

Space Weather

RESEARCH ARTICLE

10.1029/2018SW002026

Key Points:

- We introduce a new method to estimate the uncertainties associated with single-point outputs generated by a deterministic model
- The method ensures a trade-off between accuracy and reliability of the generated probabilistic forecasts
- Computationally cheap model: no ensemble simulations, model independent, only inputs needed are the observed errors of the deterministic model

Correspondence to:

E. Camporeale,
enrico.camporeale@colorado.edu

Citation:




Camporeale, E., Chu, X., Agapitov, O. V., & Bortnik, J. (2019). On the generation of probabilistic forecasts from deterministic models. *Space Weather*, 17. <https://doi.org/10.1029/2018SW002026>

Received 23 JUL 2018

Accepted 14 FEB 2019

Accepted article online 20 FEB 2019

On the Generation of Probabilistic Forecasts From Deterministic Models

E. Camporeale^{1,2} , X. Chu³ , O. V. Agapitov⁴ , and J. Bortnik⁵ 

¹Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands, ²Cooperative Institute for Research in Environmental Sciences, University of Colorado, Boulder, CO, USA, ³Laboratory for Atmospheric and Space Physics, University of Colorado, Boulder, CO, USA, ⁴Space Sciences Laboratory, University of California Berkeley, Berkeley, CA, USA, ⁵Department of Atmospheric and Oceanic Sciences, University of California, Los Angeles, CA, USA

Abstract Most of the methods that produce space weather forecasts are based on deterministic models. In order to generate a probabilistic forecast, a model needs to be run several times sampling the input parameter space, in order to generate an ensemble from which the distribution of outputs can be inferred. However, ensemble simulations are costly and often preclude the possibility of real-time forecasting. We introduce a simple and robust method to generate uncertainties from deterministic models, that does not require ensemble simulations. The method is based on the simple consideration that a probabilistic forecast needs to be both accurate and well calibrated (reliable). We argue that these two requirements are equally important, and we introduce the Accuracy-Reliability cost function that quantitatively measures the trade-off between accuracy and reliability. We then define the optimal uncertainties as the standard deviation of the Gaussian distribution that minimizes the cost function. We demonstrate that this simple strategy, implemented here by means of a deep neural network, produces accurate and well-calibrated forecasts, showing examples both on synthetic and real-world space weather data.

Plain Language Summary We introduce a simple method to calculate the uncertainty associated with the output of a deterministic model (such as a physics model), and we show how to apply this method to cases relevant to Space Weather prediction. The details of the deterministic model are not important and the method can be applied both to empirical or physics-based models.

1. Introduction

The U.S. National Space Weather Action Plan released in October 2015 has fueled interest in so-called Operations-to-Research activities, which are now explicitly funded by NASA and NOAA programs. An important element of Operations-to-Research is the enhancement of existing operational models and products with fundamental research. A major weakness of most of the state-of-the-art forecasting models used by national Space Weather agencies is that they are essentially deterministic. For any given set of input parameters, they output a single-point estimate, without providing information on the uncertainty associated with such an estimate. On the other hand, the Space Weather community is gradually recognizing the importance of probabilistic forecasts, which have been the standard in meteorological weather forecast for many years. Indeed, several probabilistic forecasting models have been proposed in the last few years, concerning solar energetic particles (Aminalragia-Giamini et al., 2018; Kahler & Ling, 2015), geomagnetic indexes (Chandorkar et al., 2017; Gruet et al., 2018; McPherron & Siscoe, 2004; Riley & Love, 2016; Zhang & Moldwin, 2014), GPS scintillation (Prikryl et al., 2012), solar flares (Barnes et al., 2007; Bloomfield et al., 2012; Gallagher et al., 2002; Lee et al., 2012; Papaioannou et al., 2015; Wheatland, 2004), solar wind speed (Bussy-Virat & Ridley, 2014; Napoletano et al., 2018; Owens & Riley, 2017), and relativistic electron fluxes (Miyoshi & Kataoka, 2008), among others.

As pointed out in Murray et al. (2017), most operational space weather forecasting centers worldwide still rely on human forecasters to adjust the issued probability of a given event, based on experience. Yet, a recent verification of geomagnetic storm and X-ray flare forecasts issued by the Met Office Space Weather Operations Centre has reported that these forecasts struggle to provide a better prediction than a reference model and tend to overforecast events (Sharpe & Murray, 2017). Moreover, comparing 11 different methods to predict flares, Barnes et al. (2016) concluded that no participating method proved substantially better

than climatological forecasts, for M-class flares and above (a climatological model is one where a long-term average of the quantity of interest is taken as forecast).

There are two major approaches in producing a probabilistic model. The first way is to apply a statistical method on historical records, trying to correlate some input parameters with the forecast output. Little or no physics assumptions enter in such models (other than maybe a judicious choice of input parameters based on physics). For instance, modern machine learning algorithms, often referred to as *black-box* models, fall in this category (Camporeale, Wing, Johnson, et al., 2018; Camporeale, Wing, Johnson, 2018; Ghahramani, 2015; K. P. Murphy, 2012). A second way of producing a probabilistic forecast is based on the use of physics-based models, which range from (almost) first-principle simulations (e.g., Luhmann et al., 2017), to semiempirical models (e.g., Möstl et al., 2017). These *white-box* models are typically deterministic, meaning that they return a single solution for any given set of inputs provided. How to assign a probabilistic interpretation to such single-point estimates, in a computationally cheap way, is a challenging open problem which forms the core of a research area called non-intrusive Uncertainty Quantification (Smith, 2013). Non-intrusive refers to the fact that one employs a deterministic model (and its existing software), and performs an ensemble of simulations, without changing the underlying equations. It is then straightforward to extract a probabilistic description from the results ensemble. However, this is usually very expensive and brings the two following problems. First, if the number of inputs is large, one encounters the infamous *curse of dimensionality*, namely, the fact that the volume of a hypercube increases exponentially with the number of dimensions. Hence, sampling the input space with a tensorial grid (i.e., with a given number of points per dimension) quickly becomes unfeasible, because each grid point corresponds to a single run of a deterministic simulation. For this reason, sampling is often done in a Monte-Carlo fashion (or one of its modifications, such as Quasi-Monte-Carlo; Caflisch, 1998), which is very robust but also very slow in achieving convergence. Not surprisingly, an active area of research focuses on the design of adaptive sampling algorithms that yield convergence faster than Monte-Carlo (Babuška et al., 2007; Camporeale et al., 2017; Xiu, 2010; Xiu & Karniadakis, 2002). The second problem is that the distribution of outputs collected from the ensemble of simulations (the probabilistic forecast) is obtained by mapping, through the nonlinear simulation, the probability density that is assumed for the input parameters. Any misfit in the distribution of the inputs propagates to the distribution of outputs, producing misleading results. For this reason, an essential step of ensemble simulations is the calibration of the model (Kennedy & O'Hagan, 2001), that is the derivation of the distribution of the input parameters that is most consistent with observations. Calibration can itself be rather expensive, when it also relies on a large number of simulation runs.

In this paper we introduce a new method to derive a probabilistic forecast based on a deterministic model that avoids the computational costs associated with collecting an ensemble and with properly calibrating a computer simulation. Specifically, we introduce a method in which an artificial neural network (ANN) is trained to predict the spread of a Gaussian probability density function, such that a continuous real predictand can be replaced by a probabilistic forecast.

1.1. Accuracy and Reliability

This method is based on the simple consideration that a probabilistic forecast needs to be both accurate and reliable. This is in line with Gneiting et al. (2007), that have proposed to evaluate the performance of a forecast based on the paradigm of maximizing the sharpness of the predictive distributions subject to calibration. Sharpness refers to the concentration of the predictive distributions and is a property of the forecasts only. Note that in this paper we refer to calibration and reliability interchangeably, the former term typically being used in meteorological literature. Following the seminal paper by A. H. Murphy and Winkler (1992), accuracy is defined as the overall degree to which forecasts correspond to observations. It can be quantified introducing a proper scoring rule (Bröcker & Smith, 2007), whose examples are the Brier score for binary events (Brier, 1950), the Rank Probability Score for multcategory events, and its generalization for forecast of continuous variables, the Continuous Rank Probability Score (CRPS; Hersbach, 2000; Wilks, 2011), that we will use here. Reliability is the property of a probabilistic model that measures its statistical consistency with observations. In particular, for forecasts of discrete events, the reliability measures if an event occurs on average with frequency p , when it has been predicted to occur with probability p . For example, consider a probabilistic, binary, meteorological model that predicts rain or no-rain. Take a large enough sample of predictions of “70% chance of rain.” The model is said to be reliable/calibrated if approximately 70% of these predictions turned out to be true (i.e., it rained) and if this holds for all forecasted probabilities. The same concept can be extended to forecasts of a continuous scalar quantity by examining the so-called rank his-

togram (Anderson, 1996; Hamill, 1997, 2001) or the reliability diagram (Pinson et al., 2010). A reliability diagram represents, for any value of probability predicted for a given output, what is the actual observed frequency for that output (i.e., How many times did it rain, when 70% chance of rain was predicted? More (underconfident), less (overconfident), or exactly 70% of the time?). In the case of continuous variables, the reliability diagram is obtained with the following straightforward procedure. One collects a (large) number of pairs observations-predictions (the former being a real number, the latter a probability density). For each observation, one computes what was the probability that was assigned to the outcome being less or equal than the observed outcome. In the case of Gaussian predictions, this is simply the cumulative distribution function $P(y) = \frac{1}{2} \left[\operatorname{erf} \left(\frac{y-\mu}{\sqrt{2}\sigma} \right) + 1 \right]$, where y is the observed outcome, μ is mean of the predicted normal distribution, and σ is standard deviation. Once the list of all these probabilities is computed, the empirical distribution function associated to such list represents the reliability diagram. Once plotted, the range of assigned probabilities (from 0 to 1) is on the horizontal axis, and the frequency with which events occur, for each given assigned probability, is on the vertical axis. A perfectly calibrated model results in the reliability diagram following a straight diagonal line, while overconfident or underconfident predictions lie, respectively, below or above the diagonal line.

In any decision-making scenario, reliability is as important as accuracy: A nonreliable model (either because overconfident or underconfident) introduces a systematic bias which is hard to account for. In summary, reliability gives a quantitative measure of how consistently trustworthy (reliable, in common language) a predictive model is.

1.2. Proposed Strategy

The key to our approach is using a large set of model errors, defined as the difference between the prediction and the observation, to generate a predictive model of the standard deviation of the probabilistic forecast. In the simple case of a deterministic model with an input vector $[x_1, x_2, \dots, x_n]$ we build a model to predict $\sigma([x_1, x_2, \dots, x_n])$ so that we can specify a Gaussian PDF instead of the single deterministic value. This method is very general and decoupled from any particular choice for the model that predicts the output targets, which can lie anywhere in the range from white to black-box models, as long as the quantity of interest is real and continuous. Indeed, in the following we will assume that such a model, whose details are not important, is provided.

It is important to emphasize that the scope of this work is not to reduce the errors associated with the model, but to estimate the uncertainty of its output, thus generating a probabilistic forecast based on a deterministic model. The probabilistic forecast is designed to be a Gaussian probability distribution centered around the values produced by the model. In this way, the only unknown quantity is the variance of the Gaussian distribution. The simple strategy proposed here is to estimate this unknown variance (which is in general a function of the model inputs) by enforcing it to be a minimizer of a newly introduced cost function, which encodes a trade-off between accuracy and reliability, and that we call Accuracy-Reliability (AR) cost function. As we will show, when interpreted as a function of the variance (or its square root, the standard deviation), for fixed errors (the difference between model output and observed values), accuracy and reliability are competing objectives. This gives rise to a two-objective optimization problem and the well-known Pareto curve (Branke et al., 2008). This curve defines a boundary on which any further optimization of one objective (e.g., a better accuracy) results in worsening of the other objective (e.g., a worse reliability).

Although our method is essentially a multidimensional optimization problem, we require the ability to obtain an optimal value of the standard deviation σ for any set of values of model inputs and to ensure that σ is a smooth function of the inputs. While, in principle, standard algorithms like Newton or quasi-Newton methods could be used to directly solve the optimization problem, they would lead to the predicted standard deviation being a nonsmooth function of the inputs, and it would not be easy to generalize the results to unseen inputs. Therefore, we use an ANN that is trained on a given sample of model errors, for which the ground truth is known (that is, the true output of interest, not the true variance, that remains a latent variable).

Hence, our method reduces to a straightforward implementation of an ANN that outputs the standard deviation (as a function of the inputs) that minimizes the AR cost function. As a general strategy (and the one used in all of our examples), one can use the same inputs used by the deterministic model in the neural network. If some additional information is known about latent variables \mathbf{z} , then other inputs can be used; the ANN could be trained using an input vector $\mathbf{x} = [x_1, x_2, \dots, x_n, z_1, \dots, z_m]$.

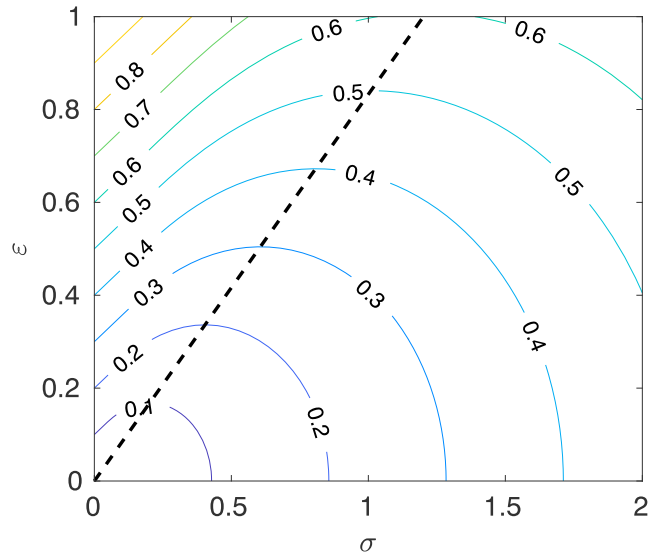


Figure 1. Lines of constant CRPS in (σ, ϵ) . The value of CRPS is indicated on the isolines. The black dashed line shows the location of σ_{\min} (i.e., the smallest CRPS for a given ϵ). CRPS = Continuous Rank Probability Score.

The paper is organized as follows. Section 2 introduces the mathematical background, the AR cost function, and it explains the methodology to derive the unknown uncertainties. Section 3 demonstrates the use of our methods for synthetic data and real-world examples relevant to space weather forecasting are presented in sections 4 and 5. Finally, conclusions are drawn in section 6.

2. Methodology

In this section we introduce and discuss the Continuous Rank Probability Score, which is widely used in many applications (Matheson & Winkler, 1976), and the new Reliability Cost for Gaussian forecasts.

2.1. Continuous Rank Probability Score (CRPS)

The CRPS is a generalization of the well-known Brier score (Wilks, 2011), used to assess the probabilistic forecast of continuous scalar variables, when the forecast is given in terms of a probability density function, or its cumulative distribution. CRPS is defined as

$$\text{CRPS} = \int_{-\infty}^{\infty} [P(y) - H(y - y^o)]^2 dy, \quad (1)$$

where $P(y)$ is the cumulative distribution (CDF) of the forecast, $H(y)$ is the Heaviside function, and y^o is the true (observed) value of the forecasted variable. CRPS is a negatively oriented score: it is unbounded and equal to zero for a perfect forecast with no uncertainty (deterministic).

In this paper we restrict our attention to the case of probabilistic forecast in the form of Gaussian distributions. Hence, a forecast is simply given by the mean value μ and the variance σ^2 of a Normal distribution. In this case $P(y) = \frac{1}{2} \left[\text{erf} \left(\frac{y - \mu}{\sqrt{2}\sigma} \right) + 1 \right]$ and the CRPS can be calculated analytically (Gneiting et al., 2005) as

$$\text{CRPS}(\mu, \sigma, y^o) = \sigma \left[\frac{y^o - \mu}{\sigma} \text{erf} \left(\frac{y^o - \mu}{\sqrt{2}\sigma} \right) + \sqrt{\frac{2}{\pi}} \exp \left(-\frac{(y^o - \mu)^2}{2\sigma^2} \right) - \frac{1}{\sqrt{\pi}} \right]. \quad (2)$$

Several interesting properties of the CRPS have been studied in the literature. Notably, its decomposition into reliability and uncertainty has been shown in Hersbach (2000). The CRPS has the same unit as the variable of interest, and it collapses to the Absolute Error $|y^o - \mu|$ for $\sigma \rightarrow 0$, that is when the forecast

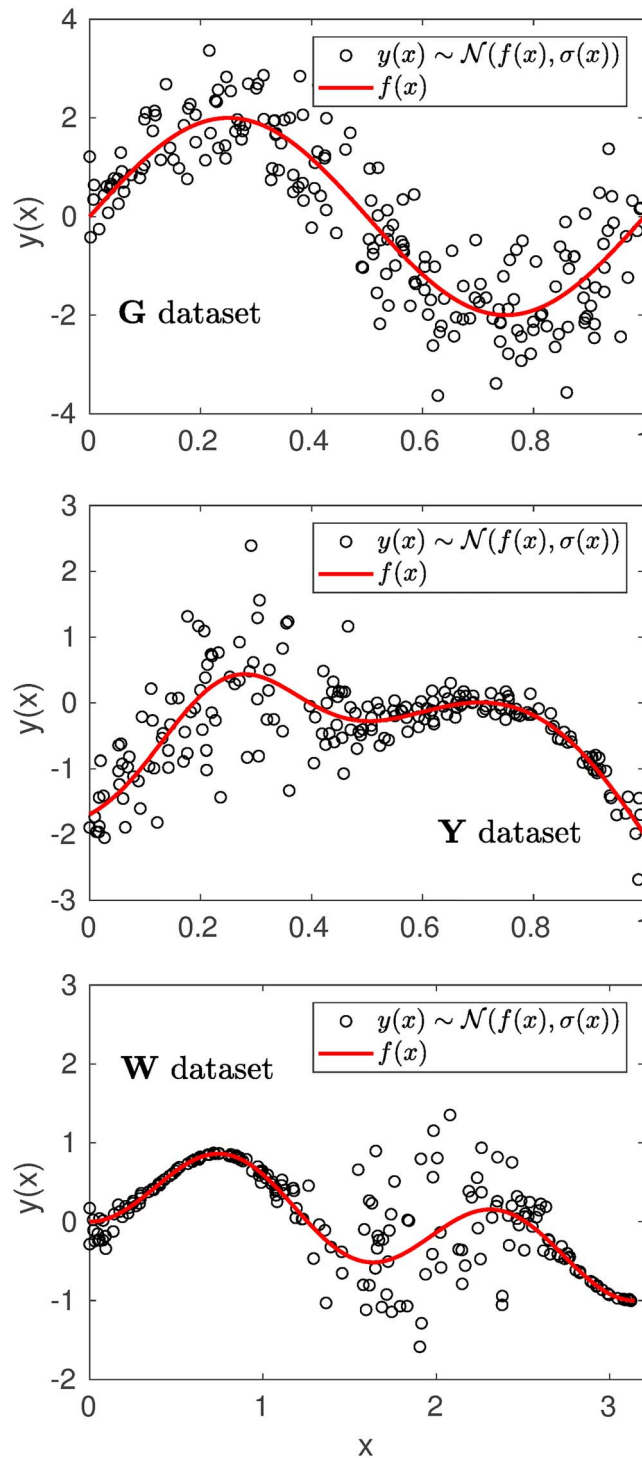


Figure 2. Circles: 200 points sampled from the **G**, **Y**, **W** data set (top, middle, bottom, respectively). The red line shows the mean function $f(x)$.

becomes deterministic. CRPS is defined for a single instance of forecast and observation, hence it is usually averaged over an ensemble of predictions of size N , to obtain the score relative to a given model: $\overline{\text{CRPS}} = \sum_k \text{CRPS}(\mu_k, \sigma_k, y_k^o) / N$. Since we are approaching the problem of variance estimation by assigning an empirical variance to predictions originally made as single-point estimates, it makes sense to minimize the CRPS as a function of σ only, for a fixed value of the error $\varepsilon = y^o - \mu$. By differentiating equation (2)

with respect to σ , one obtains

$$\frac{d\text{CRPS}}{d\sigma} = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) - \frac{1}{\sqrt{\pi}} \quad (3)$$

and the minimizer is found to be

$$\sigma_{\min}^{\text{CRPS}} = \frac{\varepsilon}{\sqrt{\log 2}}. \quad (4)$$

The CRPS penalizes underconfident and overconfident predictions in a nontrivial way. Indeed, for any value of the error ε , there are always two values of σ (one smaller and one larger than σ_{\min} , that is one overconfident and the other underconfident) that yield the same CRPS. We show in Figure 1 the isolines of CRPS in (σ, ε) space. The black dashed line indicates σ_{\min} . From this figure it is clear how a smaller error ε (for constant σ) always results in a smaller (better) score, but the same score can be achieved by changing both the error ε and the standard deviation σ . A straightforward way of understanding how CRPS works is the following. Let us start with a prediction that has a given error ε and no uncertainty (i.e., a deterministic forecast, $\sigma = 0$). CRPS attributes a certain score to such prediction. Now, if we increase ε the prediction becomes obviously worse, hence CRPS increases, unless we simultaneously increase the uncertainty σ . That is, accounting for the fact that the prediction is uncertain compensates for a larger error. In this way one can move along a constant CRPS curve, until the point (on the dashed line) where an increase in error cannot be compensated any further by an increase in uncertainty. After that point, larger uncertainties must then be compensated by a *decrease* in the error ε .

2.2. Reliability Score for Gaussian Forecast

Contrary to the CRPS, that is defined for a single pair of forecast-observation, it is clear that reliability can only be defined for a large enough ensemble of such pairs, being a statistical property of a model. We define the standardized errors η_i as

$$\eta_i = \varepsilon_i / (\sqrt{2}\sigma_i), \quad (5)$$

where the standard deviations σ_i are determined by the input vector. If $\sigma(\mathbf{x})$ is not constant then this definition acts to both standardize and transform the error distribution. While the forecast errors ε_i may not be Gaussian, in the case of a normally distributed forecast we expect η calculated over a sample of N prediction-observation pairs to follow a standard normal distribution with CDF $\Phi(\eta) = \frac{1}{2}(\text{erf}(\eta) + 1)$. Hence, we define the Reliability Score (RS) as

$$\text{RS} = \int_{-\infty}^{\infty} [\Phi(y) - C_{\eta}(y)]^2 dy, \quad (6)$$

where $C_{\eta}(y)$ is the empirical cumulative distribution of the standardized errors η , that is

$$C_{\eta}(y) = \frac{1}{N} \sum_{i=1}^N H(y - \eta_i) \quad (7)$$

with $\eta_i = (y_i^o - \mu_i) / (\sqrt{2}\sigma_i)$.

RS measures the divergence of the empirical distribution of standardized errors η from a standard normal distribution. Note that, by appropriately choosing σ , one can always obtain a distribution of η that approximates a standard normal distribution, irrespective of the distribution of the errors ε , as long as the number of instances of $\varepsilon < 0$ and $\varepsilon > 0$ are approximately equal.

From now on we will use the convention that the set $\eta = \{\eta_1, \eta_2, \dots, \eta_N\}$ is sorted ($\eta_i \leq \eta_{i+1}$). This does not imply that μ_i or σ_i are sorted as well. Interestingly, the integral in equation (6) can be calculated analytically, via expansion into a telescopic series, yielding:

$$\text{RS} = \sum_{i=1}^N \left[\frac{\eta_i}{N} (\text{erf}(\eta_i) + 1) - \frac{\eta_i}{N^2} (2i - 1) + \frac{\exp(-\eta_i^2)}{\sqrt{\pi}N} \right] - \frac{1}{2} \sqrt{\frac{2}{\pi}}. \quad (8)$$

Differentiating now the i th term of the above summation, RS_i , with respect to σ_i (for fixed ε_i), one obtains

$$\frac{d\text{RS}_i}{d\sigma_i} = \frac{\eta_i}{N\sigma_i} \left(\frac{2i - 1}{N} - \text{erf}(\eta_i) - 1 \right), \quad (9)$$

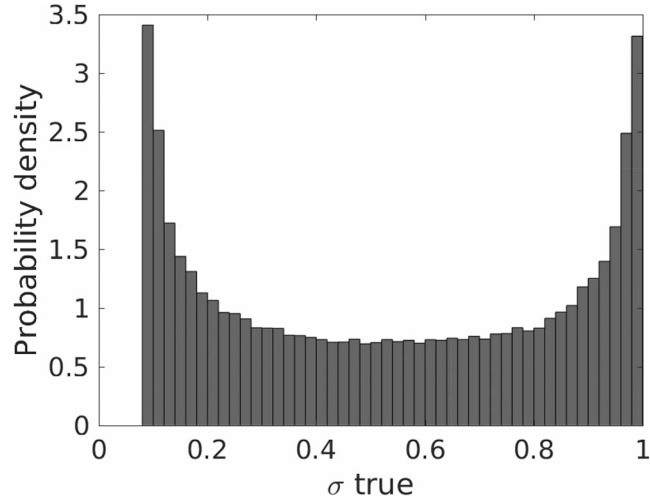


Figure 3. Distribution of true values of standard deviation σ for the 5-D data set.

which is minimized at the value $\sigma_{\min}^{\text{RS}}$ that satisfies

$$\text{erf}\left(\frac{\varepsilon_i}{\sqrt{2}\sigma_{\min}^{\text{RS}}}\right) = \frac{2i-1}{N} - 1. \quad (10)$$

This could have been trivially derived by realizing that by minimizing RS one obtains the distribution of standardized errors η_i that most closely approximates a standard normal distribution, for a given number of observations N . This is the distribution that mapped through Φ divides uniformly the interval $[0, 1]$: $\frac{1}{2}(\text{erf}(\eta_i) + 1) = \frac{i-1/2}{N}$, that is, the set $\{\frac{1}{2N}, \frac{1}{2N} + \frac{1}{N}, \frac{1}{2N} + \frac{2}{N}, \dots, 1 - \frac{1}{2N}\}$. Like CRPS, RS is negatively oriented (i.e., zero is the perfect score). It can be equal to zero only for $N \rightarrow \infty$.

2.3. The AR Cost Function

The AR cost function introduced here follows from the simple principle that the empirical standard deviations σ_i estimated from an ensemble of errors ε_i should result in a model that is both accurate (with respect to the CRPS score) and reliable (with respect to the RS score). This gives rise to a two-objective optimization problem. It is trivial to verify that CRPS and RS cannot simultaneously attain their minimum value (for fixed errors ε_i). Note that CRPS is a function of ε_i and σ_i , while RS is only a function of their scaled ratio $\eta_i = \varepsilon_i/(\sqrt{2}\sigma_i)$. By minimizing the CRPS, $\eta_i = \frac{1}{2}\sqrt{\log 4}$ for any i (see equation (4)). Obviously, a constant η_i cannot result in a minimum also for RS, according to equation (10). Moreover, notice that trying to minimize RS as a function of σ_i (for fixed errors ε_i) results in an ill-posed problem, because one can have infinite combinations of σ_i that result in the same set η , therefore there is no unique solution for the standard deviations that minimizes RS. Hence, RS can be thought of as a regularization term in the AR cost function. The simplest strategy to deal with multiobjective optimization problems is to scalarize the cost function, which we define here as

$$\text{AR} = \beta \cdot \overline{\text{CRPS}} + (1 - \beta)\text{RS}. \quad (11)$$

We choose the scaling factor β as

$$\beta = \text{RS}_{\min}/(\overline{\text{CRPS}}_{\min} + \text{RS}_{\min}). \quad (12)$$

The minimum of $\overline{\text{CRPS}}$ is $\overline{\text{CRPS}}_{\min} = \frac{\sqrt{\log 4}}{2N} \sum_{i=1}^N \varepsilon_i$, which is simply the mean of the errors, rescaled by a constant. The minimum of RS follows from equations (8) and (10):

$$\text{RS}_{\min} = \frac{1}{\sqrt{\pi N}} \sum_{i=1}^N \exp\left(-\left[\text{erf}^{-1}\left(\frac{2i-1}{N} - 1\right)\right]^2\right) - \frac{1}{2}\sqrt{\frac{2}{\pi}}. \quad (13)$$

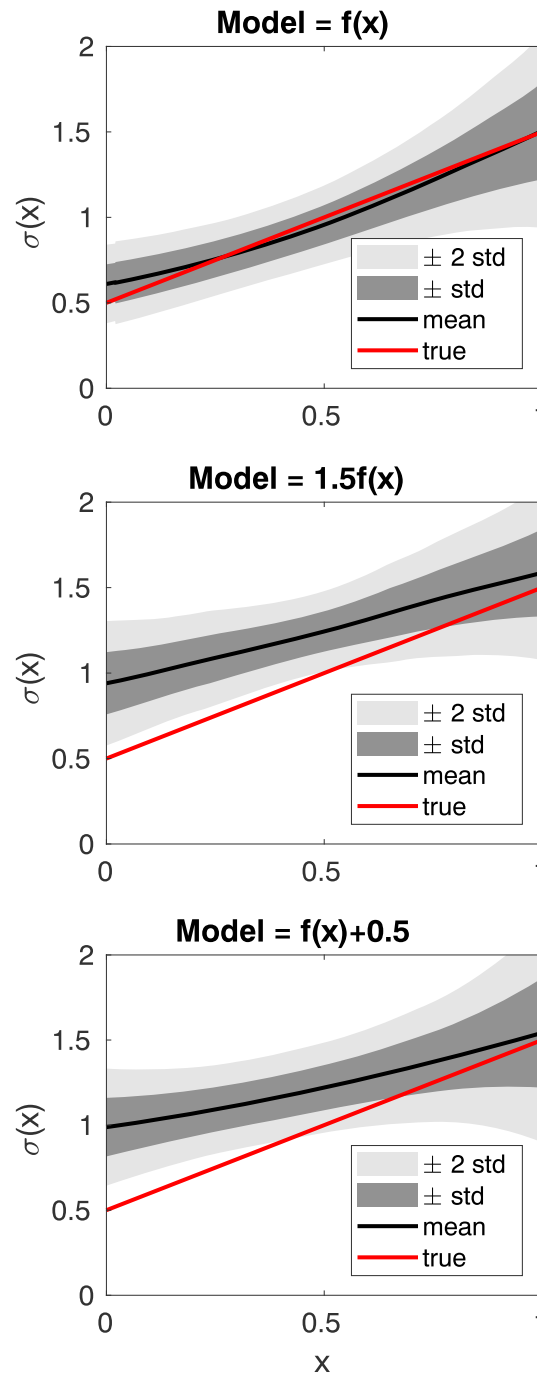


Figure 4. Results for the **G** data set. Values derived for the standard deviation σ , averaged over 200 independent runs (black), compared to the ground truth values used to generate the data (in red). The shaded gray area represents the confidence intervals of one (dark gray) and two (light gray) standard deviations calculated over the ensemble of 200 runs. (top) The correct mean function $f(x)$ is used for the model; (middle) a misspecified model that uses $1.5f(x)$ as mean; (bottom) a misspecified model that uses $f(x) + 0.5$ as mean.

Notice that RS_{min} is only a function of the size of the sample N , and it converges to zero for $N \rightarrow \infty$. The heuristic choice in equation (12) is justified by the fact that the two scores might have different orders of magnitude, and therefore we rescale them in such a way that they are comparable in our cost function (11). Indeed, the scaling factor β ensure that the two terms would be exactly equal if both could be minimized simultaneously. We believe this to be a sensible choice, although there might be applications where one

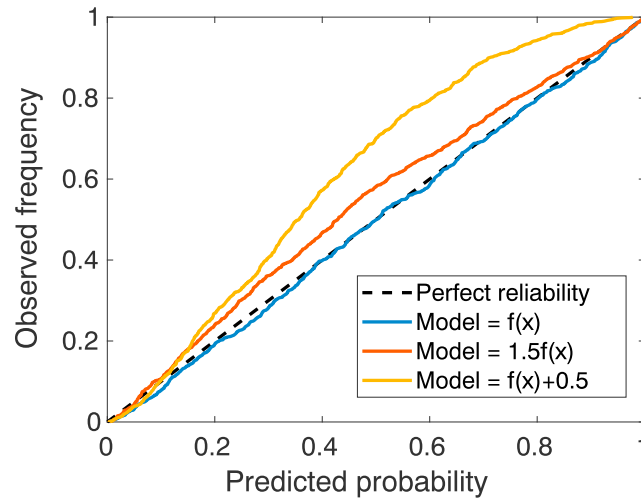


Figure 5. Reliability diagram for the method applied to the **G** data set. Blue, red, and yellow lines denote the observed frequency as function of the predicted probability, for the cases of correct mean function $f(x)$, and misspecified models $1.5f(x)$ and $f(x) + 0.5$, respectively. A perfect reliability is shown as a black dashed line.

would like to weigh the two scores differently. Also, in our practical implementation, we neglect the last constant term in the definition (8) so that, for sufficiently large N , $RS_{min} \simeq \frac{1}{2} \sqrt{\frac{2}{\pi}} \simeq 0.4$.

2.4. Neural Network

In summary, we want to estimate the input-dependent values of the empirical standard deviations σ_i associated to a sample of N observations for which we know the errors ε_i . We do so by solving a multidimensional optimization problem in which the set of estimated σ_i minimizes the AR cost function defined in equation (11).

This newly introduced cost function has a straightforward interpretation as the trade-off between accuracy and reliability, which are two essential but conflicting properties. In practice, we want to generate a model that is able to predict σ as a function of the inputs \mathbf{x} on any point of a domain. This unknown function can in general be nonlinear, and we assume no a priori information to constraint its functional form. However, we want to enforce smoothness of the unknown variance, to some degree. A very general strategy is to use a regularized ANN to model the dependency of σ as a function of the inputs. However, it is important to realize that this is not the only choice, and in case the user has some prior information on the functional form of σ , other strategies (such as polynomial regression, if the input is low-dimensional) might be better suited. For simplicity, we choose a single neural network architecture, that we use for all the tests. We use a network with two hidden layers, respectively, with 20 and 5 neurons. The activation functions are tanh and a symmetric saturating linear function, respectively. The third (output) layer uses a linear activation function. The data set, composed of the inputs \mathbf{x} and the corresponding observed errors ε , is randomly divided into training (70%) and validation (30%) sets. The network is trained using a standard Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm, and the iterations are forcefully stopped when the loss function does not decrease for 10 successive iterations on the validation set. These are all standard choices when training neural networks, and we refer the reader to specific monographs (e.g., Bishop, 1995).

A very attractive feature of our model is that the only inputs needed are the input parameters x_i and the corresponding errors ε_i (used for training only). The neural network outputs the values of $\log(\sigma_i)$, by minimizing the above-introduced AR cost function, equation (11), where $\sigma(\mathbf{x})$ is the standard deviation, and \log is used to enforce its positivity. In order to limit the expressive power and avoid overfitting, we may add a regularization term equal to the L_2 norm of the weights to the AR cost function, multiplied by a constant factor 0.2. In other words, a term $0.2 \frac{w^T w}{2}$ can be added to the AR cost function defined in equation (11), where the vector w represents the Neural Network weights. This is a standard procedure to constrain the amplitude of the weights and avoid overfitting (because highly nonlinear functions tend to increase the regularization term; see, e.g., Carè & Camporeale, 2018). In our numerical experiments (section 3) this regularization term was needed only for 1-D cases. Finally, in order to avoid local minima due to the random initialization of

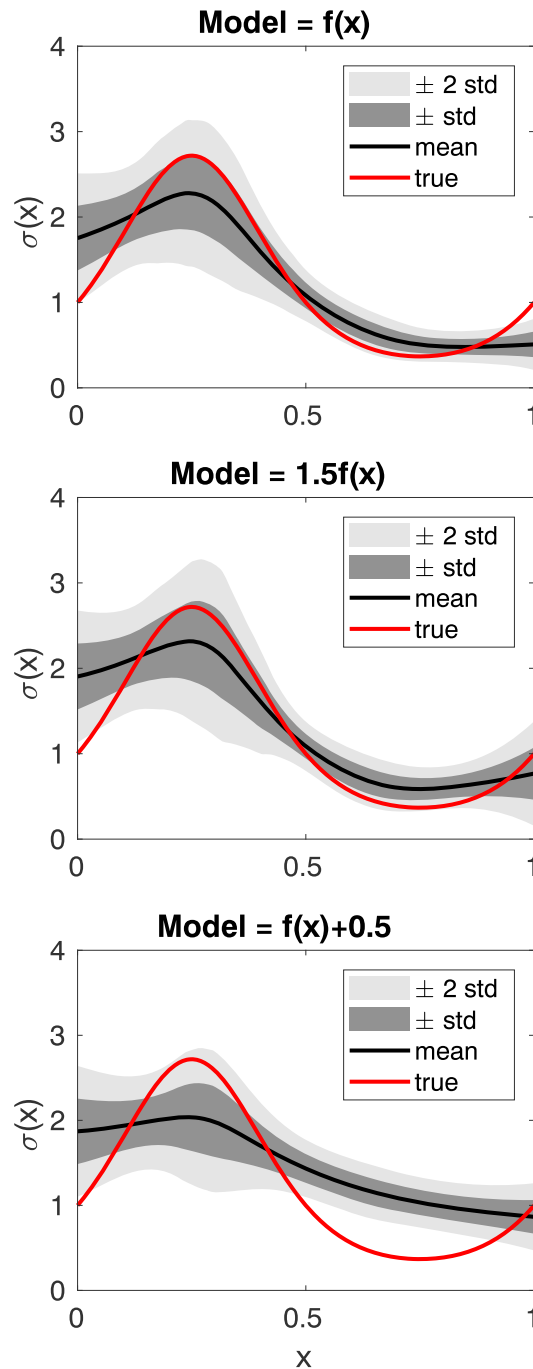


Figure 6. Results for the Y data set. Values derived for the standard deviation σ , averaged over 200 independent runs (black), compared to the ground truth values used to generate the data (in red). The shaded gray area represents the confidence intervals of one (dark gray) and two (light gray) standard deviations calculated over the ensemble of 200 runs. (top) The correct mean function $f(x)$ is used for the model; (middle) a misspecified model that uses $1.5f(x)$ as mean; (bottom) a misspecified model that uses $f(x) + 0.5$ as mean.

the neural network weights, we train five independent networks and choose the one that yields the smallest value of the cost function.

3. Experiments With Synthetic Data

In this section we show some experiments on synthetic data to demonstrate the ease, robustness, and accuracy of the presented method to derive uncertainties. Here, we assume to have an imperfect model

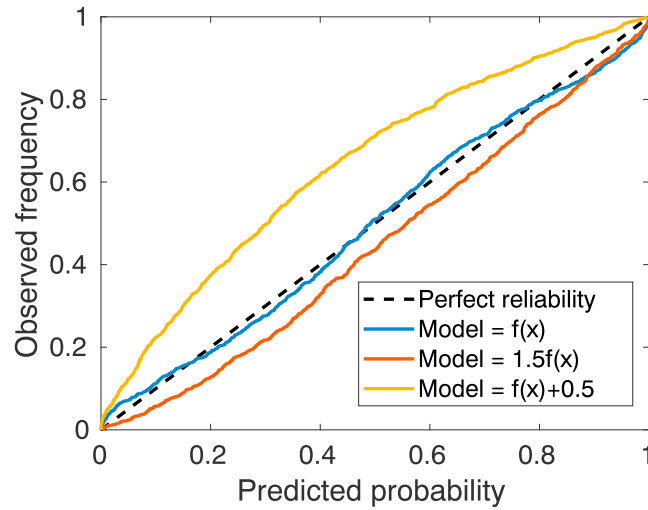


Figure 7. Reliability diagram for the method applied to the **Y** data set. Blue, red, and yellow lines denote the observed frequency as function of the predicted probability, for the cases of correct mean function $f(x)$, and misspecified models $1.5f(x)$ and $f(x) + 0.5$, respectively. A perfect reliability is shown as a black dashed line.

that produces a forecast $y = f(x)$. The synthetic observations are generated from a Gaussian distribution $\mathcal{N}(f(x), \sigma(x)^2)$, with known variance $\sigma(x)^2$. The stochastic nature of the synthetic data can be thought to mimic the existence of latent variables that are not included in the model. In other words, close values of the input x can result in very different outputs, because of unmodeled processes. The purpose of these experiments is to show that our method is capable of recovering the functional dependence of the variance $\sigma(x)^2$, that is, for real data, unknown. We choose some of the data sets routinely used in machine learning literature (Kersting et al., 2007). The first three data sets are one-dimensional in x , while in the fourth we will test the method on a five-dimensional space, thus showing the robustness of the proposed strategy.

G data set: $x \in [0, 1]$, $f(x) = 2 \sin(2\pi x)$, $\sigma(x) = x + \frac{1}{2}$ (Goldberg et al., 1998).

Y data set: $x \in [0, 1]$, $f(x) = 2(\exp(-30(x - 0.25)^2) + \sin(\pi x^2)) - 2$, $\sigma(x) = \exp(\sin(2\pi x))$ (Yuan & Wahba, 2004).

W data set: $x \in [0, \pi]$, $f(x) = \sin(2.5x) \sin(1.5x)$, $\sigma(x) = 0.01 + 0.25(1 - \sin(2.5x))^2$ (Nix & Weigend, 1994; Williams, 1996).

Examples of 200 points sampled from the **G**, **Y**, and **W** data sets are shown in Figure 2 along with their mean function $f(x)$ in red.

For the **G**, **Y**, and **W** data sets we test the case where the true mean function $f(x)$ is used as deterministic model, and two cases where the model suffers of a systematic bias and the model output is replaced by $\frac{3}{2}f(x)$ (a multiplicative error) or $f(x) + \frac{1}{2}$ (an additive error). These two cases serve also the purpose of studying the behavior of the proposed method for non-Gaussian errors. Every model is trained on 100 points uniformly sampled in the domain.

5-D data set: $\mathbf{x} \in [0, 1]^5$, $f(\mathbf{x}) = 0$, $\sigma(\mathbf{x}) = 0.45(\cos(\pi + \sum_{i=1}^5 5x_i) + 1.2)$ (Genz, 1984). Figure 3 shows the distribution of σ , which ranges in the interval $[0.09, 0.99]$.

The **5-D** data set is obviously more challenging, hence we use 10,000 points to train the model (note that this results in fewer points per dimension, compared to the one-dimensional tests). For all experiments we test 200 independent runs.

The results for the **G** data set are shown in Figure 4. The values derived for the standard deviation σ , averaged over 200 independent runs are shown in black, compared to the ground truth value used to generate the data (in red). The shaded gray area represents the confidence intervals of one (dark gray) and two (light gray) standard deviations calculated over the ensemble of 200 runs. The top, middle, and bottom panels show the results when the model uses the exact mean function used to generate the data $f(x)$ and when the model is misspecified by a multiplicative error ($1.5f(x)$), or an additive error ($f(x) + 0.5$), respectively. One can notice that our method is capable of recovering almost exactly the true variance (top), when the model is accurate.

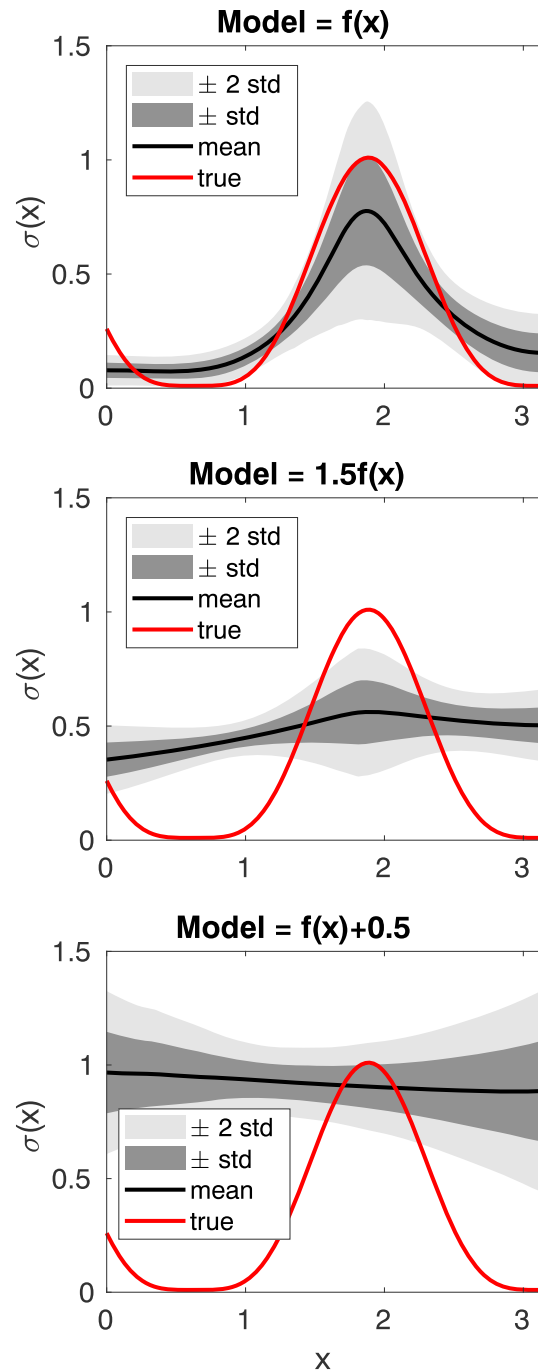


Figure 8. Results for the **W** data set. Values derived for the standard deviation σ , averaged over 200 independent runs (black), compared to the ground truth values used to generate the data (in red). The shaded gray area represents the confidence intervals of one (dark gray) and two (light gray) standard deviations calculated over the ensemble of 200 runs. (top) the correct mean function $f(x)$ is used for the model; (middle) a misspecified model that uses $1.5f(x)$ as mean; (bottom) a misspecified model that uses $f(x) + 0.5$ as mean.

On the other hand, when the model is misspecified (and the errors become non-Gaussian) the method appropriately assigns a larger uncertainty (middle and bottom panels). In particular, it is interesting that the discrepancy between the true variance and the one derived by this method is larger when the true variance is small. This is because in the regions with small (true) variance a misspecified model (mean function) causes a larger departure from Gaussianity. Since the method is designed to assign anyway a Gaussian probability density, it necessarily results in a larger uncertainty. Nevertheless, using the AR cost function as criterion to

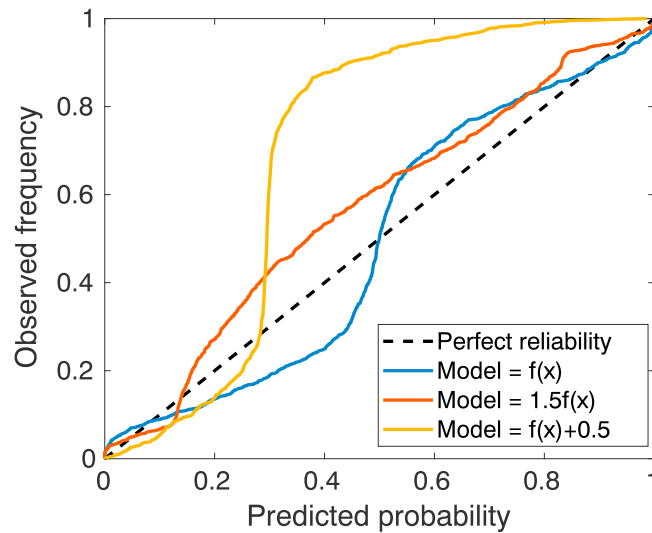


Figure 9. Reliability diagram for the method applied to the **W** data set. Blue, red, and yellow lines denote the observed frequency as function of the predicted probability, for the cases of correct mean function $f(x)$, and misspecified models $1.5f(x)$ and $f(x) + 0.5$, respectively. A perfect reliability is shown as a black dashed line.

derive the empirical variance will always results in an optimally calibrated model, meaning that ill-calibrated results are very unlikely, unless the underlying mean function is very off from the appropriate value. Figure 5 shows the reliability diagram for the three cases discussed (exact model and misspecified models). Once again, a reliability diagram represents, for any value of probability predicted for a given output, what is the actual observed frequency for that output (calculated on a large sample). A perfectly calibrated model results in a reliability diagram following the straight diagonal line (dashed black).

Not surprisingly, when we use the exact model as our mean function (blue line), the empirical variance derived with our method result in a perfectly calibrated model that indeed follow very closely the diagonal line (dashed black). When the model is misspecified (red and yellow lines), the method tries to achieve a trade-off between reliability and accuracy. The resulted reliability is still very good even though not perfect. It is very interesting that the reliability diagram can be used for our method to detect a misspecified mean function. Indeed, it is important to point out that, for the **G** data set, the model with additive error is worse

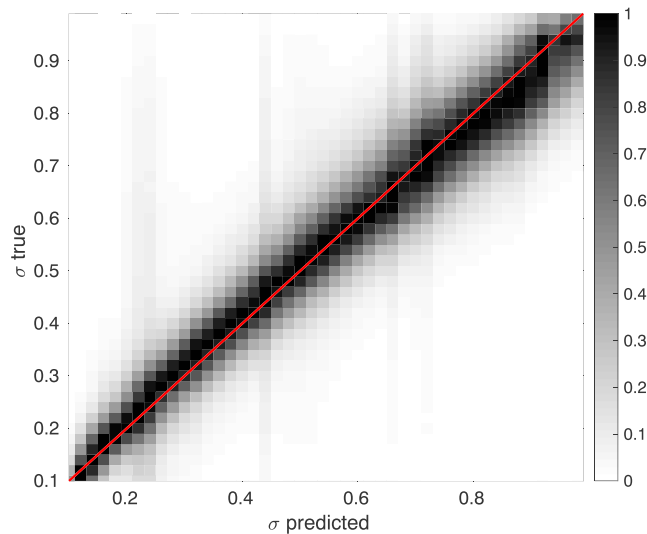


Figure 10. Probability density of the prediction versus real values of σ for the **5-D** data set. The red line denotes perfect prediction. The densities are normalized to have maximum value along each column equal to one. The 10,000,000 samples have been used to generate the plot (with a training set of 10,000 points).

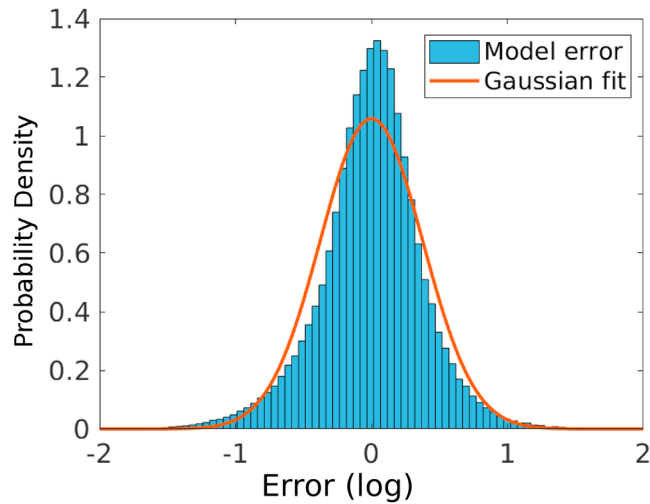


Figure 11. Density histogram of the DEN2D model errors (in logarithmic scale). The red line indicates a Gaussian fit.

than the one with multiplicative error, because $f(x)$ goes through zero in three points in the domain (hence, the multiplicative error plays no role in those points).

Results for the **Y** data sets are shown in Figures 6 and 7, with same format as previous Figures. Conclusions are very similar, with the main difference that the **Y** data set has a nonlinear true variance, which is harder to learn. Nevertheless, our method provides a good estimate of it. The **W** model is the most challenging, as shown in Figures 8 and 9. Here, a misspecification of the model becomes readily evident, producing almost constant variance and large errors in the reliability diagram.

For the **5-D** data set it is impractical to compare graphically the real and estimated $\sigma(\mathbf{x})$ in the five-dimensional domain. Instead, in Figure 10 we show the probability density of the real versus predicted values of the standard deviation. Values are normalized such that the maximum value in the colormap for any value of predicted σ is equal to one (i.e., along vertical lines). The red line shows a perfect prediction. The colormap has been generated by 10,000,000 points, while the model has been trained with 10,000 points only. For this case, we have used an exact mean function (equal to zero), in order to focus exclusively on the estimation of the variance. We believe that this is an excellent result for a very challenging task, given the sparsity of the training set, that shows the robustness of the method.

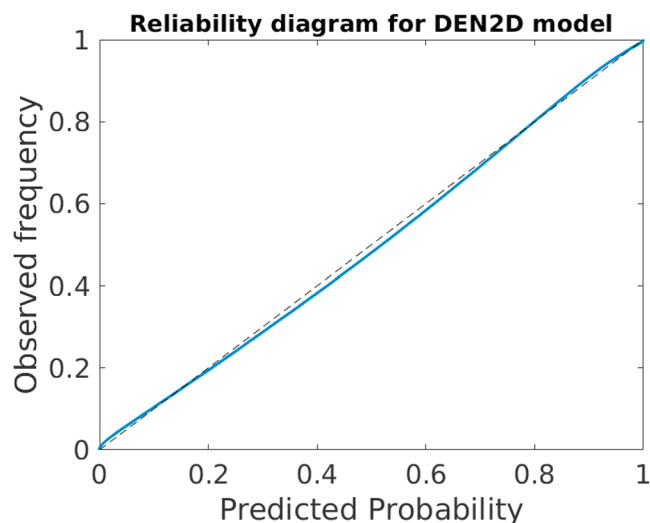


Figure 12. Reliability diagram of the probabilistic estimate of electron density, using the DEN2D model as mean function. The black dashed line indicates perfect reliability.

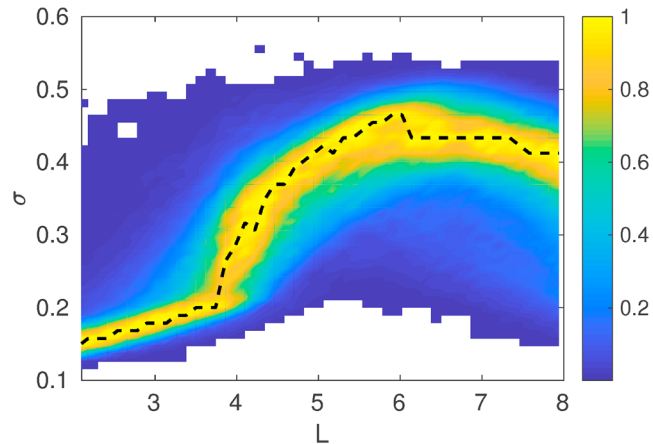


Figure 13. DEN2D model. Two-dimensional histogram of standard deviation σ versus L . The number of counts are normalized column-wise: the maximum for each value of L is equal to 1. The black dashed line follows the peak of the distribution as function of L .

4. Estimation of Electron Density in the Plasmasphere (DEN2D)

In this and the next section we show applications of our method that are relevant to Space Weather. The first example is the estimation of the electron plasma density in the plasmasphere. Chu et al. (2017) have devised a neural network model, DEN2D, that takes as inputs the time history of the SYM-H and AL geomagnetic indexes, and of $F_{10.7}$ (solar radio flux), and outputs the logarithm of the electron density at any location in the plasmasphere, as function of magnetic shell (L), and magnetic local time (MLT), at near-equatorial latitudes. DEN2D was trained and tested on about 400,000 events generated by 4 years of THEMIS data (June 2008 to December 2012), using 178 input attributes. It outputs the logarithmic value of the electron density. Obviously, DEN2D is a deterministic model, that outputs a single value for any given combinations of inputs. Hence, this model is very well posed for the method introduced in this paper. In this case we train our ANN to predict $\sigma(\mathbf{x})$ using the same input vector \mathbf{x} as DEN2D.

Moreover, a recent study performed to evaluate the propagation of uncertainties in radiation belt ensemble simulations has shown that the uncertainty in the electron density estimation carries most of the variance of the predicted electron fluxes (Camporeale et al., 2016). Therefore, the reduction of the uncertainty for electron density is a necessary step for developing reliable forecasts of electron fluxes. Figure 11 shows the distribution of the error of the NN output with respect to the true (log) electron density, calculated over the

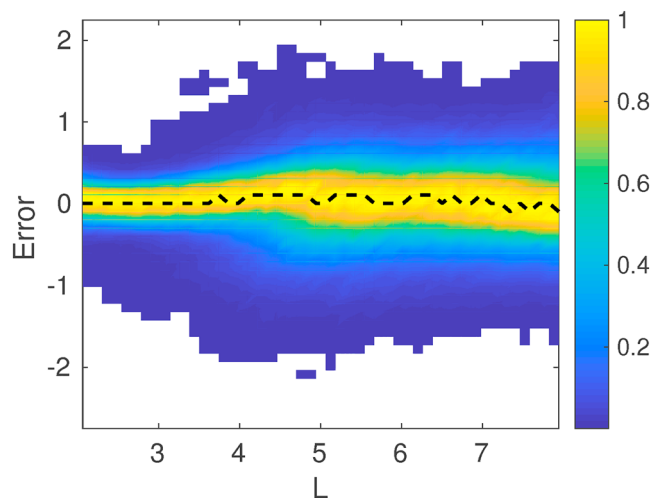


Figure 14. DEN2D model. Two-dimensional histogram of errors versus L . The number of counts are normalized column-wise: the maximum for each value of L is equal to 1. The black dashed line follows the peak of the distribution as function of L .

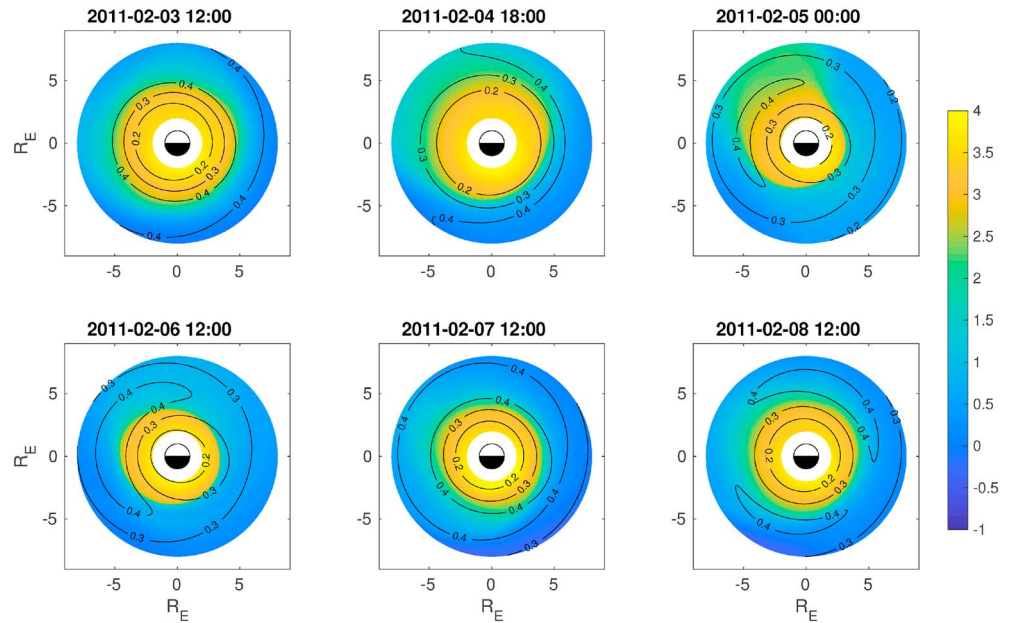


Figure 15. DEN2D model. A series of panels showing the estimated electron density (in color) and the associated standard deviation σ (as isolines) for the event of 4 February 2011, as function of L and MLT. The heat map represents the logarithm of the electron number density in el/cc (see Figure 6 in Chu et al., 2017).

whole data set. The superimposed red line shows a Gaussian fit to the distribution, which has a slighter larger variance. It is important, however, to keep in mind that our method does not assume that the model errors are normally distributed. Indeed, the method will try to enforce that the standardized errors η are Gaussian, which can be achieved even for non-Gaussian errors ϵ . This is well demonstrated by the reliability diagram, which is shown in Figure 12. Our method applied to the DEN2D model produces a probabilistic estimate of electron density that has remarkably good reliability. In other words, our method allows to estimate the Gaussian forecast, for any given distribution of errors, that produces an optimal trade-off between accuracy and reliability.

Once we have trained our model to estimate the standard deviation σ as function of the same inputs used in DEN2D, one can seek for evident relationship between σ and any of the inputs. This is in general non-trivial, given that the model takes 178 inputs. Indeed, the only evident correlation exists with the value of the magnetic L -shell. Figure 13 shows the two-dimensional histogram of L and σ . The number of counts are

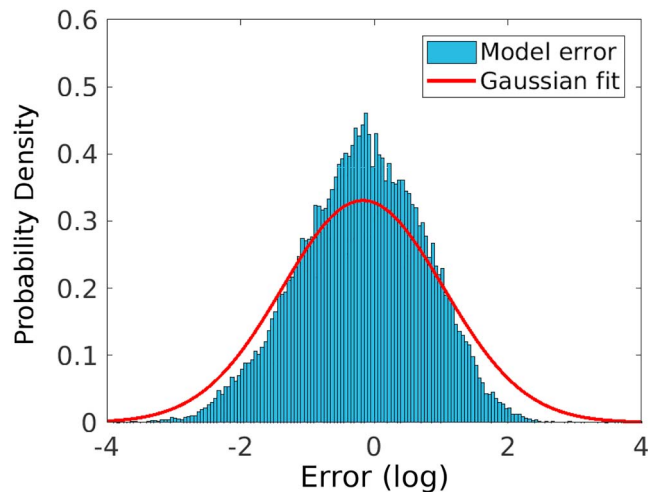


Figure 16. Density histogram of the errors of the model by Agapitov et al. (2018; in logarithmic scale). The Gaussian fit is shown in red.

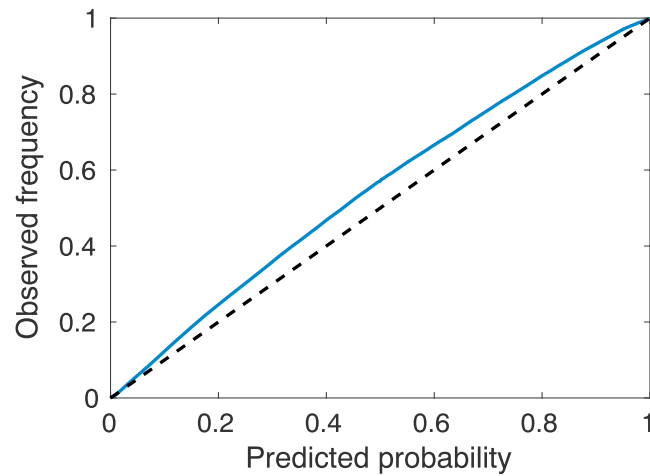


Figure 17. Reliability diagram for the probabilistic estimate of the chorus wave amplitude, based on the Agapitov et al. (2018) model. The black dashed line indicates perfect reliability.

normalized column-wise, that is for every value of L the maximum is set equal to 1. The black dashed line follows the maximum number of counts as function of L . The uncertainty of the density estimation increases with increasing L , reaching a maximum for $L \sim 6$. This is consistent with the distribution of errors, when ordered as function of L (Figure 3 in Chu et al., 2017), reproduced here in Figure 14, with the same format as before. Even though the mean value remains centered around zero, the spread of the errors increases with increasing L , hence resulting in larger uncertainties. We conclude this section by reproducing the result shown in Figure 6 of Chu et al. (2017), where the authors have applied the DEN2D model to the moderate storm of 4 February 2011. Figure 15 reproduces the estimated electron density at six different times, ranging from the quiet time before the storm to the recovery phase after the storm. The color bar indicates the (log) electron density. Superimposed to each image we show the isolines of the standard deviation calculated with our new method. It is interesting to notice how σ is as dynamic as the electron density. Being derived from the DEN2D model, the uncertainty is itself dependent on the time history of geomagnetic indexes and on geographical location.

5. Estimation of Chorus Wave Amplitude

Whistler-mode chorus waves play a crucial role for wave-particles interaction and particles scattering in the inner magnetosphere (Camporeale, 2015; Camporeale & Zimbardo, 2015; Thorne, 2010). The estimation of the wave amplitude is an important step in the calculation of pitch angle and energy diffusion by means of quasi-linear Fokker-Planck equations. Recently, Agapitov et al. (2018) have presented an empirical model to estimate the chorus wave amplitude and wave normal angle distribution, derived from the statistical analysis of Cluster and Van Allen Probes VLF measurements. The model takes as inputs the MLT, the magnetic latitude λ , the value of the L -shell, and the geomagnetic index K_p (or Dst Agapitov et al., 2015) providing

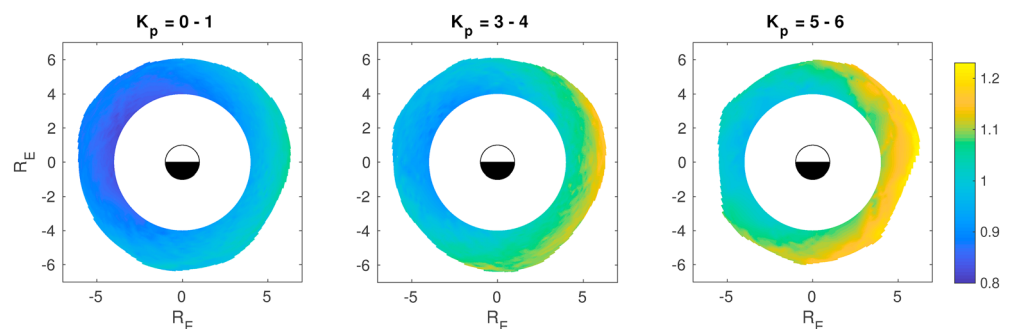


Figure 18. The standard deviation σ estimated for the Agapitov et al. (2018) model (chorus wave amplitude), for three different ranges of the geomagnetic index K_p , as a function of different magnetic local time and L shells.

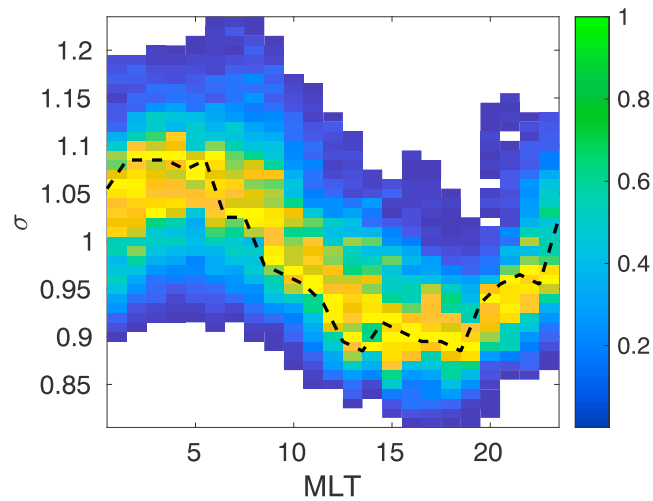


Figure 19. Agapitov et al. (2018) model (chorus wave amplitude). Two-dimensional histogram of standard deviation σ as function of MLT. The number of counts are normalized column-wise: the maximum for each value of MLT is equal to 1. MLT = magnetic local time.

the distribution of chorus wave amplitude and wave normal angle in the outer radiation belt (from the plasmapause to $L = 7$) for all MLT values in the latitudinal range from -45° to 45° . The model was developed in the polynomial form for chorus wave amplitude $B_w(\lambda, K_p) = a_{ij} \lambda^i K_p^j$ ($i, j = 0, 3$) with the coefficients calculated based on Cluster STAFF-SA measurements in 2001–2011 (Agapitov et al., 2015) and with the coefficients updated making use of the combined Cluster observations and the recent Van Allen Probes VLF measurements (Agapitov et al., 2018). In order to apply our new method to the model of Agapitov et al. (2018), we have produced an estimation of the chorus wave amplitude for the period 1 January 2015 to 30 December 2016, at 1-min resolution at the corresponding location of the Van Allen Probes spacecraft and the corresponding level of the geomagnetic activity. The ground truth value is taken directly from the Van Allen Probes EMFISIS observations. Note that this time interval was not included in the original training of the model. This produced a total of 213,937 data points for which the model error was calculated. Since the wave amplitude can range within two orders of magnitude, the errors are in log scale.

Figure 16 shows the histogram of the model error (computed as the difference between the logarithm of predictions and observations), compared with its Gaussian fit. Similarly to the model discussed in the previous section, this model does not yield errors that are exactly log-normal distributed. This, however, does not affect the goodness of our uncertainty estimate, in terms of accuracy and reliability. As previously, we train our algorithm to estimate the standard deviation $\sigma(\mathbf{x})$ using the same inputs as the original model. The reliability diagram, calculated over the entire data set, is shown in Figure 17. The largest mismatch, for a predicted probability equal to 50%, is about 7%, hence demonstrating that the model is very well calibrated. Figure 18 shows the heat map of the standard deviation σ at different locations $4 < L < 6.5$, and for different ranges of the geomagnetic index K_p (left panel: $K_p = [0 - 1]$; center panel: $K_p = [3 - 4]$; right panel: $K_p = [5 - 6]$). Not surprisingly, the largest uncertainties occur during storm-time, and in the pre-noon sector. Finally, Figure 19 shows the two-dimensional histogram of the standard deviation σ , as function of the MLT. A column-wise normalization is applied, such that the maximum value along a constant MLT is equal to one. Consistently with the previous Figure, the largest uncertainties occur for MLT in the range 0–10.

6. Conclusions

The estimation of uncertainties associated with the output of deterministic models is a key element of any forecasting method. The standard approach for evaluating such uncertainties is to rely on time-consuming ensemble simulations. In this paper, we have introduced a novel methodology to estimate uncertainties that does not require running costly ensembles. The guiding principle behind our method is that the uncertainty of the output distribution, here represented by the standard deviation of a Gaussian centered around the values predicted by the deterministic model, should produce a probabilistic forecast that is both accurate and reliable (well calibrated). We have introduced a cost function that encodes the trade-off between

accuracy and reliability for Gaussian distributions. The minimization of such AR cost function yields the optimal standard deviation $\sigma(\mathbf{x})$. The proposed method is ignorant with respect to the deterministic model it is applied to. In fact, it only requires the algebraic errors between predictions and true values, in order to be trained. A deep neural network is used to generate the unknown standard deviation for inputs other than the ones used for training. Even though this method is not Bayesian, it does make use of all possible information (in terms of available observations) to train the neural network and estimate the unknown standard deviation. Obviously, the larger the training set and its range of inputs, the more accurate will the result be.

Concerning the choice of restricting to Gaussian forecasts, its advantage is essentially in the fact that the AR cost function is analytically tractable. However, this might not be the best choice when the distribution of outputs is skewed. In that case, this method must be understood as estimating the optimal (in the sense of trade-off between accuracy and reliability) normal distribution for the available data. Future works will explore the extension of the method to non-Gaussian distributions.

We have shown experiments with synthetic data sets (for one- and five-dimensional examples), that demonstrate how our method is able to learn the underlying functional dependence of the standard deviation, which is, in real-world problems, unknown. These experiments also show how the method deals with cases when the underlying deterministic model contains a systematic error. In this cases, the reliability diagram represents a sanity check, indicating the presence of systematic errors. Indeed, it is understood that any problem with the underlying deterministic model is ultimately reflected in the reliability diagram.

Finally we have applied the method to two recently developed models, relevant to space weather: the estimation of the electron density in the plasmasphere (section 4) and of the chorus wave amplitude (section 5). In both cases, we use as inputs the same inputs employed in the original model. The probabilistic forecast produced with our method show excellent reliability diagrams, pinpointing the lack of a systematic bias in the original models.

Our code is available on the website www.mlinspaceweather.org and [zenodo.org](https://zenodo.org/doi/10.5281/zenodo.1485608) (doi:10.5281/zenodo.1485608) and we encourage the space weather community to produce probabilistic forecasts based on deterministic models, using our method. Finally, we point out that an interesting future extension to this method would be the case of multivariate outputs (in contrast to scalars). In that case, the definitions of CRPS and RS will need to account for covariances between variables.

Acknowledgments

E.C. is partially funded by NWO-Vidi grant 639.072.716. The work of O.A. was supported by NASA Grants NNX16AF85G S004, 80NSSC19K0264, and JHU/APL contract 922613 (RBSP-EFW). J.B. acknowledges NASA grants 80NSSC18K1227, NNX14AN85G and AFOSR grant FA9550-15-1-0158. All data and codes used to generate the results are available on the website www.mlinspaceweather.org and [zenodo.org](https://zenodo.org/doi/10.5281/zenodo.1485608) (doi:10.5281/zenodo.1485608).

References

- Agapitov, O., Artemyev, A., Mourenas, D., Mozer, F., & Krasnoselskikh, V. (2015). Empirical model of lower band chorus wave distribution in the outer radiation belt. *Journal of Geophysical Research: Space Physics*, *120*, 10–425. <https://doi.org/10.1002/2015JA021829>
- Agapitov, O., Mourenas, D., Artemyev, A., Mozer, F., Hospodarsky, G., Bonnell, J., & Krasnoselskikh, V. (2018). Synthetic empirical chorus wave model from combined Van Allen Probes and cluster statistics. *Journal of Geophysical Research: Space Physics*, *123*, 297–314. <https://doi.org/10.1002/2017JA024843>
- Aminalragia-Giamini, S., Sandberg, I., Papadimitriou, C., Daglis, I. A., & Jiggins, P. (2018). The virtual enhancements-solar proton event radiation (VESPER) model. *Journal of Space Weather and Space Climate*, *8*(A06), 10.
- Anderson, J. L. (1996). A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *Journal of Climate*, *9*(7), 1518–1530.
- Babuška, I., Nobile, F., & Tempone, R. (2007). A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, *45*(3), 1005–1034.
- Barnes, G., Leka, K., Schrijver, C., Colak, T., Qahwaji, R., Ashamari, O., et al. (2016). A comparison of flare forecasting methods. I. Results from the “all-clear” workshop. *The Astrophysical Journal*, *829*(2), 89.
- Barnes, G., Leka, K. D., Schumer, E. A., & Della-Rose, D. J. (2007). Probabilistic forecasting of solar flares from vector magnetogram data. *Space Weather*, *5*, S09002. <https://doi.org/10.1029/2007SW000317>
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. New York, NY: Oxford University Press.
- Bloomfield, D. S., Higgins, P. A., McAteer, R. J., & Gallagher, P. T. (2012). Toward reliable benchmarking of solar flare forecasting methods. *The Astrophysical Journal Letters*, *747*(2), L41.
- Branke, J., Deb, K., & Miettinen, K. (2008). *Multiobjective optimization: Interactive and evolutionary approaches* (Vol. 5252). Verlag Berlin Heidelberg: Springer Science & Business Media.
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, *78*(1), 1–3.
- Bröcker, J., & Smith, L. A. (2007). Scoring probabilistic forecasts: The importance of being proper. *Weather and Forecasting*, *22*(2), 382–388.
- Bussy-Virat, C. D., & Ridley, A. J. (2014). Predictions of the solar wind speed by the probability distribution function model. *Space Weather*, *12*, 337–353. <https://doi.org/10.1002/2014SW001051>
- Caffisch, R. E. (1998). Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, *7*, 1–49.
- Camporeale, E. (2015). Resonant and nonresonant whistlers-particle interaction in the radiation belts. *Geophysical Research Letters*, *42*, 3114–3121. <https://doi.org/10.1002/2015GL063874>
- Camporeale, E., Agnihotri, A., & Rutjes, C. (2017). Adaptive selection of sampling points for uncertainty quantification. *International Journal for Uncertainty Quantification*, *7*(4), 285–301.

- Camporeale, E., Shprits, Y., Chandorkar, M., Drozdov, A., & Wing, S. (2016). On the propagation of uncertainties in radiation belt simulations. *Space Weather*, *14*, 982–992. <https://doi.org/10.1002/2016SW001494>
- Camporeale, E., Wing, S., & Johnson, J. R. (2018). *Machine learning techniques for space weather*. Amsterdam, Netherlands: Elsevier.
- Camporeale, E., Wing, S., Johnson, J., Jackman, C., & McGranaghan, R. (2018). Space weather in the machine learning era: A multi-disciplinary approach. *Space Weather*, *16*, 2–4. <https://doi.org/10.1002/2017SW001775>
- Camporeale, E., & Zimbardo, G. (2015). Wave-particle interactions with parallel whistler waves: Nonlinear and time-dependent effects revealed by particle-in-cell simulations. *Physics of Plasmas*, *22*(9), 092104.
- Carè, A., & Camporeale, E. (2018). Regression. In *Machine learning techniques for space weather* (pp. 71–112). Amsterdam, Netherlands: Elsevier.
- Chandorkar, M., Camporeale, E., & Wing, S. (2017). Probabilistic forecasting of the disturbance storm time index: An autoregressive Gaussian process approach. *Space Weather*, *15*, 1004–1019. <https://doi.org/10.1002/2017SW001627>
- Chu, X., Bortnik, J., Li, W., Ma, Q., Angelopoulos, V., & Thorne, R. (2017). Erosion and refilling of the plasmasphere during a geomagnetic storm modeled by a neural network. *Journal of Geophysical Research: Space Physics*, *122*, 7118–7129. <https://doi.org/10.1002/2017JA023948>
- Gallagher, P. T., Moon, Y.-J., & Wang, H. (2002). Active-region monitoring and flare forecasting—I. Data processing and first results. *Solar Physics*, *209*(1), 171–183.
- Genz, A. (1984). Testing multidimensional integration routines. *Proc. Of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation* (pp. 81–94). New York, NY: Elsevier North-Holland, Inc.
- Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452.
- Gneiting, T., Balabdaoui, F., & Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *69*(2), 243–268.
- Gneiting, T., Raftery, A. E., Westveld III, A. H., & Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, *133*(5), 1098–1118.
- Goldberg, P. W., Williams, C. K., & Bishop, C. M. (1998). Regression with input-dependent noise: A Gaussian process treatment. In *Advances in neural information processing systems* (pp. 493–499). Colorado.
- Gruet, M., Chandorkar, M., Sicard, A., & Camporeale, E. (2018). Multiple hours ahead forecast of the Dst index using a combination of long short-term memory neural network and Gaussian process. *Space Weather*, *16*, 1882–1896. <https://doi.org/10.1029/2018SW001898>
- Hamill, T. M. (1997). Reliability diagrams for multicategory probabilistic forecasts. *Weather and Forecasting*, *12*(4), 736–741.
- Hamill, T. M. (2001). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, *129*(3), 550–560.
- Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, *15*(5), 559–570.
- Kahler, S. W., & Ling, A. (2015). Dynamic SEP event probability forecasts. *Space Weather*, *13*, 665–675. <https://doi.org/10.1002/2015SW001222>
- Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(3), 425–464.
- Kersting, K., Plagemann, C., Pfaff, P., & Burgard, W. (2007). Most likely heteroscedastic Gaussian process regression. In *Proceedings of the 24th international conference on Machine learning* (pp. 393–400). ACM, New York, NY.
- Lee, K., Moon, Y.-J., Lee, J.-Y., Lee, K.-S., & Na, H. (2012). Solar flare occurrence rate and probability in terms of the sunspot classification supplemented with sunspot area and its changes. *Solar Physics*, *281*(2), 639–650.
- Luhmann, J. G., Mays, M. L., Odstrcil, D., Li, Y., Bain, H., Lee, C. O., et al. (2017). Modeling solar energetic particle events using ENLIL heliosphere simulations. *Space Weather*, *15*, 934–954. <https://doi.org/10.1002/2017SW001617>
- Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, *22*(10), 1087–1096.
- McPherron, R. L., & Siscoe, G. (2004). Probabilistic forecasting of geomagnetic indices using solar wind air mass analysis. *Space Weather*, *2*, S01001. <https://doi.org/10.1029/2003SW000003>
- Miyoshi, Y., & Kataoka, R. (2008). Probabilistic space weather forecast of the relativistic electron flux enhancement at geosynchronous orbit. *Journal of Atmospheric and Solar-Terrestrial Physics*, *70*(2-4), 475–481.
- Möstl, C., Amerstorfer, T., Palmerio, E., Isavnin, A., Farrugia, C. J., Lowder, C., et al. (2017). Forward modeling of coronal mass ejection flux ropes in the inner heliosphere with 3DCORE. *Space Weather*, *16*, 216–229. <https://doi.org/10.1002/2017SW001735>
- Murphy, K. P. (2012). *Machine learning: A probabilistic perspective. Adaptive computation and machine learning*. Cambridge, MA: MIT Press.
- Murphy, A. H., & Winkler, R. L. (1992). Diagnostic verification of probability forecasts. *International Journal of Forecasting*, *7*(4), 435–455.
- Murray, S. A., Bingham, S., Sharpe, M., & Jackson, D. R. (2017). Flare forecasting at the Met Office Space Weather Operations Centre. *Space Weather*, *15*, 577–588. <https://doi.org/10.1002/2016SW001579>
- Napoletano, G., Forte, R., Del Moro, D., Pietropaolo, E., Giovannelli, L., & Berrilli, F. (2018). A probabilistic approach to the drag-based model. arXiv preprint arXiv:1801.04201.
- Nix, D. A., & Weigend, A. S. (1994). Learning local error bars for nonlinear regression. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Proceedings of the Neural Information Processing Systems (NIPS) 1994*. Denver, CO.
- Owens, M. J., & Riley, P. (2017). Probabilistic solar wind forecasting using large ensembles of near-Sun conditions with a simple one-dimensional “upwind” scheme. *Space Weather*, *15*, 1461–1474. <https://doi.org/10.1002/2017SW001679>
- Papaioannou, A., Anastasiadis, A., Sandberg, I., Georgoulis, M., Tsiropoula, G., Tziotziou, K., et al. (2015). A novel forecasting system for solar particle events and flares (forspef). *Journal of Physics: Conference Series*, *632*, 012075. IOP Publishing.
- Pinson, P., McSharry, P., & Madsen, H. (2010). Reliability diagrams for non-parametric density forecasts of continuous variables: Accounting for serial correlation. *Quarterly Journal of the Royal Meteorological Society*, *136*(646), 77–90.
- Prikryl, P., Jayachandran, P., Mushini, S., & Richardson, I. (2012). Toward the probabilistic forecasting of high-latitude GPS phase scintillation. *Space Weather*, *10*, S08005. <https://doi.org/10.1029/2012SW000800>
- Riley, P., & Love, J. J. (2016). Extreme geomagnetic storms: Probabilistic forecasts and their uncertainties. *Space Weather*, *15*, 53–64. <https://doi.org/10.1002/2016SW001470>
- Sharpe, M. A., & Murray, S. A. (2017). Verification of space weather forecasts issued by the Met Office Space Weather Operations Centre. *Space Weather*, *15*, 1383–1395. <https://doi.org/10.1002/2017SW001683>
- Smith, R. C. (2013). *Uncertainty quantification: Theory, implementation, and applications* (Vol. 12). Philadelphia: Siam.
- Thorne, R. M. (2010). Radiation belt dynamics: The importance of wave-particle interactions. *Geophysical Research Letters*, *37*, L22107. <https://doi.org/10.1029/2010GL044990>
- Wheatland, M. (2004). A Bayesian approach to solar flare prediction. *The Astrophysical Journal*, *609*(2), 1134.

- Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences* (Vol. 100). Oxford: Academic Press.
- Williams, P. M. (1996). Using neural networks to model conditional multivariate densities. *Neural Computation*, 8(4), 843–854.
- Xiu, D. (2010). *Numerical methods for stochastic computations: A spectral method approach*. Princeton, NJ: Princeton University Press.
- Xiu, D., & Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2), 619–644.
- Yuan, M., & Wahba, G. (2004). Doubly penalized likelihood estimator in heteroscedastic regression. *Statistics & Probability Letters*, 69(1), 11–20.
- Zhang, X., & Moldwin, M. B. (2014). Probabilistic forecasting analysis of geomagnetic indices for southward IMF events. *Space Weather*, 13, 130–140. <https://doi.org/10.1002/2014SW001113>