Towards a Real-Time In-Flight Ice Detection System via Computational Aeroacoustics and Bayesian Neural Networks

Beckett Y. Zhou* and Nicolas R. Gauger†
Chair for Scientific Computing, TU Kaiserslautern
Bldg 34, Paul-Ehrlich-Strasse, 67663 Kaiserslautern, Germany

Myles Morelli‡ and Alberto Guardone§
Department of Aerospace Sciences and Technologies, Politecnico di Milano
Building B12 Campus Bovisa, Via La Masa, 34 20156 Milano, Italy

Jeremiah Hauth¶ and Xun Huan‖
Department of Mechanical Engineering, University of Michigan
1231 Beal Ave, Ann Arbor, MI 48109-2133, USA

In-flight rotor icing presents a serious problem in the operation of rotorcraft in cold climates, as complex ice shapes can significantly degrade the aerodynamic performance and handling characteristics of rotorcraft. Reliable real-time detection of ice formation is a critical enabling technology in improving rotorcraft safety. In this work, we propose a novel approach towards developing a real-time in-flight ice detection system using computational aeroacoustics and Bayesian neural networks. In particular, an icing simulation code based on a fully unsteady collection efficiency approach is coupled with the aeroacoustic solver available in the open-source software suite SU2, in order to compute far-field noise signatures corresponding to discrete iced rotor blades in various icing conditions. Additionally, Bayesian neural networks are constructed from the dataset thus generated to enable rapid predictions together with uncertainty information, of aerodynamic performance indicators from acoustic signal, as a first step in developing an in-flight ice detection and warning system.

I. Introduction

Rotorcraft possess the unique ability to vertically take-off and land, allowing them to operate in many challenging flight conditions where more conventional aircraft cannot. Their operational flight path however lends itself to low-level altitudes, and many missions are often at the boundary of the flight envelope. This is accompanied by additional challenges, one of which involves operations in harsh cold environments, such as during search and rescue missions. Flight operations under such situations involve the possibility of experiencing dangerous icing conditions where the presence of ice may drastically alter the aircraft's performance and stability.[1,2] An inferred climatology of icing conditions from Bernstein et al.[3,4] in Canada, Continental United States, Europe, and Asia has helped map the frequency of icing events, and illustrated that in winter months icing events occur at relatively low-level altitudes. Unlike fixed-wing aircraft, rotorcraft are unable to escape icing conditions via climbing to high enough altitudes where the presence of super-cooled water droplets decreases. In conjunction, in situ observations from reconnaissance aircraft has shown that it is in precisely the flight envelope in which rotorcraft typically operate where super-cooled water droplets naturally thrive to exist.[5] In the US alone since as lately as 2016, there have been multiple in-flight rotorcraft icing accidents recorded, many of which involved fatalities.[6–8]

*Research Scientist, Member AIAA, yuxiang.zhou@scicomp.uni-kl.de
†Professor, Associate Fellow AIAA, nicolas.gauger@scicomp.uni-kl.de
‡PhD Candidate, mylescarlo.morelli@polimi.it
§Professor, alberto.guardone@polimi.it
¶PhD Candidate, Student Member AIAA, hauthj@umich.edu
‖Assistant Professor, Member AIAA, xhuan@umich.edu

The icing phenomenon becomes even more problematic as the fundamental principle governing how rotorcraft generate their lift becomes a hindrance for the implementation of ice protection systems, since heating moving components is inherently expensive and inefficient. Thus, from a design and economic perspective civil manufacturers rarely employ de-icing mechanisms, which consequently limits rotorcraft operations under adverse conditions. For this reason, only 5% of US rotorcraft have full ice protection systems installed, and hence problems inevitably arise from light weight rotorcraft not intended for flight in icing conditions.[9] The use of advanced ice detection systems for warning pilots of ice formation in real-time is thus becoming ever more important to helicopter flight safety. However, rotor blades rotating at high speed makes visual detection of ice formation in-flight nearly impossible. The need to further understand rotorcraft in-flight icing is the critical first step in devising warning systems to detect early stage of ice formation.

One way to detect rotor blade in-flight icing is through its acoustic signatures—a severely iced rotor blade emits significantly different broadband noise component than a clean rotor blade. An experimental study by Cheng et al.[10] in 2016 first began quantifying the surface roughness from ice accretion via broadband noise measurements to assess whether noise signals may be used for an early warning ice detection system. These experimental efforts were labor-intensive and required highly specialized wind-tunnels to allow ice formation. An alternative approach later proposed by Chen et al.[11] in 2018 involved numerical simulations of ice accretion and computing the far-field noise signature of the iced rotor blade shape using computational aeroacoustics. To that end, the turbulent flow field around the effective shape of the blade, including the ice accretion, can be numerically simulated by solving the unsteady Reynolds-Averaged Navier Stokes (URANS) equations or via scale-resolving methods such as detached-eddy simulations (DES) or large-eddy simulations (LES). The unsteady fluctuations due to both blade motion and icing-induced flow separation can then be propagated to the far-field as noise by solving the Ffowcs-Williams-Hawkings (FWH) equation.[12]

While numerical simulations offer high-fidelity flow and acoustic solutions based on physical principles, their utilization for on-board real-time ice detection remains impractical. For example as illustrated in Fig. 1, after an acoustic signal is collected, an inverse problem needs to be solved to infer the corresponding ice profile that generated this acoustic signal, followed by an aerodynamic computation from that ice profile to obtain the relevant performance indicators such as the rotor blade lift and moment coefficients. Each URANS flow solve, even under a simplified two-dimensional representation of a pitching rotor blade, currently take minutes to hours to complete despite being parallelized with tens of processors. The computational cost would increase significantly when realistic three-dimensional rotor blades are simulated with scale-resolving methods. Additionally, under an inverse problem that is inherently multi-query and requiring possibly hundreds or more forward solutions, the overall computation time would be orders of magnitude higher than real-time requirements. We thus seek accelerations through machine learning (ML) techniques, and leverage:

- access to fast data-driven models and close the gap to real-time requirements;
- offline model construction making use of available computational resources for large-scale data generation; and
- "skipping" the inverse problem, instead creating a mapping (in the form of a ML model) directly from acoustic signal to the performance indicators (see Fig. 1).

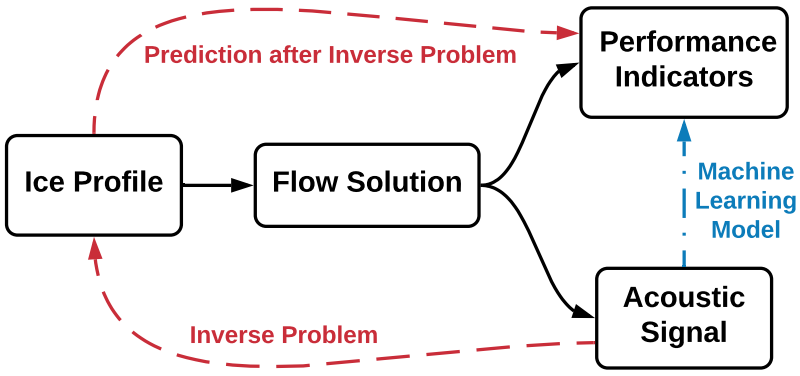


Figure 1: Workflow routes illustrating the physics causation (black solid), traditional inverse problem (dashed red), and proposed machine learning model (dotted-dash blue).

In this paper, we investigate the use of a particular type of ML model: deep neural networks (DNNs).[13,14] There has been a rapid growth of DNN usage in engineering applications and computational aerospace sciences, including

in rotorcraft aeroacoustics such as the utilization of multilayer feed-forward neural networks to classify and predict helicopter noise.[15,16] Indeed, DNNs are powerful functional forms observed to have high representability (theoretically shown to be a universal approximators[17]) and do not need user specification of good features for efficient model prediction. The training of well-performing of DNNs typically requires a large amount of data, which we plan to amass via numerical simulations using available offline computational resources. However, theoretical understanding of the performance and behavior for DNNs is relatively undeveloped, and they must be exercised with caution. One aspect that we emphasize in this paper concerns the uncertainty of ML models: *How much can we trust the DNN prediction?* This question is crucial for mission-critical scenarios where safety of the aircraft and crew are at risk, and must be addressed before certifying and deploying any ice detection technology based on ML models. We will introduce computational techniques to help address this question, by performing uncertainty quantification (UQ) for DNNs.

Traditional training of a DNN utilizes an optimization approach to "fit" the DNN to the training data, with the most popular algorithms based on stochastic gradient descent.[18,19] Upon completion, best fitted values are obtained for the DNN's weights (free parameters). Subsequently, when given an input, the DNN produces a *single-value* output prediction. That is, no uncertainty information is available to indicate the quality and credibility of that output value. Indeed, we expect the prediction uncertainty to vary depending on the architecture of the DNN; the quantity, quality, and informativeness of the training data; and the input regime. As a first step, we incorporate uncertainty in DNN by forming probability distributions to its weights, and updating this uncertainty from training data using Bayes' theorem—this DNN is then known as a *Bayesian neural network (BNN)*.[20–24] Modern DNNs in complex applications can easily contain millions of weights, and performing Bayesian inference on such high-dimensional spaces is often prohibitive. We approach this challenge by using variational inference (VI), where the closest approximation to the true uncertainty distribution is sought from a parametric family of distributions.[22,23] VI thus cannot capture structures outside of the selected family of distributions, but is extremely powerful for accommodating high-dimensional stochastic spaces. In this paper, we will employ VI to construct the BNNs.

This study focuses on an experimental initiative from the NASA Glenn Icing Research Tunnel (IRT) in which ice accretion on an oscillating airfoil was assessed to enhance the scarce dataset for the rotorcraft icing problem during pitching motion. Prior work on the validation of numerically computed ice shapes from this experimental test made the first attempt in introducing computational aeroacoustics to detect and characterize the noise signals of different ice structure produced from glaze and rime ice regimes.[25] Within this context, we seek to extend the research by employing ML techniques on different noise signals to train an ice detection system for warning pilots of potentially hazardous fight conditions. In particular, we propose to couple an icing simulation code with an aeroacoustic solver available in the open-source software suite SU2,[26] in order to compute the far-field noise signature corresponding to iced rotor blade shapes due to various icing conditions. Additionally, we propose to construct BNNs from the dataset thus generated to enable rapid predictions *with uncertainty information* of aerodyanmic performance indicators from acoustic signal, as a first step in developing an in-flight ice detection warning system. A two-dimensional pitching airfoil is considered for this proof-of-concept study while three-dimensional rotating blades are deferred to future studies.

II. Methodology

A. Turbulent Flow Simulation using SU2

The SU2 software suite[26] is an open-source collection of tools written in C++ and Python created for multi-physics simulation and design. It is built specifically for the analysis of partial differential equations (PDEs) and PDE-constrained optimization problems on unstructured meshes with state-of-the-art methods and algorithms, and is particularly well suited for aerodynamic shape design. Finite volume method (FVM) is applied on arbitrary unstructured meshes using a standard edge-based data structure on a dual grid with control volumes constructed using a median-dual, vertex-based scheme. The core of the suite is a Reynolds-averaged Navier-Stokes (RANS) solver which is used in this study in tandem with the Menter shear-stress transport (SST) turbulence model. Regarding time integration, SU2 is capable to solve implicitly steady and unsteady problems, using a dual-time stepping strategy, leading to second-order accuracy in space and time.

B. Particle Tracking and Ice Accretion

In the case of inherently unsteady flows such as the pitching airfoil considered in this study, the unsteady flow field simulated by the SU2 solver is seeded at every time step with super-cooled water droplets and their trajectories and

impingement limits are computed. The particle tracking solver used is an in-house code called PoliDrop, which is a Lagrangian based particle tracking solver developed primarily for computational icing simulations. The particle tracking method is based on the fully unsteady collection efficiency approach introduced in a preceding study.[25] It introduces the concept to ensure that the super-cooled water droplets fully take into account any unsteady aerodynamic effects which may be present in the flow field of an oscillating airfoil. This is shown to be particularly important during conditions where mixed glaze-rime icing conditions exist, producing complex ice shapes.

Information of the super-cooled water droplet impingement locations and collection efficiency is then passed onto an icing solver. The in-house code PoliMIce is used for computing the ice accretion.[27] The PoliMIce software library provides state-of-the-art ice formation models. The model used in this work to capture the complex experimental ice shapes is the local exact solution of the unsteady Stefan problem for the temperature profiles within the ice layer in glaze conditions.[28] The ice shapes are then computed using an unsteady multi-step approach, whereby non-linear ice accretion is accounted for by iteratively updating the surface solution on which the ice accretes. The working process for simulating ice shapes using multi-step ice accretion and the fully unsteady collection efficiency approach is shown in Fig. 2.

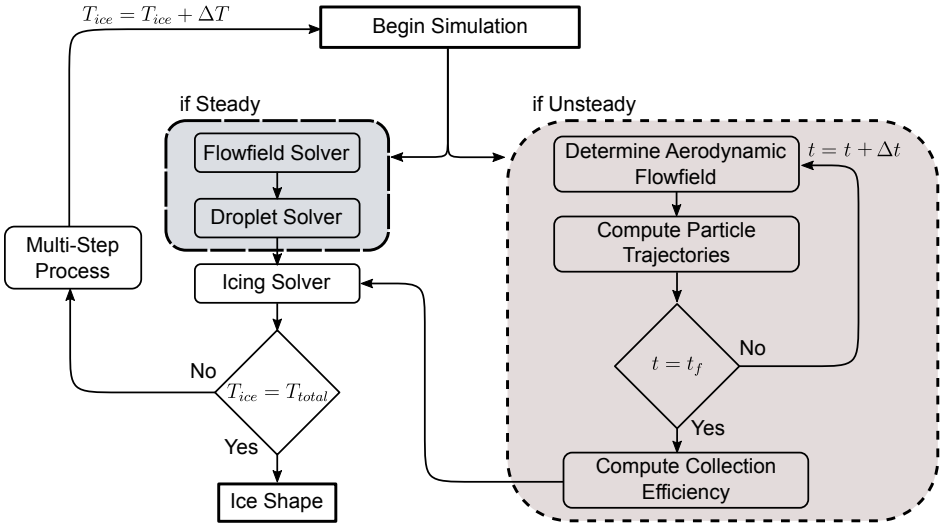


Figure 2: Schematic describing the working process for computing ice shapes using the fully unsteady collection efficiency approach.

C. Hybrid CFD-CAA Noise Prediction in SU2

Various iced airfoil shapes are subject to a computational aeroacoustic (CAA) analysis using the permeable-surface Ffowcs-Williams-Hawkings (FWH) formulation in SU2,[12] in order to help predict the early stages on ice accretion during flight to aid in warning the pilot of ice formation. The permeable FWH formulation allows fluid to flow through the fictitious surface Γ_p. Details of the flow field are then extracted and the noise source is then propagated to the far-field. This implementation is shown in the schematic from Figure 3. The computational domain is thus divided into two separate regions: the near-field computational fluid dynamics (CFD) region Ω_1, and the far-field CAA region Ω_2. The permeable surface Γ_p can be described by the shape function $f = 0$, where $f < 0$ indicates inside the surface and $f > 0$ outside the surface. The position of the FWH lies $\frac{3}{4}$ chords length from the airfoil trailing edge. The position of the observer locations is chosen based on where noise would likely be perceived on a conventional main rotor/tail rotor helicopter, which is directly below the main rotor.

In accordance with the icing simulation, the acoustic analysis is likewise considered as a two-dimensional model, although it is understood the rotor blade noise is most definitely not two-dimensional in nature. Thus this analysis aims at providing proof of concept for the detection of rotor blade performance changes under different ice structure characterizations.

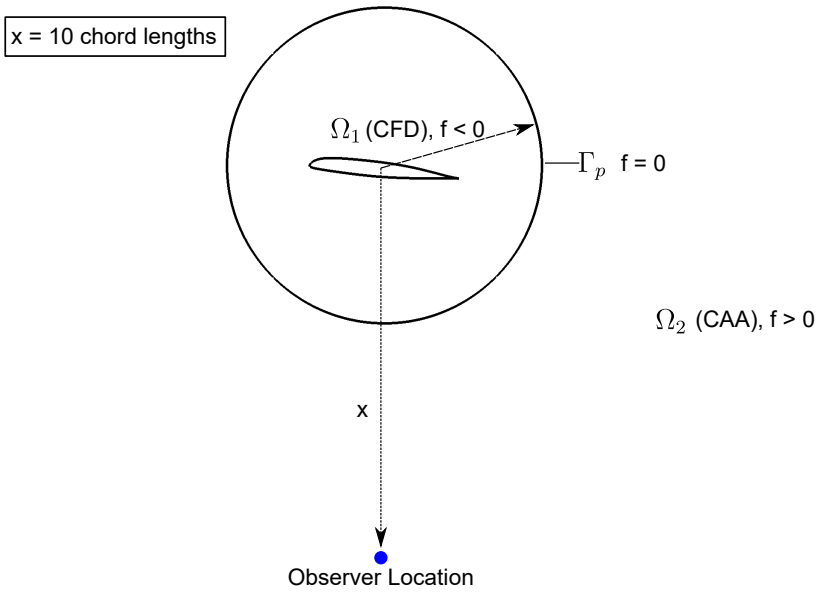


Figure 3: Schematic of the permeable control surface Γ_p separating the CFD and CAA domains and the relative observer locations.

D. Machine Learning Models

In this section, we first present background information for DNNs and its training process that does not capture uncertainty information, and then introduce BNNs that can provide uncertainty information to its predictions. Our goal is to build BNNs that map directly from acoustic signals to airfoil performance indicators.

1. Deep Neural Networks

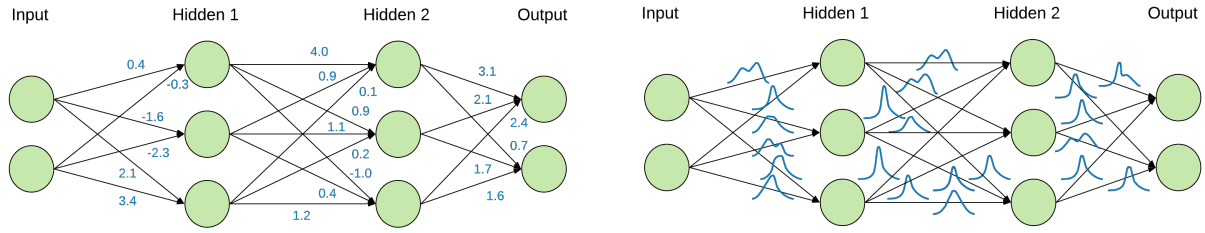
A neural network (NN) is a function that maps input x to output $\hat{y}$: we write $\hat{y} = f(x)$ where the hat in $\hat{y}$ denotes NN *prediction*. The interior composition of an NN can be highly complicated. For example, a densely connected feed-forward NN is organized into layers of different nodes, where each node performs calculations on output values from the previous layer before passing the result onto the next layer. Mathematically, the computation for node j in layer ℓ is

$$a_{\ell,j} = \phi_\ell \left(\sum_i W_{\ell,ij} a_{\ell-1,i} + b_{\ell,j} \right), \tag{1}$$

where $a_{\ell,j}$ is the nodal output, ϕ_ℓ is an activation function (e.g., rectified linear units (ReLU), hyperbolic tangent (tanh)), $a_{\ell-1,i}$ is the ith nodal input (which is the output from the ith node in the previous layer), $W_{\ell,ij}$ is the weight placed on the ith nodal input, and $b_{\ell,j}$ is a biasing term. Definitions vary as to what qualifies a neural network to be "deep", though generally a DNN consists of two or more hidden layers (i.e., layers that are not the input or output layer). As an example, Fig. 4 shows a graphical representation of a DNN with an input layer, two hidden layers, and an output layer. Adopting matrix-vector notation, it can be expressed as

$$\begin{aligned} a_0 &= x \\ a_1 &= \phi_1(W_1^T a_0 + b_1) \\ a_2 &= \phi_2(W_2^T a_1 + b_2) \\ a_3 &= \phi_3(W_3^T a_2 + b_3) = \hat{y}. \end{aligned}$$

The number of layers, number of nodes in each layer, and choice of ϕ_ℓ—together often referred to as hyperparameters—are usually selected before training, and choices leading to good performance are often unclear and require trial-and-error and experience. Nonetheless, once the network architecture is decided, the collection of all weights and all bias



(a) Deep neural network

(b) Bayesian neural network

Figure 4: DNN and BNN, each with an input layer, two hidden layers, and an output layer. Weights are shown in blue as deterministic values in DNN, and as probability density functions in BNN.

terms (which we denote by $w \equiv \{(W_\ell, b_\ell), \forall \ell\}$) constitute the free parameters to be tuned. Accordingly, we update our notation to $\hat{y} = f(x;w)$.

Given N training data points in the form of input-output pairs $(x_T, y_T) = \{x_n, y_n\}_{n=1}^N$, DNN training typically involves choosing w to minimize a loss function that reflects the degree of mismatch between the DNN predictions and the true output values. For example, the least squares loss involves solving

$$w^* = \arg\min_w \left\{ \frac{1}{N} \sum_{n=1}^{N} [f(x_n;w) - y_n]^2 \right\}. \tag{2}$$

The optimization is often carried out with gradient-based algorithms, such as stochastic gradient descent.[18,19] However, regardless of how Eqn. (2) is solved, its solution returns a single-valued w^*. Consequently, the prediction at a given input x is also single-valued: $\hat{y} = f(x;w^*)$, and it does not carry uncertainty information describing the quality and credibility of this predicted number. We now describe BNNs that are able to capture the uncertainty in DNN weights.

2. *Bayesian Neural Networks*

BNNs treat w as random variables with associated probability density functions (PDFs) representing the uncertainty on w. When training data become available, these PDFs are updated through Bayes' theorem:

$$p(w|x_T, y_T) = \frac{p(y_T|x_T, w)p(w)}{p(y_T|x_T)}, \tag{3}$$

where $p(w)$ is the prior PDF on the weight parameters and we also assumed $p(w|x_T) = p(w)$ (i.e., the prior uncertainty should not change from knowing only the input values of the training data), $p(y_T|x_T, w)$ is the likelihood function, $p(w|x_T, y_T)$ is the posterior PDF, and $p(y_T|x_T)$ is the Bayesian evidence. Solving the Bayesian inference problem then entails computing the posterior $p(w|x_T, y_T)$—that is, our updated uncertainty on w given the training dataset (x_T, y_T).

Characterizing the full $p(w|x_T, y_T)$ is extremely challenging considering w can easily be thousands- or even millions-dimensional for DNNs arising from complex engineering and science applications. While Markov chain Monte Carlo-based algorithms[29] can generate samples from the true posterior PDF, the extreme dimensionality renders direct applications of these methods intractable. We thus seek to approximate the posterior through VI, thereby turning the sampling task into an optimization problem, which is generally computationally less intensive. The aim of VI is to approximate the posterior PDF with another PDF from a parametric (and usually standard-form) family that is "close" in some sense. Denoting the approximating family to be $q(w;\theta)$ parameterized by θ, the Kullback-Leibler (KL) divergence is often adopted to quantify the degree of dissimilarity between $q(w;\theta)$ and the true posterior $p(w|x_T, y_T)$:

$$D_{\mathrm{KL}}[q(w;\theta) \parallel p(w|x_T, y_T)] = \int_{-\infty}^{\infty} q(w;\theta) \log \left[\frac{q(w;\theta)}{p(w|x_T, y_T)} \right] dw = \mathbb{E}_{q(w;\theta)} \left[\log q(w;\theta) - \log p(w|x_T, y_T) \right]. \tag{4}$$

Solving the VI problem entails finding θ^* that yields the closest posterior approximation:

$$\theta^* = \arg\min_\theta D_{\mathrm{KL}} \left[q(w;\theta) \parallel p(w|x_T, y_T) \right]. \tag{5}$$

We note that while both are optimization statements, the objective function in Eqn. (5) is not directly comparable to the least squares loss in Eqn. (2) since they target different goals: Eqn. (2) aims to best fit the data, while Eqn. (5) seeks to best approximate the true posterior PDF that is defined through conditional probability. A popular choice for $q(w;\theta)$ is the independent (mean-field) Gaussian family:

$$q(w;\theta) = \prod_{k=1}^{K} q(w_k;\theta_k) \tag{6}$$

where K is the total number of weight parameters in the network. The choice of independence allows decomposing the joint PDF into a product of marginal PDFs $q(w_k;\theta_k)$, each now following a univariate Gaussian $\mathcal{N}(\mu_k,\sigma_k^2)$ where the parameter $\theta_k = \{\mu_k,\sigma_k\}$ carries the values of mean and standard deviation. We will adopt this approximation family in this paper.

To numerically estimate the objective in Eqn. (4), we first apply Bayes' rule to the posterior term

$$D_{\mathrm{KL}}\left[q(w;\theta) \parallel p(w|x_T,y_T)\right] = \mathbb{E}_{q(w;\theta)}\left[\log q(w;\theta) - \log\left(\frac{p(y_T|x_T,w)p(w)}{p(y_T|x_T)}\right)\right]. \tag{7}$$

Since $p(y_T|x_T)$ is a constant with respect to θ and w, it can be excluded from the objective function without altering the minimizer. This is in fact a crucial step since the evidence $p(y_T|x_T)$ (also known as the marginal likelihood or simply the normalization constant) is generally very difficult to estimate. The remaining expression can then be approximated using an unbiased Monte Carlo estimator:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{q(w;\theta)}\left[\log q(w;\theta) - \log p(y_T|x_T,w) - \log p(w)\right] \tag{8}$$

$$\approx \arg\min_{\theta}\left\{\frac{1}{M}\sum_{m=1}^{M}\left[\log q\left(w^{(m)};\theta\right) - \log p\left(y_T|x_T,w^{(m)}\right)\right] - \log p\left(w^{(m)}\right)\right\} \tag{9}$$

where samples $w^{(m)}$ are drawn from the PDF $q(w;\theta)$. The objective function in Eqn. (8) is precisely the negative of the well-known *evidence lower bound (ELBO)*. We emphasize that Eqn. (9) is computable, since the terms $q\left(w^{(m)};\theta\right)$, $p\left(y_T|x_T,w^{(m)}\right)$, and $p\left(w^{(m)}\right)$ all can be evaluated. The optimization problem often can be approached leveraging gradient-based algorithms, where gradient with respect to θ can be obtained via applications of the chain-rule[23] or through automatic differentiation.

III. Results

A. In-Flight Icing

1. Test Conditions

The geometrical properties used with this work are based on the NASA Glenn IRT experimental test.[30] This consists of an untwisted Sikorsky SC2110 wing designed by Sikorsky/Lednicer and Owen as a rotorcraft airfoil and is undergoing pitch oscillation. The wing dimension has a 0.381 m chord and stretches from the floor to the ceiling of the wind tunnel. The airfoil has a maximum thickness at 9.9% at 37.7% chord and has a maximum camber of 1.9% at 15.7% chord. The results from this work demonstrated that there is significant similarity in ice shapes along the spanwise direction of the airfoil and thus further icing simulations can be considered essentially two-dimensional.

The environmental test conditions are based on the baseline low-speed flight test and are outlined in Table 1. These environmental conditions remain constant throughout the simulations. These operating conditions represent an airfoil at low flight speed and at a low mean angle of attack with significant amplitudes of oscillations. The length of time operating in icing conditions is long enough to expect significant ice accretion such that it could have severely damaging effects on the airfoil performance. The frequency of oscillation is high enough that unsteady hysteresis behaviour would be expected and is representative of a rotor blade pitching once per revolution. The outside air temperature depicts classical mixed glaze-rime ice accretion which can lead to the formation of severe ice shapes that cause significant disturbance to the flow field.

For this study, we conducted a total of 101 simulations, and assessed how the liquid water content and mean volume diameter of the super-cooled water droplets influenced the final ice shapes. The influencing parameters can be found in Table 2. The simulations begin with relatively low liquid waters contents and mean volume diameters of

Table 1: Test conditions remaining constant throughout the simulations.

Flight Speed (m/s)	Outside Air Temperature $(^{\circ}C)$	Angle of Attack $(^{\circ})$	Icing Time $(seconds)$	Oscillating Frequency (Hz)
77	-14	5 ± 6	600	5.8

super-cooled water droplets where minimal icing would be expected. There is then a steady incremental increase in the liquid water content and mean volume diameter of the super-cooled water droplets. The final simulations are then representative of super-cooled large water droplets with high liquid water contents which typically produce extreme and severe ice shapes.

Table 2: Test matix of parameters influencing the ice accretion simulations.

Simulation Number	Liquid Water Content (Kg/m^3)	Mean Volume Diameter (m)
1	0.000050	0.000005
2	0.000060	0.000006
3	0.000070	0.000007
.	.	.
.	.	.
.	.	.
↓	↓	↓
101	0.001050	0.000105

2. Matrix of Ice Shapes

The results of the icing simulations influenced by the test matrix from Table 2 are shown in Fig. 5. The simulation results indicate that lower liquid water contents and mean volume diameters of super-cooled water droplets produce less severe ice shapes and higher liquid water contents and mean volume diameters of super-cooled water droplets produce more severe ice shapes. It displays that the uniform incremental increase of the liquid water content and mean volume diameter of the super-cooled water droplets does not correspond to a uniform increase in ice thickness. This can be further explained through two main reasons:

1. As the liquid water content of the super-cooled water droplets increases, the icing conditions move from the dominantly rime ice regime to mixed glaze-rime ice conditions. These smoother ice shapes hence transition into far sharper and more acute ice shapes.

2. As the mean volume diameter increases the inertial forces of the super-cooled water droplets becomes greater. However, the model for determining if the inertial forces exceed the aerodynamic forces for impingement is a non-linear problem.

An overview of the effect the severe and mild ice shapes have on the performance characteristics of the airfoil is shown in Fig. 6. The severe ice shape is characteristic of an airfoil undergoing dynamic stall at high angles of attack with a sudden reduction in lift whilst the mild ice shape appears to have minimal effect on the lift at high angles of attack as shown in Fig. 6a. This dynamic stall behaviour of the severely iced airfoil is also displayed in the moment coefficient where a strong nose-up pitching moment is produced at high angles of attack as shown in Fig. 6b.

B. Noise Prediction for Different Ice Shapes

The far-field acoustic pressure signals at 10 chord-length directly below the aerodynamic center of clean, mildly iced and severely iced airfoils are shown in Fig. 7. The fluctuations in p' of the severely iced airfoil in this case shows vastly different noise signals than the clean and mildly iced airfoils. The frequency of the oscillating airfoil drives

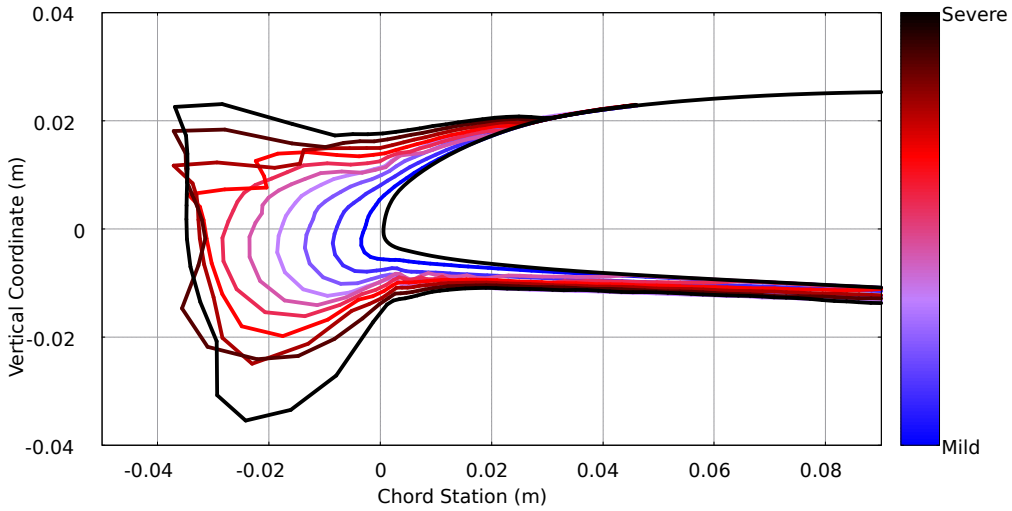
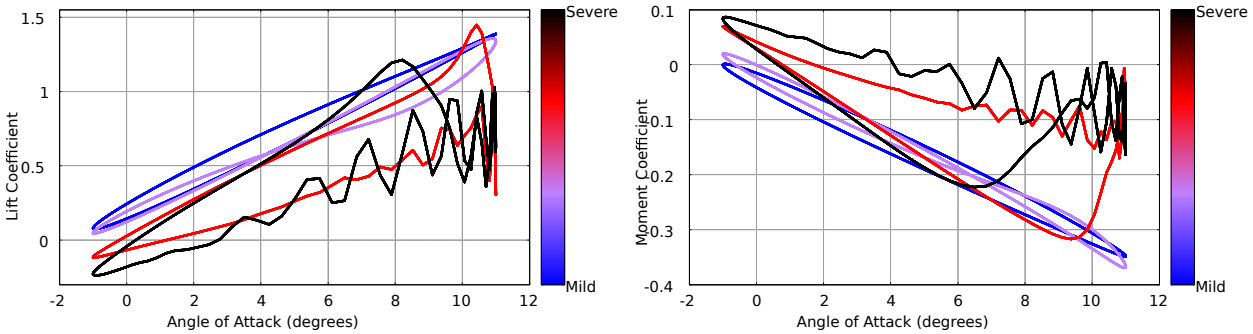


Figure 5: Sample of results produced from the icing simulations displaying the different ice shapes achieved from varying the liquid water content and mean volume diameter of the super-cooled water droplets.



(a) Lift Coefficient Hysteresis Loop

(b) Moment Coefficient Hysteresis Loop

Figure 6: Performance characteristics of mildly iced and severely iced airfoils.

the pressure fluctuations so there is clear periodicity present. The visibly different noise signals are attributed to be caused by the significant differences in the flow field—severe flow separation is observed on the iced airfoil. On the upstroke of the iced airfoil the perturbations of p' show similar values to the clean airfoil. That is until the occurrence of the dynamic stall behaviour at which point large scale vortices are produced developing peaks of broadband noise at a magnitude far greater than the peaks of the clean airfoil noise signal. These peaks then begin to oscillate at a far higher frequency than the frequency of the main noise signal. What is remarkable is the similarity between the high-frequency oscillations over each main oscillation. The far-field noise spectra of four iced airfoils are compared on Fig. 8. The correlation between ice shapes and their associated noise levels is evident—the severely iced airfoil produces a significantly elevated broadband noise level.

C. Preliminary Design of A Real-Time In-Flight Ice Detection System

We now proceed to construct BNN models that produce predictions of airfoil performance indicators given an input acoustic signal. First, training data are pre-processed. For each acoustic signal (e.g., Fig. 8), the spectral power density is discretized into 151 bins covering the frequency range 0–140 Hz. Log values of the spectral power density are then taken, and normalized using the mean and standard deviation statistics across all training data simulations. While normalization of input data is not strictly required, it is often observed to accelerate the the BNN optimization convergence. For each performance characteristics data (e.g., Fig. 6), we extract from within each simulation run its minimum, maximum, and mean values of C_L and C_M—these 6 quantities are designated as the output quantities of interest for our BNNs. To summarize, a BNN takes in 151 discretized spectral power density log values to predict a

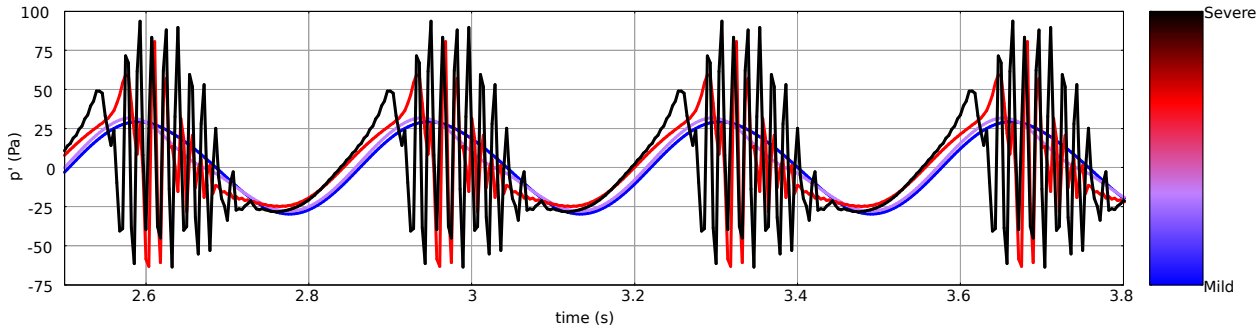


Figure 7: Far-field noise signal of mildly iced and severely iced airfoils.

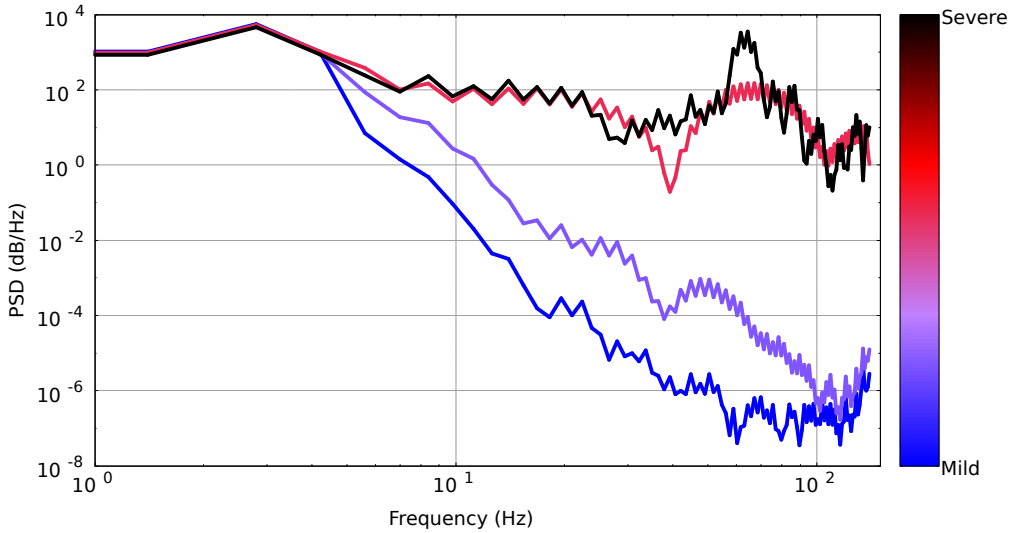


Figure 8: Far-field sound spectra of the mildly iced and severely iced airfoils.

total 6 airfoil performance indicators: $C_{L,min}, C_{L,max}, C_{L,mean}, C_{M,min}, C_{M,max}, C_{M,mean}$.

To set up the BNN optimization problem in Eqns. (8) and (9), the prior $p(w)$ is chosen to be independent Gaussian $w_k \sim \mathcal{N}(0,1^2)$; the likelihood $p(y_T|x_T,w)$ is constructed to reflect an additive independent Gaussian data noise $y_k = f(x_k;w) + \varepsilon_k, \varepsilon_k \sim \mathcal{N}(0,2^2)$; $M = 1000$ samples are used in the Monte Carlo estimator; and the ADAM[31] optimization algorithm is employed with a learning rate $\alpha = 0.02$ and mini-batch size of 20. Table 3 summarizes the BNN architecture adopted in our investigation, leading to a total of $K = 10,456$ network weights which corresponds to 20,912 variational parameters (each weight has a corresponding μ_k and σ_k).

Table 3: BNN architecture

Layer	Name	No. of Neurons	Activation Fcn.
1	input	151	-
2	hidden 1	50	tanh
3	hidden 2	50	tanh
4	output	6	linear

A total of 101 CFD-CAA simulations is currently available (with their ice accretion parameters in Table 2), and we compare BNNs trained using different number of data points: $N = 0, 1, 20, 80$. We note that for all of these four cases, we intentionally reserve some simulations as testing data, and hence only a maximum of 80 training data points are used (leaving 21 testing data for that largest case). As an example, Fig. 9 shows the convergence history of the BNN objective in Eqn. (9) (i.e., negative ELBO) with $N = 80$, and a plateau appears to have been reached with fewer

than 50 epochs. We make a few comments regarding the interpretation of this plot. First, the objective function being minimized here reflects the degree of approximation to the true posterior PDF that properly accounts the uncertainty in the weights, and is therefore different from the least squares loss in Eqn. (2) (or its variants such as with regularization) used in classical DNN training which generally directly targets data-fitting only. Second, the minimal objective value typically does not approach zero for two main reasons: (a) the estimator Eqn. (9) omits the constant term from the log evidence, $\log p(y_T|x_T)$, compared to Eqn. (7); and (b) the true posterior usually does not reside within the family of approximating distributions. Third, it is possible for the plateau to be a local minima, and optimization techniques to guard against local optima would be useful.

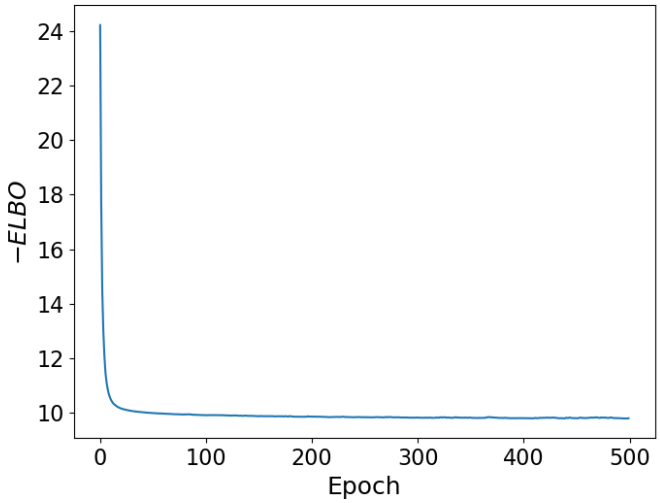


Figure 9: Convergence history of the BNN optimization objective (negative ELBO) based on 80 training data points, using the ADAM[31] optimizer with learning rate $\alpha = 0.02$ and mini-batch size 20.

Figure 10 presents results for the testing data points from each of the four $N = 0, 1, 20, 80$ cases (i.e., for simulations not use in training). On the x-axis are BNN predictions along with their uncertainty intervals ($\pm$ 1 standard deviation), and on the y-axis are their true airfoil performance values obtained from the CFD-CAA solver. Results for $N = 0$ thus reflect the prior predictive uncertainty (i.e., uncertainty before seeing any data). The reduction of uncertainty level can be observed as the number of training data increases: for example, increasing N from 20 to 80 decreased the predictive standard deviation by a factor of around 2. Figure 11 more closely analyzes the 6 predictive distributions but focusing on a specific input data point, corresponding to simulation number 1 in table 2. The figure contains histograms of predictions collected from 1000 samples drawn from the posterior PDFs of the weights. A number of interesting observations can be made. First, all predictive distributions appear near-Gaussian. This is likely due to that: the approximate posterior PDFs are currently restricted to the independent Gaussian family; the relationships between the weights to the outputs may be close to linear, especially since only two hidden layers are used; and the Bayesian asymptotic analysis (e.g., Chapter 5.3 of Bernardo & Smith[32]) guarantees convergence towards a Gaussian shaped posterior as N grows towards infinity. Second, the variation of the predictions can be dramatically different depending on the quantity being predicted. $C_{L,max}$ for example has a much greater spread than $C_{M,max}$ or $C_{M,mean}$. This is reflective of the reality that some observables may be more informative than others, and may follow more complex underlying trends.

The entire workflow is implemented in Python, with Tensorflow handling the linear algebra manipulations and automatic differentiation calculations only. On a 6 core 2.6 GHz Intel i7 processor, performing 1000 prediction evaluations for a given input to generate the histograms in Fig. 11 takes approximately 3.8 ms. Further code optimization, and implementing in languages such as C++ and Fortran would further improve the online prediction speed. These provide positive outlook for achieving real-time detection requirements.

Lastly, we discuss limitations of the BNN that warrant future investigations. These include the selection of the prior PDF and likelihood model, as well as the effects of hyperparameter choices on the neural network performance. For example, since the training data are from computer experiments, they do not contain observation noise. Then, it is more appropriate to interpret ε_k in the likelihood to be the model discrepancy between the BNN and the CFD-CAA

code, which may be difficult to characterize *a priori*. The approximating family of independent Gaussians also cannot capture any correlation among weights, or any non-Gaussian structures such as multi-modality and skewness in the posterior. Indeed, these properties arise from nonlinear models, but their extent and effect on the predictive uncertainty are problem-dependent and need to be explored. Numerically, our experience for solving the optimization problem in Eqn. (9) also involved trying different optimizers and their settings. For instance, employing stochastic gradient descent without momentum experienced difficulties in convergence. All of these challenges present avenues of future work, and are crucial for making BNN more robust and its predictive uncertainty characterization more accurate.

IV. Conclusion

In-flight rotor icing presents a serious problem in the operation of rotorcraft in cold climates. Reliable real-time detection of ice formation is a critical enabling technology in improving rotorcraft safety. In this work, we propose a novel approach towards developing a real-time in-flight ice detection system using computational aeroacoustics and Bayesian neural networks. In particular, a hybrid CFD-CAA solver in SU2 is used to compute both the aerodynamic forces of airfoils with various levels of ice accretion, as well as their corresponding far-field noise levels. The dataset generated is then used to construct a Bayesian neural network that provides (1) mapping from observed far-field broadband noise level to predictions of aerodynamic performance indicators, and (2) uncertainty information indicating the quality and credibility of these predictions. While the icing simulation, aeroacoustic prediction as well as the construction of Bayesian neural network model all require potentially extensive computational resources in the *offline* phase, the prediction of aerodynamic performance indicators based on measured far-field noise level as well as their associated uncertainty information can both be obtained rapidly in the time-critical *online* phase in-flight.

It should be noted that this first work as-is only serves as a proof-of-concept study. It is clear that the two-dimensional pitching airfoil considered in this study is a highly simplified representation of the complex rotorcraft icing problem in real life which involves turbulent blade-vortex interaction with rotating blades and three-dimensional ice formation along the blades. Various noise sources such as those from multiple rotors and engines, as well as rotor-fuselage interaction must be taken into account and differentiated in developing an ice detection system using the multi-disciplinary framework proposed in the current work.

V. Acknowledgement

The computational resources provided by the RHRK high performance computing center via the Elwetritsch high performance cluster at the TU Kaiserslautern is gratefully acknowledged. The work from Politecnico di Milano has received funding from the European Unions H2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 721920. Further information can be found at the Network for Innovative Training on ROtorcraft Safety (NITROS) project website.

References

[1]Hartman, P., Narducci, R., Peterson, A., Dadone, L., Mingione, G., Zanazzi, G., and Brandi, V., "Prediction of ice accumulation and airfoil performance degradation: a Boeing-CIRA research collaboration," *American Helicotoper Society*, Vol. 62, 2006, p. 123.

[2]Flemming, R., Alldridge, P., and Doeppner, R., "Artificial icing tests of the S-92A helicopter in the McKinley Climatic Laboratory," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, 2004, p. 737.

[3]Bernstein, B. C., Wolff, C. A., and McDonough, F., "An inferred climatology of icing conditions aloft, including supercooled large drops. Part I: Canada and the continental United States," *Journal of Applied Meteorology and Climatology*, Vol. 46, No. 11, 2007, pp. 1857–1878.

[4]Bernstein, B. C. and Le Bot, C., "An inferred climatology of icing conditions aloft, including supercooled large drops. Part II: Europe, Asia, and the globe," *Journal of Applied Meteorology and Climatology*, Vol. 48, No. 8, 2009, pp. 1503–1526.

[5]Roach, W., Forrester, D. A., Crewe, M. E., and Watt, K. F., "An icing climatology for helicopters," *Royal Meteorological Office Special Investigations Memo*, , No. 9, April 1984, pp. 41.

[6]NTSB, "National Transportation Safety Board Aviation Accident Final Report," NTSB No. WPR17FA047, December 2017.

[7]NTSB, "National Transportation Safety Board Aviation Accident Final Report," NTSB No. WPR16LA104, June 2018.

[8]NTSB, "National Transportation Safety Board Aviation Accident Final Report," NTSB No. ANC16FA023, January 2018.

[9]Heinrich, A., Ross, R., Zumwait, G., Provorse, J., Padmanabhan, V., Thompson, J., and Riley, J., "Aircraft Icing Handbook, Volumes I-III," *Atlantic City International Airport, NJ: FAA Technical Center*, 1991, pp. 8–1.

[10]Cheng, B., Han, Y., Brentner, K. S., Palacios, J. L., and Morris, P. J., "Quantification of rotor surface roughness due to ice accretion via broadband noise measurement," *American Helicopter Society 70th Annual Forum Proceedings*, 2014.

[11]Chen, X., Zhao, Q., and Barakos, G., "Numerical Analysis of Rotor Aero-acoustic Characteristics for Ice Detection based on HMB Solver," *American Helicopter Society International 73rd Annual Forum & Technology Display*, 2018.

[12]Zhou, B. Y., Albring, T., Gauger, N. R., Ilario, C. R., Economon, T. D., and Alonso, J. J., "Reduction of Airframe Noise Components Using a Discrete Adjoint Approach," AIAA-2017-3658, 2017.

[13]Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, The MIT Press, Cambridge, MA, 2006.

[14]LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature*, Vol. 521, No. 7553, 2015, pp. 436–444.

[15]Elshafei, M., Akhtar, S., and Ahmed, M. S., "Parametric models for helicopter identification using ANN," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 4, Oct 2000, pp. 1242–1252.

[16]Cenedese, F., *Helicopter Noise Prediction Using an Artificial Neural Network Tool*.

[17]Hornik, K., "Approximation Capabilities of Multilayer Neural Network," *Neural Networks*, Vol. 4, No. 1989, 1991, pp. 251–257.

[18]LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R., "Efficient BackProp," *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K.-R. Müller, Springer-Verlag Berlin Heidelberg, 2012, pp. 9–48.

[19]Robbins, H. and Monro, S., "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, Vol. 22, No. 3, 1951, pp. 400–407.

[20]MacKay, D. J. C., "A Practical Bayesian Framework for Backpropagation Networks," *Neural Computation*, Vol. 4, No. 3, 1992, pp. 448–472.

[21]Neal, R. M., *Bayesian Learning for Neural Networks*, Springer-Verlag New York, New York, NY, 1996.

[22]Graves, A., "Practical Variational Inference for Neural Networks," *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, Granada, Spain, 2011, pp. 2348–2356.

[23]Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D., "Weight Uncertainty in Neural Networks," *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37, 2015, pp. 1613–1622.

[24]Gal, Y., *Uncertainty in Deep Learning*, Ph.D. thesis, University of Cambridge, 2016.

[25]Morelli, M., Zhou, B. Y., and Guardone, A., "Simulation and Analysis of Oscillating Airfoil Ice Shapes via a Fully Unsteady Collection Efficiency Approach," *Proceedings of the 75th Annual Forum*, VFS International, Philadelphia, May 2019.

[26]Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J., " SU2: An Open-Source Suite for Multiphysics Simulation and Design," *AIAA Journal*, Vol. 54, No. 3, 2016, pp. pp. 828–846.

[27]Gori, G., Zocca, M., Garabelli, M., Guardone, A., and Quaranta, G., "PoliMIce: A simulation framework for three-dimensional ice accretion," *Applied Mathematics and Computation*, Vol. 267, September 2015, pp. 96–107.

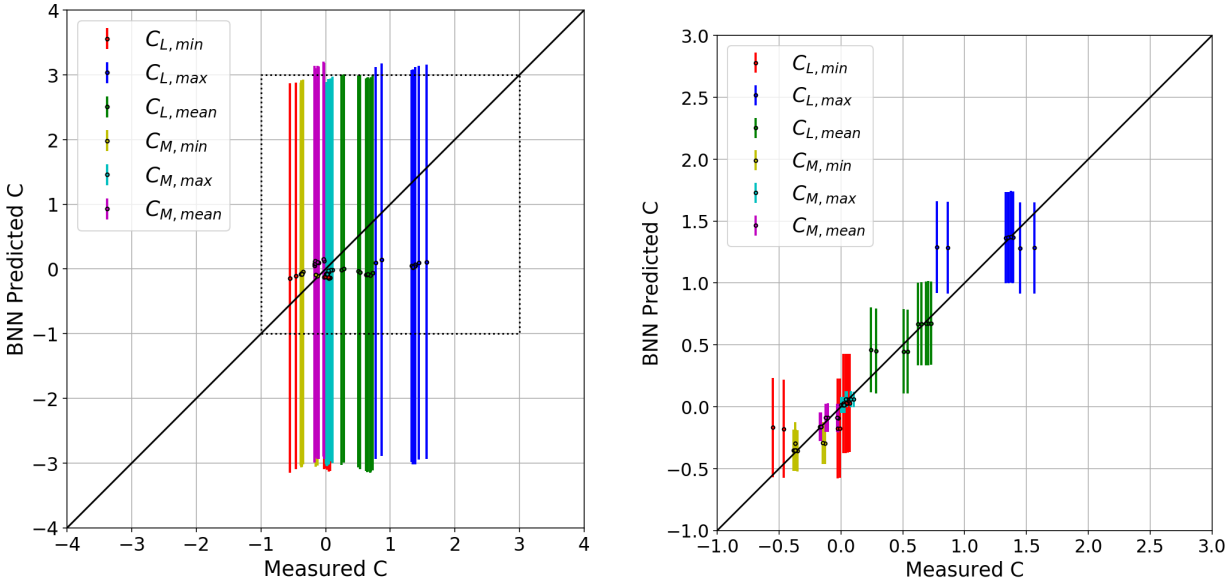
[28]Gori, G., Parma, G., Zocca, M., and Guardone, A., "Local Solution to the Unsteady Stefan Problem for In-Flight Ice Accretion Modeling," *Journal of Aircraft*, Vol. 55, No. 1, 2017, pp. 251–262.

[29]Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., *Markov Chain Monte Carlo in Practice*, Chapman & Hall, New York, NY, 1996.

[30]Reinert, T., Flemming, R. J., Narducci, R., and Aubert, R. J., "Oscillating Airfoil Icing Tests in the NASA Glenn Research Center Icing Research Tunnel," SAE Technical Paper No. 2011-38-0016., June 2011.

[31]Kingma, D. P. and Ba, J. L., "Adam: A method for stochastic gradient descent," *ICLR: International Conference on Learning Representations*, 2015.

[32]Bernardo, J. M. and Smith, A. F. M., *Bayesian Theory*, John Wiley & Sons, New York, NY, 2000.



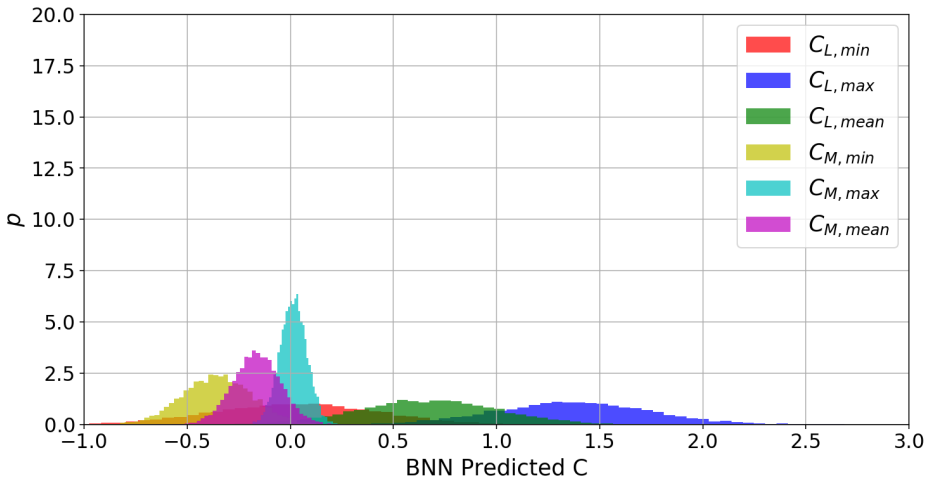
(a) $N = 0$ training data

(b) $N = 1$ training datum

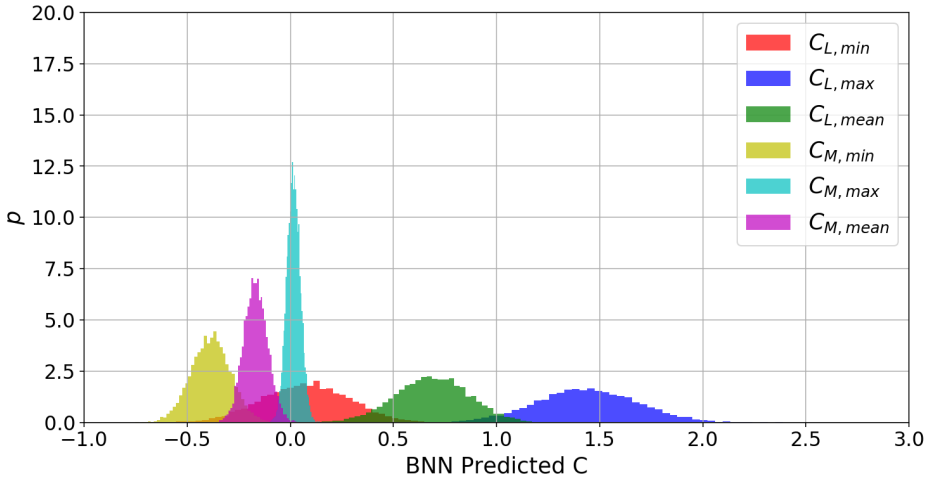
(c) $N = 20$ training data

(d) $N = 80$ training data

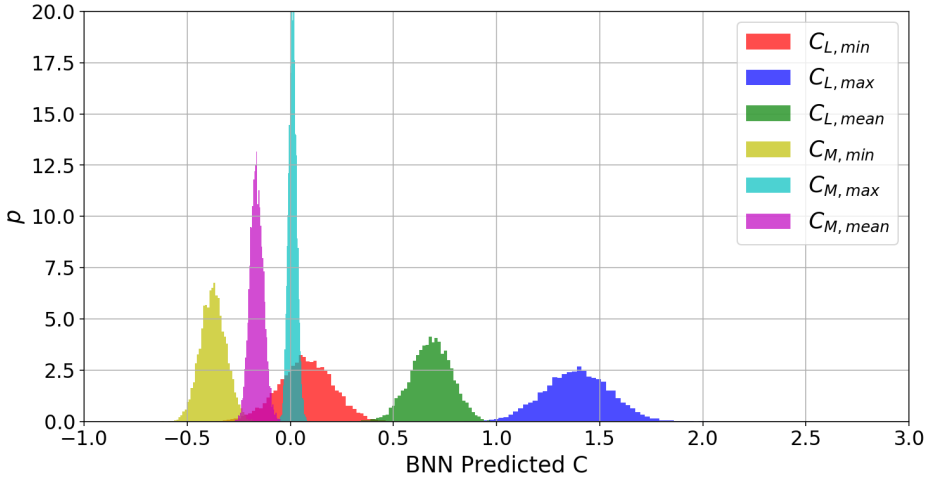
Figure 10: Prediction results for test data (simulations not used for BNN training) for each of the four cases, with the center dot representing the mean based on 1000 predictions, and error bars showing $\pm$ 1 standard deviation. Dashed box in Fig. 10a indicates the zoomed-in region adopted in plotting Figs. 10b-10d.



(a) $N = 1$ training datum



(b) $N = 20$ training data



(c) $N = 80$ training data

Figure 11: Normalized histograms from BNN predictions evaluated at a fixed single input (run number 1 from Table 2), where predictions are from 1000 samples drawn from the posterior PDFs of the weights. This is shown for networks trained on $N = 1, 20$, and 80 training data.