Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

ROBUST DECISION-MAKING IN THE INTERNET OF BATTLEFIELD THINGS USING BAYESIAN NEURAL NETWORKS

Adam D. Cobb

CCDC DEVCOM Army Research Laboratory
U.S. Army Futures Command
2800 Powder Mill Road
Adelphi, MD 20783, USA

Brian A. Jalaian

CCDC DEVCOM Army Research Laboratory
U.S. Army Futures Command
2800 Powder Mill Road
Adelphi, MD 20783, USA

Nathaniel D. Bastian

Army Cyber Institute
United States Military Academy
2101 New South Post Road
West Point, NY 10996, USA

Stephen Russell

CCDC DEVCOM Army Research Laboratory
U.S. Army Futures Command
2800 Powder Mill Road
Adelphi, MD 20783, USA

ABSTRACT

The Internet of Battlefield Things (IoBT) is a dynamically composed network of intelligent sensors and actuators that operate as a command and control, communications, computers, and intelligence complex-system with the aim to enable multi-domain operations. The use of artificial intelligence can help transform the IoBT data into actionable insight to create information and decision advantage on the battlefield. In this work, we focus on how accounting for uncertainty in IoBT systems can result in more robust and safer systems. Human trust in these systems requires the ability to understand and interpret how machines make decisions. Most real-world applications currently use deterministic machine learning techniques that cannot incorporate uncertainty. In this work, we focus on the machine learning task of classifying vehicles from their audio recordings, comparing deterministic convolutional neural networks (CNNs) with Bayesian CNNs to show that correctly estimating the uncertainty can help lead to robust decision-making in IoBT.

1 INTRODUCTION

The Internet of Battlefield Things (IoBT) is comprised of a network of sensors, wearables, and devices that use cloud and edge computing to create a cohesive fighting force. The IoBT consists both the network itself, as well as the autonomous devices (the "things") that leverage inexpensive compute, networking, sensing, and actuation capabilities to sense the physical world and act on it. As such, the IoBT is a dynamically composed network of intelligent sensors and actuators that operate as a command and control, communications, computers, and intelligence complex-system. As such, the characteristics of an IoBT network include devices that: 1) vary in heterogeneity and quantity; 2) are individually and collectively intelligent; 3) operate over multiple time-scales and multiple domains; 4) have ambiguous ownership, resulting in potential security risks.

The IoBT is a crucial enabler for the warfighting concept of multi-domain operations (MDO), where there is a degraded and cluttered battlefield with numerous actors to include humans, robots, unmanned vehicles, and signals. The use of artificial intelligence (AI) can help deal with the overwhelming amounts

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

of IoBT data to increase the speed of action, improve data analytics for better decision-making, and adapt to uncertain and unknown situations (Lott et al. 2020). Among the inherent risks with using AI, there is often a lack of understanding of how machine learning (ML) performance impacts safety. Further, there is uncertainty of information obtained from the IoBT, which affects the decision-making process in MDO; humans rely on underlying reasons for uncertainty in making decisions (Raglin et al. 2020). While ML has been shown to outperform human-level performance (Krizhevsky et al. 2012) and has become an attractive choice for use in decision-making tasks, it is also the case that ML methods can be poorly calibrated and often result in overconfident outputs (Gal 2016; Guo et al. 2017).

The application of AI in the IoBT–MDO context includes leveraging ML techniques to execute tasks across different domains and data modalities including images, video, audio, unstructured/structured text, and signal processing (e.g., acoustic, radar, communications, infrared). Vehicles are critical elements of modern warfare used by military forces for both combat and transportation applications across the battlefield. Acoustic data, to include military vehicle acoustics, is an important modality for any information-gathering system operating on the battlefield and in the real-world. Acoustic data, such as vehicle audio recordings, can be used for harmonic feature extraction of ground vehicles for acoustic classification, identification, direction of arrival estimation and beamforming. Most devices come with built-in microphones and there is no limiting requirement to be directed at the object of interest in the same way as a camera.

While many ML approaches can be applied successfully to audio, the majority of research has been focused on computer-vision applications. In particular, deep learning methods, such as convolutional neural networks (CNNs), have become the model architecture of choice when it comes to pattern recognition in image data. While CNNs have seen great success in computer-vision applications (Krizhevsky et al. 2012), it has been shown that applying CNNs to spectrograms also leads to favorable results (Kiskin et al. 2020; Kiskin et al. 2020). The fact that CNNs can also be applied to acoustic data supports their use in real-world systems where multiple data streams are available. However, one limiting factor, also relevant to vision applications, is a need for large data sets with correct labels. In practice the data that is available is likely to be imbalanced and sparse. The other limiting factor of CNNs is that they can be poorly calibrated and therefore overly confident in their outputs (e.g. see Jha et al. (2019)). When we combine this limitation of poor calibration with the potential to over-fit on a small imbalanced data set, then we see where the challenge may lie in applying CNNs in real-world applications.

In this work, we seek to enable robust decision-making in the IoBT context via the use of Bayesian neural networks to classify vehicles from their acoustic microphone recordings while accounting for uncertainty. Specifically, we propose to overcome the limitations of applying CNNs to acoustic data by replacing deterministic CNNs with Bayesian CNNs (BCNNs) (MacKay 1992; Neal 1995). BCNNs offer a probabilistic alternative to CNNs by specifying prior distributions over the weights. The motivation for working with BCNNs comes from the improved uncertainty estimates when making predictions. This ability to provide more reliable uncertainty estimates is the reason why using BNNs can lead to more robust, safer decision-making in MDO in the highly complex IoBT with multiple data streams.

In particular, we compare deterministic CNNs with BCNNs for the task of classifying vehicles from stationary microphone recordings. We use the audio recordings from the Acoustic-seismic Classification Identification Data Set (Hurd and Pham 2002), where we are faced with the challenge of identifying nine classes of vehicles in a highly imbalanced noisy data set. We show that deterministic neural networks are highly confident over their outputs and that this overconfidence can cause undesirable outputs for specific tasks such as identifying the least common class in the data. As a result of the poor performance of standard CNNs, we propose to employ BCNNs as a more reliable approach for operating with imbalanced data in a potentially safety-critical scenario of identifying rare classes correctly. This scenario of detecting rare classes that are seldom seen in the data has clear relevance to military and national security applications where examples of adversarial behaviour may be sparse and difficult to include in data sets.

2 RELATED WORK

While most ML researchers and practitioners are solely focused on the "accuracy" of ML models, it has become more apparent that accuracy is not enough for high-performing ML techniques (Vadera et al. 2020). In recent years, there has been a surge in research towards deploying uncertainty quantification (UQ) in deep learning to solve real-world applications in science and engineering. A comprehensive overview of recent UQ techniques in the context of deep learning within several applications is presented in the recent Abdar et al. (2020) survey article.

As deep learning models continue to improve their predictive accuracy across many application domains, significant issues remain with respect to other highly important aspects of performance including their ability to robustly quantify uncertainty (Guo et al. 2017) and their ability to provide robust predictions in the presence of adversarial manipulations (Goodfellow et al. 2015) and out-of-distribution examples (Ovadia et al. 2019). Approximate Bayesian inference methods (Neal 1996; Jaakkola and Jordan 2000) hold considerable promise for addressing such issues, and recent advances have significantly improved the feasibility of deploying approximate Bayesian inference methods to increasingly larger deep learning models (Welling and Teh 2011; Blundell et al. 2015; Gal and Ghahramani 2016; Zhang et al. 2020).

There is an imminent need for uncertainty management and quantification for IoBT, which enables risk-aware and, thus, robust decision-making and control for subsequent intelligent systems and/or humans within the IoBT environment (Jalaian and Russell 2019). The focus on the management of UQ in large-scale intelligent data fusion and processing systems such as IoBT is also studied in Marlin et al. (2020), where they make recommendations for future areas of work.

3 METHODOLOGY

We employ Bayesian neural networks for the classification of vehicles from their audio recordings. Unlike deterministic neural networks, Bayesian neural networks require learning a distribution over the neural network weights. This is an important area of research and there exists a large range of inference schemes varying from cheaper variational approximations such as Monte Carlo dropout (Gal and Ghahramani 2016) to purely sampling directly from the posterior via Markov chain Monte Carlo (MCMC) approaches such as Hamiltonian Monte Carlo (Neal 1995). We focus on the latter approach and use the inference scheme of Cobb and Jalaian (2020) to learn the posterior distribution over the model parameters. This inference scheme employs Hamiltonian Monte Carlo (HMC) in a way that makes it feasible to run on a single GPU.

3.1 Hamiltonian Monte Carlo

HMC is a gradient-based MCMC sampler that employs Hamiltonian dynamics to traverse the parameter space of models. We can use HMC to overcome the challenge of performing inference in highly complex Bayesian models, such as Bayesian neural networks. To materialize samples, we start from the unnormalized log posterior, which is defined via the likelihood $p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega})$ and the prior $p(\boldsymbol{\omega})$ as follows:

$$p(\boldsymbol{\omega}|\mathbf{Y},\mathbf{X}) \propto p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega})\, p(\boldsymbol{\omega}).$$

The likelihood is a function of the parameters, $\boldsymbol{\omega} \in \mathbb{R}^D$, otherwise known as weights of the neural network model. The prior encodes any assumptions over the weights such as the magnitude. Finally, to perform HMC, the Bayesian model must be transformed into a Hamiltonian system by introducing a momentum variable $\mathbf{p} \in \mathbb{R}^D$, such that we now have a log joint distribution, $\log[p(\boldsymbol{\omega},\mathbf{p})] = \log[p(\boldsymbol{\omega}|\mathbf{Y},\mathbf{X})\, p(\mathbf{p})]$, that is proportional to the Hamiltonian, $H(\boldsymbol{\omega},\mathbf{p})$. If we let $p(\mathbf{p}) = \mathcal{N}(\mathbf{p}|\mathbf{0},\mathbf{M})$, where the covariance $\mathbf{M}$ denotes the mass matrix, our Hamiltonian can then be written as:

$$H(\boldsymbol{\omega},\mathbf{p}) = \underbrace{-\log[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega})\, p(\boldsymbol{\omega})]}_{\substack{\text{Potential Energy} \\ U(\boldsymbol{\omega})}} + \underbrace{{}^1\!/_2\, \mathbf{p}^\top \mathbf{M}^{-1}\mathbf{p}}_{\substack{\text{Quadratic Kinetic Energy} \\ K(\mathbf{p})}}. \tag{1}$$

This form consists of a quadratic kinetic energy term derived from the log probability distribution of a Gaussian and a potential energy term, which is our original Bayesian model. Then we can sample from this model by simulating the dynamics of the Hamiltonian system via the Stormer–Verlet (or leapfrog) integration scheme. We leave details of this scheme to Neal et al. (2011), which provides a general overview, and Cobb and Jalaian (2020), which gives the specific symmetric scheme used in this work.

3.2 Sampling and Evaluation

We compare a deterministic neural network with a Bayesian neural network. To learn the weights of the deterministic model, we use stochastic gradient descent to optimize over the weights. At the end of this process, we are left with a single model with parameters, $\boldsymbol{\omega}$, such that for a given input $\mathbf{x}^*$ the network makes a single prediction $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}^*; \boldsymbol{\omega})$. For the Bayesian CNN, as noted above, we materialize samples from the posterior distribution resulting in a set of samples $\{\boldsymbol{\omega}_s\}_{s=1}^{S} \sim p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$ such that the predictive distribution can be approximated via multiple network draws for each test image $\hat{\mathbf{y}}_s = \mathbf{f}(\mathbf{x}^*; \boldsymbol{\omega}_s)$. In comparing the deterministic and Bayesian models, we show how the ability to capture uncertainty results in more robust decision-making for IoBT.

One way to evaluate the performance of the models is to report the accuracy of the models, whereby the prediction for the Bayesian model is computed by averaging over the samples and selecting the class with the highest probability (i.e., $c = \text{argmax}_c \frac{1}{S}\sum_s \hat{\mathbf{y}}_s$, where c is the vehicle class). The other way is to evaluate the performance according to a cost function, $\mathscr{C}(h, y)$, that is relevant to the problem. This is where h corresponds to the final selection and y corresponds to the prediction of the network. We can then calculate the conditional risk for each decision, h,

$$\begin{aligned}\mathscr{R}(\mathbf{h}|\mathbf{x}^*) &= \frac{1}{S}\sum_s \mathscr{C}(\mathbf{h}, \hat{\mathbf{y}}_s) \\ &= \left[\frac{1}{S}\sum_s \hat{\mathbf{y}}_s\right]^\top \mathbf{C}\ \mathbf{h}, \end{aligned} \tag{2}$$

and select the action, $\mathbf{h}^*$, that minimizes the conditional risk. Note that we have simplified the cost function to take the form of a cost matrix here. As an example, setting the cost matrix to a constant minus the identity matrix and selecting the minimum risk decision, would result in maximizing the accuracy. Other cost matrices, such as the one we require here, penalize errors on certain classes more relative to others.

Once we have minimized the conditional risk to select action, $\mathbf{h}^*$, we must then evaluate how the expected minimum risk actions perform in the presence of labelled data. Therefore, we need to introduce the decision cost, which is the actual cost accrued once we know the true decision that ought to have been made. The decision cost, $\mathscr{C}_T$, is the actual cost when the Bayes optimal decision, $\mathbf{h}^*$, is applied (corresponding to the minimization of the conditional risk). To calculate this cost, we must be careful how we use the cost function. Unlike before where we calculated the expected cost by integrating over the posterior predictive distribution, we now have two pairs of vectors $\mathbf{h}^*$ and $\mathbf{h}_{\text{True}}$ that correspond to the decisions taken and the true labels.

Therefore, we must be cautious in how to combine $\mathbf{h}^*$, $\mathbf{h}_{\text{True}}$, and $\mathbf{C}$, such that we calculate the true cost correctly. We can do a small thought experiment, whereby $\mathbf{h}^*$ dictates that we should decide on class 'F', when the right decision is $\mathbf{h}_{\text{True}}$ = 'G'. Given the setup of our experimentation, an error on class 'G' should accrue a cost of 1.00 (refer to Table 1). The decision cost must be calculated by replacing the prediction in Equation (2) with $\mathbf{h}^*$ and the decision with $\mathbf{h}_{\text{True}}$:

$$\mathscr{C}_T(\mathbf{h}_{\text{True}}, \mathbf{h}^*) = \mathbf{h}^{*\ \top}\mathbf{C}\ \mathbf{h}_{\text{True}}. \tag{3}$$

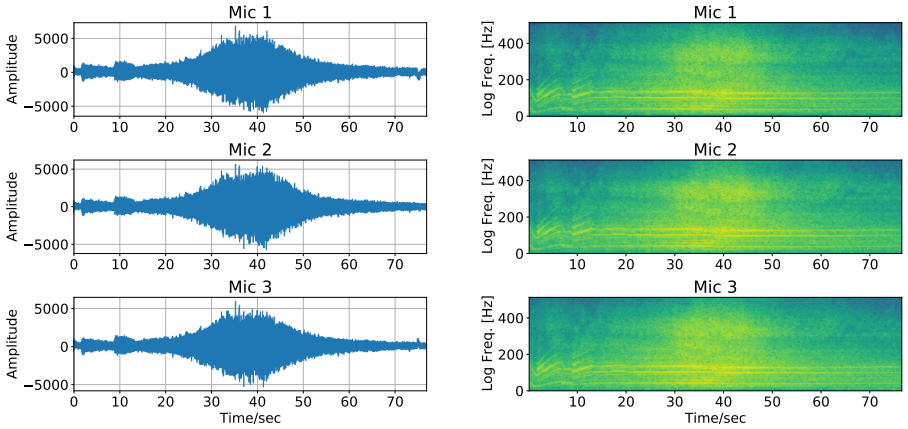
This result is the true cost accrued by the taking the decision $\mathbf{h}^*$.

4 COMPUTATIONAL EXPERIMENTATION

Our overall objective is to classify vehicles from their acoustic microphone recordings. However, the data set is highly imbalanced and certain classes of vehicles will carry differing levels of cost for incorrect classifications. We use the Acoustic-seismic Classification Identification Data Set (ACIDS) data set that consists of 223 audio recordings. ACIDS was originally used by Hurd and Pham (2002) for the harmonic feature extraction of ground vehicles for acoustic classification, identification, direction of arrival estimation and beamforming, but here we focus on acoustic classification. There are nine classes of vehicles, where each vehicle is recorded via a triangular array of three microphones.[1] As each vehicle drives past the array of microphones the audio recordings are taken. Figure 1a shows the raw time-series recordings of the three microphones for a single vehicle driving past. The increasing then decreasing amplitude corresponds to how close the vehicle is to the microphone, where the closest point of approach can be seen from the maximum amplitude.

For the purposes of our experimentation, we build seven train-validation splits where we set aside 40% of the recordings for validation and use the remaining for training. We then transform each recording (for both the training and validation) into the frequency domain using a short-time Fourier transform (STFT), with the Scikit-learn default settings of `scipy.signal.spectrogram` (Pedregosa et al. 2011). Figure 1b shows the STFT of the three time-series from Figure 1a.

Although 223 recordings may not sound sufficiently large for training and validation, the median elapsed time for each recording is 139 seconds and the total elapsed time of all the recordings in the entire data set is just over 10 hours. Finally, we divide all the spectrograms into equal 129×150 arrays that correspond to approximately 10 seconds of recordings from the triangular array of three microphones. Figure 1 shows the process of transforming from the three microphone recordings to the spectrogram, and Figure 2 includes a single example of the final array to be passed into the ML model. The spectrograms from all three microphones (aligned in time) are concatenated into one image, which is then passed into the ML model. The total 129×150 array has a resolution of 4.0 Hz in the vertical axis and a resolution of 0.22 seconds in the horizontal axis.



(a) Raw Microphone Recording (b) Short Time Fourier Transform

Figure 1: Representations of the Data of a Single Recording from the Triangular Array of Three Microphones

The data set is highly imbalanced, where certain classes appear less frequently in the data set. The histogram in Figure 3 shows the total data set after the pre-processing of both the training and validation data into the 129×150 arrays. The resulting histogram shows the large discrepancy in frequency between the class labels. For example, vehicle class 'G' only makes up 1.5% of the data, compared to vehicle class

[1] Audio was recorded at a sampling rate of 1025.641 Hz.

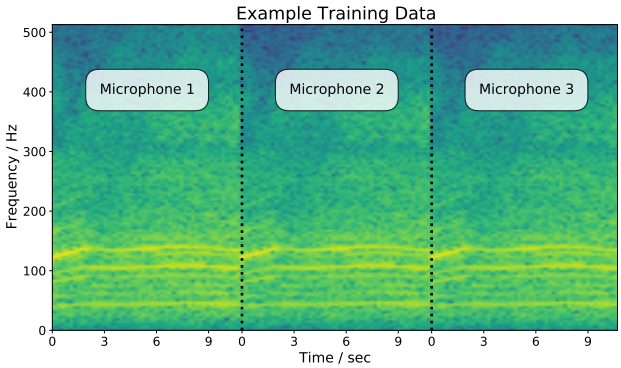


Figure 2: An Example of a Single Input Datum

'A' which makes up 32.2% of the data. Such differences in the prevalence of different classes can cause challenges in learning models for classification. One solution to these problems, which we propose herein, is to use Bayesian neural networks to account for these types of data imbalance.

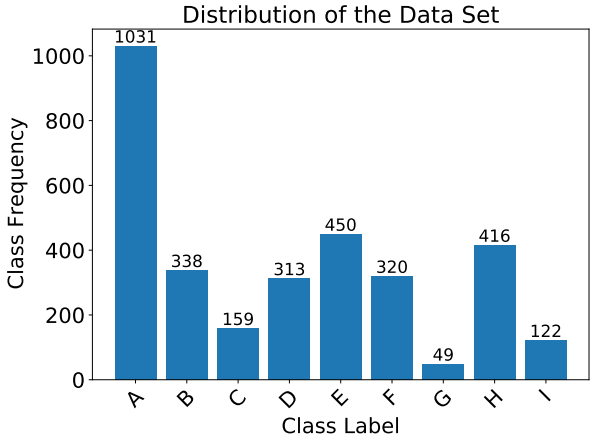


Figure 3: Distribution of the Imbalanced Data Set

4.1 The Neural Network Architecture

We use a CNN model architecture as it has been shown that applying CNNs to spectrograms also leads to favorable results (Kiskin et al. 2020; Kiskin et al. 2020). A CNN replaces the traditional fully-connected layers of a dense neural network with convolutional filters. These filters are learned in the same way that the weights and biases are learned for fully-connected neural networks. The structure of a CNN facilitates the extraction of shift invariant features from incoming images. We expect to automatically learn the patterns in the spectrogram that correspond to different vehicle types. Our CNN architecture consists of four convolutional layers with max-pooling, followed by a fully-connected last layer (see Figure 4). We use Scaled Exponential Linear Units (SELUs) as the activation function (Klambauer et al. 2017), which we find yields an improvement over commonly-used alternatives such as rectified linear units.

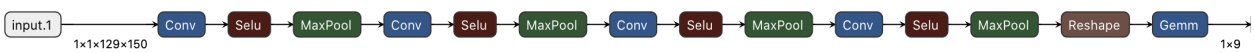


Figure 4: CNN Model Architecture

Both the deterministic and Bayesian approaches have the same CNN model architecture. We used the backpropagation algorithm (Rumelhart et al. 1986) implemented in PyTorch (Paszke et al. 2017) to learn the weights of the deterministic CNN. As discussed in Section 3, we used HMC to learn the posterior over the weights given the data for the Bayesian CNN and use the same hyperparameters as in Cobb and Jalaian (2020). Other common choices for approximating the posterior are by performing Monte Carlo sampling (Neal 1995; Welling and Teh 2011) and using variational inference (Graves 2011; Blundell et al. 2015; Gal and Ghahramani 2016)).

4.2 The Decision-making Task: Avoiding Catastrophic Failure

We now outline both the objective of our task, as well as how to design a cost function to achieve this goal. The objective is to correctly classify all vehicles from their acoustic recordings. However, as part of the task, vehicles in class 'G' carry a higher level of threat to our classification scheme compared to the other classes. As a result, erroneous decisions on class 'G' incur a larger penalty than for the other vehicle classes. If we refer back to Figure 3, we can see that there are very few instances of class 'G' in the data. The rarity of class 'G' makes the overall objective especially challenging given that this least frequent class is also the one that requires the most caution when making decisions. This scenario is not unusual in practice. It is often the case in safety-critical applications that the classes we are most concerned about appear the least often. Furthermore, it might be infeasible to collect more instances of these classes due to their rarity or cost. In such a scenario, one might be tempted to up-weight instances of these rare cases in the data. In some cases this can lead to poor behaviour when data is mislabelled or noisy in the input space (Cobb 2020, Chapter 4). If we can achieve the goal of closely approximating the posterior distribution, then data that appears less frequently ought to result in highly uncertain predictions if the input data are not easily distinguishable from other classes.

As mentioned in Section 3.2, we can use an appropriate cost matrix, $\mathbf{C}$, to encode user preferences. To encode the preferences outlined in the previous paragraph, we can penalize errors on class 'G' more than for other classes. The cost matrix is shown via the Table 1, where errors for class 'G' are assigned a penalty of 1.0, whereas errors for the other classes are assigned a penalty of 0.01. Each column corresponds to the decision, $\mathbf{h}$, and each row the prediction, $\hat{\mathbf{y}}$.

Table 1: Cost Matrix for the Decision-making Process

		Decision								
	Cost	A	B	C	D	E	F	G	H	I
Prediction	A	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	B	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	C	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01
	D	0.01	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01
	E	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.01	0.01
	F	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.01
	G	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00
	H	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.01
	I	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00

As a simple example, suppose the samples from the posterior predictive distribution resulted in the mean $\left[\frac{1}{S}\sum_s \hat{\mathbf{y}}_s\right]^\top$ being a flat distribution corresponding to a vector of nine elements each with a value of $1/9$. This prediction corresponds to maximum predictive entropy or, in other words, our model is predicting that any class is equally likely. We will see that this prediction of the environment is actually informative for decision-making. If we follow Equation (2), we can calculate the risk of deciding on class

'G', $\mathscr{R}(\mathbf{h} = \text{'G'}|\mathbf{x}^*)$, versus selecting any of the other classes, $\mathscr{R}(\mathbf{h} = \neg\text{'G'}|\mathbf{x}^*)$:

$$\mathscr{R}(\mathbf{h} = \text{'G'}|\mathbf{x}^*) = \frac{1}{9}(0.01 \times 8 + 0.00 \times 1) = 0.009$$
$$\mathscr{R}(\mathbf{h} = \neg\text{'G'}|\mathbf{x}^*) = \frac{1}{9}(1.0 \times 1 + 0.01 \times 7 + 0.00 \times 1) = 0.119.$$

Therefore, the optimal decision, which minimizes the expected cost, is achieved by selecting $\mathbf{h}$ that minimizes the conditional risk. For the cost matrix in Table 1, the very fact that some probability mass of the posterior predictive distribution falls on class 'G' is enough to result in setting $\mathbf{h} = \text{'G'}$.

4.3 Results

We compare a deterministic CNN with a Bayesian CNN to demonstrate the importance of inferring a distribution over the weights rather than just optimizing for a single network parameterization. Since the output of a CNN is normalized such that the elements of each vector $\hat{\mathbf{y}}$ sums up to one, we can directly compare the Bayesian CNN with the deterministic CNN by replacing the summation in Equation (2) with the single vector point estimate of the distribution, i.e., $\mathscr{R}(\mathbf{h}|\mathbf{x}^*) = \hat{\mathbf{y}}^\top \mathbf{C}\mathbf{h}$.

We directly calculate the true cost in Equation (3) for both the prediction, $\mathbf{h} = \hat{\mathbf{y}}$, and the Bayes optimal decision that minimizes the conditional-risk, $\mathbf{h} = \mathbf{h}^*$. The purpose of comparing the two is to show that taking into account the user preferences of this task results in a lower actual cost. The comparison also highlights that using the output of a neural network may not always be the best choice if the outputs are being used in downstream tasks. In the context of IoBT, military decision-makers often have different risk attitudes (or thresholds) associated with their planned or real-time decision options, which is why modeling uncertainty of information obtained from the IoBT is so critical as it affects decision-making in MDO.

In addition to the true cost over the validation data, we also compare our model's accuracy performance for both $\mathbf{h} = \hat{\mathbf{y}}$ and $\mathbf{h} = \mathbf{h}^*$. We note that when one is interested in achieving the highest class accuracy, the effective cost matrix is a constant minus the identity matrix. In other words, the cost in making an error is the same for all classes, and the reward for a correct classification is equal for every class. We implicitly use this cost matrix when directly assigning classes according to the output of the model. Therefore, we expect the accuracy performance to be the highest for $\mathbf{h} = \hat{\mathbf{y}}$, where the cost is aligned with the prediction.

We display the results in Table 2, where they are evaluated over seven cross-validation splits of the data. The table displays both the accuracy performance and the cost over the validation sets. There are two main comparisons that can be made from this table. The first comparison is between the performance of the predictions in the first two columns versus the optimal decisions in the last two columns. In comparing these two, we see that the cost for both the deterministic CNN and the Bayesian CNN drop from 24.8 and 24.3 to 10.0 and 5.0, respectively, in moving from the predictions to the decisions. This result highlights how the cost matrix is incorporated into the decision-making process and leads to a lower cost than just going with the highest probability output from the network.

Table 2: Predictive and Decision-making Performance

MODELS	PRED. $\mathbf{h} = \hat{\mathbf{y}}$		DEC. $\mathbf{h} = \mathbf{h}^*$	
	ACC.	COST	ACC.	COST
DETERMINISTIC CNN	80.3 ± 3.2	24.8 ± 7.8	76.5 ± 3.8	10.0 ± 6.9
BAYESIAN CNN	84.1 ± 2.7	24.3 ± 8.9	73.0 ± 6.0	5.0 ± 1.4

The second key comparison that can be made is the advantage of the Bayesian CNN over the deterministic CNN. The better the approximation of the posterior predictive distribution, the more robust the decision, and this result is precisely what is seen in Table 2. In the last column of the table, the cost over the

validation data is half that of the deterministic CNN. It is also interesting that the accuracy performance over the decisions is better for the deterministic CNN, despite its worse cost. This difference highlights that accuracy can be highly skewed by more populous classes and provide an incorrect proxy for performance. The error bounds represent standard deviation intervals across the validation splits. From Table 2 we find that Bayesian neural networks ought to be considered and applied if we are to capture task-specific preferences. It is not sufficient to just select the maximal probability outputs from the model, if they do not align with user-preferences. Further, a model with a better approximation of the posterior predictive distribution will result in more robust decision-making, especially for highly skewed cost matrices and imbalanced data sets.

We further highlight the importance of having a well-calibrated predictive uncertainty by displaying the specific performance over class 'G' in Table 3. The clear difference between the predictions and the decisions is even more evident than in Table 2. The accuracy of the prediction is poor for both models, with values of 14.7% and 17.3%. However, after minimizing the conditional-risk, the accuracy increases to 74.8% and 94.4%. This result can also be seen via the significant drop in cost from 22.3 for both models to costs of 6.9 and 1.3 for the deterministic and Bayesian CNNs, respectively. What makes this result especially interesting is that these cross-validation splits vary from having as little as 12 training examples in the training data to having 39 out of a total 49 examples. Therefore, such differences in the frequency of class 'G' ought to lead to higher standard deviations in the values of Table 3 compared to Table 2. However, we see that the performance of the Bayesian model is consistent and better overall than the deterministic model when used in conjunction with the cost matrix. The Bayesian CNN demonstrates its superiority over the deterministic CNN by achieving a low mean cost and with a low standard deviation. This consistent performance of the Bayesian CNN demonstrates how better UQ leads to robust decision-making.

Table 3: Predictive and Decision-making Performance for Class 'G'

MODELS	PRED. $\mathbf{h} = \hat{\mathbf{y}}$		DEC. $\mathbf{h} = \mathbf{h}^*$	
	ACC.	COST	ACC.	COST
DETERMINISTIC CNN	14.7 ± 10.4	22.3 ± 7.8	74.8 ± 20.4	6.9 ± 7.0
BAYESIAN CNN	17.3 ± 15.6	22.3 ± 9.1	94.4 ± 8.3	1.3 ± 2.1

As depicted in Table 3, the Bayesian CNN has well-calibrated uncertainties and does a better job at estimating the expected risks of the different decisions. The resulting cost is lower for the Bayesian CNN when compared to the deterministic CNN. Also, notice how naively using the predictive output of the networks does equally poorly (see the two left-hand side columns).

5 CONCLUSION

In this work, we introduced the benefits of using Bayesian neural networks to reduce uncertainty when employing ML techniques in safety-critical applications, such as IoBT. As a result, better calibrated uncertainty estimates allow users to estimate the cost of possible decisions. Building a system that relies on Bayesian inference techniques provides advantages over systems with end-to-end controllers. Clear preferences can be encoded into the decision-making process that allow for more understandable actions. The separation between the inference component and decision-making component enables the model to concentrate on pattern recognition tasks. We demonstrated how one might use such a methodology in practice with a highly imbalanced and challenging acoustic data set. For our task, the minority class in the data set was set to have the highest penalty for incorrect decisions. We then observed how the model that better approximated the uncertainty, resulted in more robust and safer decision-making for the task.

One of the challenges in the design of a robust decision-making system is determining the preferences of a decision-maker and how to incorporate preferences into the training of the ML model. In non-Bayesian

paradigms, models are typically trained by directly minimizing the empirical risk. The advantage of Bayesian methodology is that a model can be trained without the need to hand-craft loss functions, to which the subjectivity of the relative preferences are kept away from the ML engineer and left to the domain expert. Instead, inference over the model parameters can be kept as objective as possible, while a decision-maker calibrates the cost matrix according to personal preferences. Of course, this leaves many challenges open such as how to encode domain-specific preferences into a cost function and, ultimately, which model is best at approximating the posterior predictive distribution of the data.

As future work, we aim to investigate other approaches that have been used in the literature to tackle the problem of making safe, robust decisions; these include optimization over different user risk preferences by incorporating fuzzy set theory (Ekin et al. 2016) or using strict robust optimization (Bastian et al. 2020).

ACKNOWLEDGEMENTS

This work was funded by the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 and Interagency Agreement No. USMA21050. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Army Research Laboratory, the United States Military Academy, the United States Army, the Department of Defense, or the United States Government.

REFERENCES

Abdar, M., F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya et al. 2020. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". *arXiv preprint arXiv:2011.06225*.

Bastian, N. D., B. J. Lunday, C. B. Fisher, and A. O. Hall. 2020. "Models and methods for workforce planning under uncertainty: Optimizing US Army cyber branch readiness and manning". *Omega* 92:102171.

Blundell, C., J. Cornebise, K. Kavukcuoglu, and D. Wierstra. 2015. "Weight uncertainty in neural networks". In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, 1613–1622. JMLR. org.

Cobb, A. D. 2020. *The practicalities of scaling Bayesian neural networks to real-world applications*. Ph. D. thesis, University of Oxford.

Cobb, A. D., and B. Jalaian. 2020. "Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting". *arXiv preprint arXiv:2010.06772*.

Ekin, T., O. Kocadagli, N. D. Bastian, L. V. Fulton, and P. M. Griffin. 2016. "Fuzzy decision making in health systems: a resource allocation model". *EURO Journal on Decision Processes* 4(3-4):245–267.

Gal, Y. 2016. *Uncertainty in deep learning*. Ph. D. thesis, University of Cambridge.

Gal, Y., and Z. Ghahramani. 2016. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning". In *International Conference on Machine Learning*, 1050–1059.

Goodfellow, I. J., J. Shlens, and C. Szegedy. 2015. "Explaining and Harnessing Adversarial Examples". In *ICLR*.

Graves, A. 2011. "Practical variational inference for neural networks". In *Advances in Neural Information Processing Systems*, 2348–2356.

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger. 2017. "On calibration of modern neural networks". In *International Conference on Machine Learning*, 1321–1330. PMLR.

Hurd, H., and T. Pham. 2002. "Target association using harmonic frequency tracks". In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, Volume 2, 860–864. IEEE.

Jaakkola, T. S., and M. I. Jordan. 2000. "Bayesian parameter estimation via variational methods". *Statistics and Computing* 10(1):25–37.

Jalaian, B., and S. Russell. 2019. "Chapter 2 - Uncertainty Quantification in Internet of Battlefield Things". In *Artificial Intelligence for the Internet of Everything*, edited by W. Lawless, R. Mittu, D. Sofge, I. S. Moskowitz, and S. Russell, 19–45. Academic Press.

Jha, S., S. Raj, S. Fernandes, S. K. Jha, S. Jha, B. Jalaian, G. Verma, and A. Swami. 2019. "Attribution-based confidence metric for deep neural networks".

Kiskin, I., A. D. Cobb, L. Wang, and S. Roberts. 2020. "HumBug Zooniverse: a crowd-sourced acoustic mosquito dataset". In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 916–920. IEEE.

Kiskin, I., D. Zilli, Y. Li, M. Sinka, K. Willis, and S. Roberts. 2020. "Bioacoustic detection with wavelet-conditioned convolutional neural networks". *Neural Computing and Applications* 32(4):915–927.

Klambauer, G., T. Unterthiner, A. Mayr, and S. Hochreiter. 2017. "Self-normalizing neural networks". In *Advances in neural information processing systems*, 971–980.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "Imagenet classification with deep convolutional neural networks". In *Advances in Neural Information Processing Systems*, 1097–1105.

Lott, D. A., A. Raglin, and S. Metu. 2020. "On the use of operations research for decision making with uncertainty for IoT devices in battlefield situations: simulations and outcomes". In *Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations*, Volume 11426, 1142609. International Society for Optics and Photonics.

MacKay, D. J. 1992. "A practical Bayesian framework for backpropagation networks". *Neural Computation* 4(3):448–472.

Marlin, B. M., T. Abdelzaher, G. Ciocarlie, A. D. Cobb, M. Dennison, B. Jalaian, L. Kaplan, T. Raber, A. Raglin, P. K. Sharma, M. Srivastava, T. Trout, M. P. Vadera, and M. Wigness. 2020. "On Uncertainty and Robustness in Large-Scale Intelligent Data Fusion Systems". In *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*, 82–91.

Neal, R. M. 1995. *Bayesian Learning for Neural Networks*. Ph. D. thesis, University of Toronto.

Neal, R. M. 1996. *Bayesian Learning for Neural Networks*. Springer-Verlag.

Neal, R. M. et al. 2011. "MCMC using Hamiltonian dynamics". *Handbook of Markov chain Monte Carlo* 2(11):2.

Ovadia, Y., E. Fertig, J. Ren, Z. Nado, D. Sculley, J. Nowozin, Sebastian Dillon, B. Lakshminarayanan, and J. Snoek. 2019. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In *NeurIPS*, 13969–13980.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. 2017. "Automatic Differentiation in PyTorch". In *NIPS Autodiff Workshop*.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. 2011. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12(85):2825–2830.

Raglin, A., M. Dennison, S. Metu, T. Trout, and D. James. 2020. "Decision making with uncertainty in immersive systems". In *Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations*, Volume 11426, 114260L. International Society for Optics and Photonics.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. "Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1". Chapter Learning Internal Representations by Error Propagation, 318–362. Cambridge, MA, USA: MIT Press.

Vadera, M. P., A. D. Cobb, B. Jalaian, and B. M. Marlin. 2020. "URSABench: Comprehensive Benchmarking of Approximate Bayesian Inference Methods for Deep Neural Networks". *arXiv preprint arXiv:2007.04466*.

Welling, M., and Y. W. Teh. 2011. "Bayesian learning via stochastic gradient Langevin dynamics". In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 681–688.

Zhang, R., C. Li, J. Zhang, C. Chen, and A. G. Wilson. 2020. "Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning". In *ICLR*.

AUTHOR BIOGRAPHIES

ADAM COBB served as an ORAU Postdoctoral Research Fellow at the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory when completing this work. He currently works as an Advanced Computer Scientist at SRI International. He holds a Ph.D. degree in Autonomous Intelligent Machines and Systems from the University of Oxford, where his dissertation received a commendation from the Department of Engineering Science. His website is https://adamcobb.github.io/ and his email is cobb.derek.adam@gmail.com.

BRIAN JALAIAN is Senior AI Research Scientist at the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory, AI Test Tool Lead at the DoD Joint Artificial Intelligence Center (JAIC), and an Adjunct Assistant Professor in the Electrical and Computer Engineering Department at Virginia Tech. He holds a Ph.D. degree in Electrical Engineering from Virginia Tech, and M.S. degrees in Electrical Engineering (Network and Communication Systems) and Industrial System Engineering (Operations Research) from Virginia Tech, respectively. His research interests are the broad area of safe, robust, and resilient AI. His email address is brian.a.jalaian.civ@mail.mil. His website is https://www.brianjalaian.com/.

NATHANIEL BASTIAN is Chief Data Scientist and Senior Research Scientist at the Army Cyber Institute (ACI) at West Point, also serving as Assistant Professor of Operations Research and Data Science at the United States Military Academy (USMA). He leads the ACI's Data Science Research Team and oversees the ACI's Intelligent Cyber-Systems and Analytics Research Laboratory. He holds a Ph.D. degree in Industrial Engineering and Operations Research from the Pennsylvania State University, a M.Eng. degree in Industrial Engineering from Penn State, a M.S. degree in Econometrics and Operations Research from Maastricht University, and a B.S. degree in Engineering Management (Electrical Engineering) with Honors from USMA.

His email address is nathaniel.bastian@westpoint.edu. His website is https://cyber.army.mil/Research/Research-Team/Bastian/.

STEPHEN RUSSELL is the Information Sciences Division Chief at the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory. He leads a research program on the Internet of Battlefield Things, which is focused on multiple challenges to incorporating Internet of Things concepts and capabilities within the battlefield environment. He holds a B.S. degree in Computer Science along with M.S. and Ph.D. degrees in Information Systems from University of Maryland. He has published over 100 papers in the primary research areas of decision support systems, machine learning, systems architectures, and intelligent systems. His email address is stephen.russell15.civ@mail.mil.