, we showed the promise of using KS distance to improve the classification of OOD images. It might, therefore, be natural to wonder whether other, simpler statistical metrics might provide similar performance improvements. This section is meant to highlight why we did not choose a metric like standard deviation or variance of the outputs as our uncertainty metric. Greater variation in the predicted probabilities of the OOD images was expected, and we saw that, but the amount of variation seen did not drastically differ between the in-domain and the no-wake probabilities. Figure 7 shows the kernel density estimation of the standard deviation (STD) for the C-band results of the MCDO classifier. There is some overlap, but judging any one sample of STD would be

difficult. If we look at all the in-domain and OOD deviations (Figure 8), we can see that there was a lot of overlap between the OOD wake and the no-wake results (especially for S- and CS-bands), which would make determining which distribution the results came from difficult. We could couple this with the mean of the MCDO outputs, to see if they are high for a wake case or low for a no-wake case. This brings us to approximating the results as a normal distribution, despite the fact that we know that the in-domain results were not normally distributed (see Figure 3), they were heavily skewed near or at 0 or 1. Therefore, we felt that relying on statistical tests of the distribution was a better method to make full use of the information contained therein.

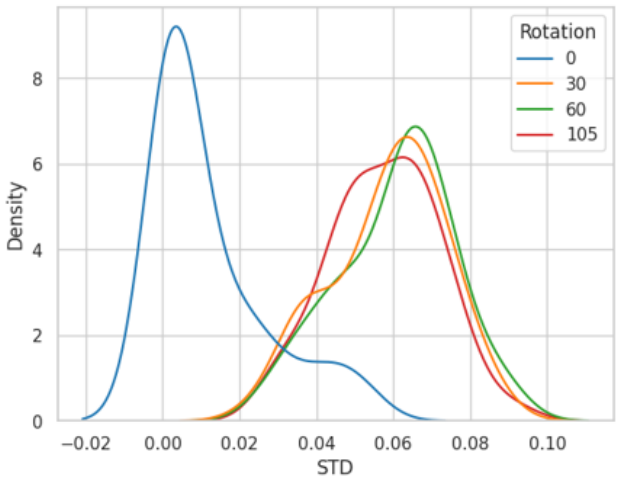

Figure 7. Kernel density estimations for the distribution of C-band standard deviations (STD) for the MCDO Classifier. The 0-degree rotation (blue) is in-domain. The 30-, 60-, and 105-degree rotated images (orange, green, and red, respectively) are OOD.

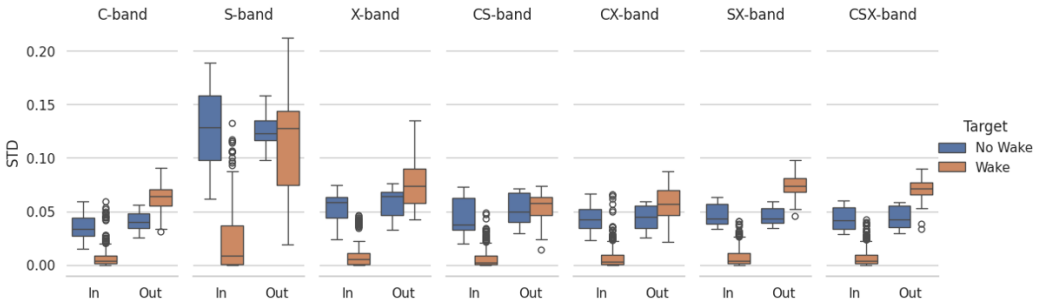

Figure 8. Standard deviation of the MCDO classifier results. Each column shows a different SAR band. The results in each subplot are grouped on the left-hand-side for in-domain angles, while the results on the right-hand-side are OOD angles. Color indicates the target or ground truth of the image, either orange for wake or blue for no-wake. The standard deviations of in-domain no-wake images and OOD images often overlap, making it hard to use standard deviation to distinguish between in- and out-of-domain images. Note that the circles are outlier data points within that given distribution.

Our motivation was to assess MCDO for a CNN model trained with limited data, our small SAR ship wake dataset. Most of the existing literature relied on using the mean for MCDO predictions and variance or standard deviation for measuring uncertainty; however, in our study, we show that the approach is less useful than using a statistical distribution test. Mean MCDO predictions failed to generalize to OOD data and even performed worse than the baseline model without dropout (Table 3). When using standard deviation for uncertainty, we showed that changes between in-domain and OOD uncertainty in most cases were slight and in some cases non-existent (S- and CS-bands in Figure 8). Instead, we used the KS test, which compared the inference distribution of MCDO predictions with a

known distribution from the validation data, allowing us to account for uncertainty in the OOD samples, where predictions are not expected to be Gaussian distributions.

6. Conclusions and Future Work

We used MCDO in a deep learning model trained on limited data to help assess the epistemic uncertainty and improve model performance on OOD test samples. We looked at how the model performed on simulated SAR images of ship wakes on the ocean surface. The model was trained with 134 images with wakes rotated to specific angles and tested with OOD samples, which were rotated to different angles. We used a simple CNN because the dataset is small and we did not want it to overfit the data, because that could be a source of brittleness when testing on OOD samples.

We investigated the classifier performance on OOD rotated images during testing, with the expectation that this would degrade the model performance. This enabled us to investigate the probability distributions around model predictions and determine if the addition of a MCDO approach increased the model robustness, ultimately leading to increased performance for OOD test/inference data. The MCDO classifier demonstrated an improvement in model performance for OOD classification when the entire distribution of model predictions was compared to the validation data for UQ, rather than using an ensembling approach. We used a Kolmogorov–Smirnov test to measure the uncertainty and determine if the predicted probabilities were in-domain or out-of-domain, which enabled us to make further improvements to the OOD performance. We also found that the average precision metric did not provide adequate detail about model performance to make a meaningful comparison. We finished the discussion by outlining some limitations with the current approach, mainly that rotated no-wake images looked identical to non-augmented images, meaning that no-wake images always appeared in-domain and our method only needed to identify an image as OOD to know it contained a wake. This, coupled with the simplicity of the binary classification task and our synthetic ship wake data, allows our assumptions to hold, but as problems become more complex, with either more classes or data with additional features, our assumptions may need to be revisited and revised. Applying the KS test had, to the best of our knowledge, not previously been used for UQ of a MCDO model. We demonstrated that, rather than measures of variance or entropy, the KS score for UQ is a better approach in our application for detecting OOD samples.

One potential future line of inquiry would be to apply Bayesian neural networks to the wake detection problem. In this case, one would hope to see that the Bayesian model's predictions would be more consistent (e.g., showing a lower standard deviation) when they were more accurate. For example, one might expect the model performance to deteriorate as imagery deviated from the training data (e.g., for larger rotations or different image augmentations or combinations of multiple augmentations), and that the model predictions might also become more uncertain in these regimes. If such a result was confirmed, then model consistency could be used as a surrogate for accuracy, enabling detection of out-of-distribution imagery.

Author Contributions: M.C.: conceptualization, software, writing—original draft preparation. D.S.: conceptualization, methodology, software, data curation, validation, analysis, investigation, visualization, writing—original draft preparation. R.G.: analysis, writing—original draft preparation. W.C.: analysis, writing—original draft preparation. J.K.: conceptualization, writing—review and editing. J.A.K.: funding acquisition, supervision, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: Funding for this work was provided by Department of Navy award N00174-22-1-0028 issued by the Office of Naval Research.

Data Availability Statement: The data presented in this study are openly available on GitHub at https://github.com/dssobien/wake_data_augmentation (accessed on 6 May 2022).

Acknowledgments: The authors acknowledge both the Virginia Tech Advanced Research Computing (ARC) and DoD High Performance Computing Modernization Program (HPCMP) for support and maintenance of computational resources that enabled this effort.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243–297. [CrossRef]
2. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
3. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
4. Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **2015**, *521*, 452–459. [CrossRef]
5. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1050–1059. ISSN: 1938-7228.
6. Diederik, P.K.; Tim Salimans, M.W. Variational Dropout and the Local Reparameterization Trick. *arXiv* **2015**, arXiv:1506.02557.
7. Gal, Y.; Hron, J.; Kendall, A. Concrete Dropout. *arXiv* **2017**, arXiv:1705.07832.
8. Gal, Y.; Ghahramani, Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *arXiv* **2016**, arXiv:1512.05287.
9. Angela, F.; Edouard Grave, A.J. Reducing Transformer Depth on Demand with Structured Dropout. *arXiv* **2019**, arXiv:1909.11556.
10. Wu, Z.; Wu, L.; Meng, Q.; Xia, Y.; Xie, S.; Qin, T.; Dai, X.; Liu, T.Y. UniDrop: A Simple yet Effective Technique to Improve Transformer without Extra Cost. *arXiv* **2021**, arXiv:2104.04946.
11. Abdar, M.; Fahami, M.A.; Chakrabarti, S.; Khosravi, A.; Pławiak, P.; Acharya, U.R.; Tadeusiewicz, R.; Nahavandi, S. BARF: A new direct and cross-based binary residual feature fusion with uncertainty-aware module for medical image classification. *Inf. Sci.* **2021**, *577*, 353–378. [CrossRef]
12. Du, L.; Wang, W.; Pu, J.; Zhao, Z. Quantifying Uncertainty in Potato Leaf Disease Detection: A Comparative Study of Deep Learning Models Using Monte Carlo Dropout. In Proceedings of the International Conference on Internet of Things, Communication and Intelligent Technology, Xuzhou, China, 22–24 September 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 522–530.
13. Caldeira, J.; Nord, B. Deeply uncertain: Comparing methods of uncertainty quantification in deep learning algorithms. *Mach. Learn. Sci. Technol.* **2020**, *2*, 015002. [CrossRef]
14. Habibpour, M.; Gharoun, H.; Mehdipour, M.; Tajally, A.; Asgharnezhad, H.; Shamsi, A.; Khosravi, A.; Nahavandi, S. Uncertainty-aware credit card fraud detection using deep learning. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106248. [CrossRef]
15. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
16. Abdar, M.; Samami, M.; Mahmoodabad, S.D.; Doan, T.; Mazoure, B.; Hashemifesharaki, R.; Liu, L.; Khosravi, A.; Acharya, U.R.; Makarenkov, V.; et al. Uncertainty quantification in skin cancer classification using three-way decision-based Bayesian deep learning. *Comput. Biol. Med.* **2021**, *135*, 104418. [CrossRef]
17. Salari, S.; Rasoulian, A.; Battie, M.; Fortin, M.; Rivaz, H.; Xiao, Y. Uncertainty-aware transformer model for anatomical landmark detection in paraspinal muscle MRIs. In Proceedings of the Medical Imaging 2023: Image Processing, San Diego, CA, USA, 19–24 February 2023; Volume 12464, pp. 246–252.
18. Lewis, B.; Liu, J.; Wong, A. Generative adversarial networks for SAR image realism. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXV, Orlando, FL, USA, 15–19 April 2018; Volume 10647, pp. 37–47.
19. Lewis, B.; DeGuchy, O.; Sebastian, J.; Kaminski, J. Realistic SAR data augmentation using machine learning techniques. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXVI, Baltimore, MD, USA, 14–18 April 2019; Volume 10987, pp. 12–28.
20. Higgins, E.; Sobien, D.; Freeman, L.; Pitt, J.S. Data Fusion for Combining Information for Disparate Data Sources for Maritime Remote Sensing. In Proceedings of the AIAA Scitech 2021 Forum, Online, 19–21 January 2021; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2021. [CrossRef]
21. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer: New York, NY, USA, 1996.
22. Lampinen, J.; Vehtari, A. Bayesian approach for neural networks—Review and case studies. *Neural Netw.* **2001**, *14*, 257–274. [CrossRef] [PubMed]
23. Goan, E.; Fookes, C. Bayesian Neural Networks: An Introduction and Survey. In *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*; Mengersen, K.L., Pudlo, P., Robert, C.P., Eds.; Lecture Notes in Mathematics; Springer International Publishing: Cham, Switzerland, 2020; pp. 45–87. [CrossRef]
24. Lyzenga, D.R.; Bennett, J.R. Full-Spectrum modeling of Synthetic Aperture Radar Internal Wave Signatures. *J. Geophys. Res.* **1988**, *93*, 12345–12354. [CrossRef]

25. Sobien, D.; Higgins, E.; Krometis, J.; Kauffman, J.; Freeman, L. Improving Deep Learning for Maritime Remote Sensing through Data Augmentation and Latent Space. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 665–687. [CrossRef]
26. Sobien, D.; Kauffman, J.A.; Higgins, E.; Freeman, L.; Pitt, J.S. Evaluation of Machine-Learning Data Fusion Classifier Performance for Ship-Wake Detection with Modified Data Sets. In Proceedings of the AIAA SCITECH 2023 Forum, National Harbor, MD, 23–27 January 2023; p. 0195.
27. Miner, E.W.; Ramberg, S.E.; Swean, T.F., Jr. *A Method for Approximating the Initial Data Plane for Surface Ship Wake Simulations*; Technical Report; Maritime Technical Information Facility: Washington, DC, USA, 1988.
28. Rodi, W. Examples of calculation methods for flow and mixing in stratified fluids. *J. Geophys. Res.* **1987**, *92*, 5305. [CrossRef]
29. Higgins, E.T. Machine Learning and Data Fusion of Simulated Remote Sensing Data. Ph.D. Thesis, Virginia Tech, Blacksburg, VA, USA, 2023.
30. Sussman, M.; Dommermuth, D.G. The numerical simulation of ship waves using Cartesian grid methods. *arXiv* **2014**, arXiv:1410.1952.
31. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
32. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 1–13. [CrossRef] [PubMed]
33. Chicco, D.; Tötsch, N.; Jurman, G. The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Min.* **2021**, *14*, 1–22. [CrossRef] [PubMed]