University of Exeter
College of Engineering, Mathematics, and Physical Sciences

Methods in machine learning for probabilistic modelling of environment, with applications in meteorology and geology

Submitted by **Charlie Kirkwood**
to the University of Exeter as a thesis for the degree of
Doctor of Philosophy in Mathematics
September, 2022

I certify that all material in this thesis which is not my own work has been identified and that any material that has previously been submitted and approved for the award of a degree by this or any other University has been acknowledged.

Signed: ....

Abstract

Earth scientists increasingly deal with ‘big data’. Where once we may have struggled to obtain a handful of relevant measurements, we now often have data being collected from multiple sources, on the ground, in the air, and from space. These observations are accumulating at a rate that far outpaces our ability to make sense of them using traditional methods with limited scalability (e.g., mental modelling, or trial-and-error improvement of process based models). The revolution in machine learning offers a new paradigm for modelling the environment: rather than focusing on tweaking every aspect of models developed from the top down based largely on prior knowledge, we now have the capability to instead set up more abstract machine learning systems that can ‘do the tweaking for us’ in order to learn models from the bottom up that can be considered optimal in terms of how well they agree with our (rapidly increasing number of) observations of reality, while still being guided by our prior beliefs.

In this thesis, with the help of spatial, temporal, and spatio-temporal examples in meteorology and geology, I present methods for probabilistic modelling of environmental variables using machine learning, and explore the considerations involved in developing and adopting these technologies, as well as the potential benefits they stand to bring, which include improved knowledge-acquisition and decision-making. In each application, the common theme is that we would like to learn predictive distributions for the variables of interest that are well-calibrated and as sharp as possible (i.e., to provide answers that are as precise as possible while remaining honest about their uncertainty). Achieving this requires the adoption of statistical approaches, but the volume and complexity of data available mean that scalability is an important factor — we can only realise the value of available data if it can be successfully incorporated into our models.

Acknowledgements

I would sincerely like to thank those who have formally supervised me during the course of this PhD: Theo Economou, Henry Odbert, Nicolas Pugeault, Gavin Shaddick, and Ben Lambert, as well as others from the University, the Met Office, and beyond who have helped to make the PhD such an enriching experience. Your wisdom and support has been invaluable. Special thanks must got to Theo, my primary supervisor, for his seemingly limitless knowledge, care, and patience and for the opportunity to embark on this PhD in the first place.

It has been a real privilege to be part of the statistical science group at the University of Exeter, and to work with applied scientists at the Met Office. Without a doubt this has been the most rewarding chapter of my life's somewhat unconventional curriculum so far. It cannot go unmentioned that the COVID-19 pandemic (March 2020 onwards; mostly normal again by May 2022) caused significant disruption to just about everything over the last couple of years. However, over time the community spirit and support among fellow PhD students and staff at the University and at the Met Office has proven to be an even stronger force of nature than the (less) novel (by the day) coronavirus.

No acknowledgement would be complete without thanking my friends and family who have given me so much inspiration, and my parents in particular for having gone beyond the call of duty for at least the last 33 years. Thank you!

Funding

This PhD was supported by a Doctoral Training Partnership studentship from the Engineering and Physical Sciences Research Council (EPSRC) [ref: 2071900], part of UK Research and Innovation (UKRI). In addition, the Met Office provided support in the form of an Industrial Cooperative Award in Science & Technology (CASE). I am extremely grateful for what both parties have enabled me to do.

Associated publications

Kirkwood, C., 2022. Geological mapping in the age of artificial intelligence. Feature article for *Geoscientist: magazine of the Geological Society of London*, Autumn 2022 issue, p.16-23 and front cover. doi.org/10.1144/geosci2022-023

Kirkwood, C., Economou, T., Pugeault, N. and Odbert, H., 2022. Bayesian deep learning for spatial interpolation in the presence of auxiliary information. *Mathematical Geosciences*, 54, p.507–531. doi.org/10.1007/s11004-021-09988-0

Kirkwood, C., Economou, T., Odbert, H. and Pugeault, N., 2022. A deep mixture density network for outlier-corrected interpolation of crowd-sourced weather data. *arXiv preprint* arXiv:2201.10544. doi.org/10.48550/arXiv.2201.10544

Kirkwood, C., Economou, T., Odbert, H. and Pugeault, N., 2021. A framework for probabilistic weather forecast post-processing across models and lead times using machine learning. *Philosophical Transactions of the Royal Society A*, 379(2194), p.20200099. doi.org/10.1098/rsta.2020.0099

Haupt, S.E., Chapman, W., Adams, S.V., **Kirkwood, C.**, Hosking, J.S., Robinson, N.H., Lerch, S. and Subramanian, A.C., 2021. Towards implementing artificial intelligence post-processing in weather and climate: proposed actions from the Oxford 2019 workshop. *Philosophical Transactions of the Royal Society A*, 379(2194), p.20200091. doi.org/10.1098/rsta.2020.0091

Kirkwood, C., Economou, T., Odbert, H. and Pugeault, N., 2021. Bayesian deep learning for large scale environmental data modelling. Poster presented at *the Alan Turing Institute's AI UK research showcase*. hdl.handle.net/10871/125229

Kirkwood, C., 2020. Deep covariate-learning: optimising information extraction from terrain texture for geostatistical modelling applications. *arXiv preprint* arXiv:2005.11194. doi.org/10.48550/arXiv.2005.11194

Contents

List of Figures

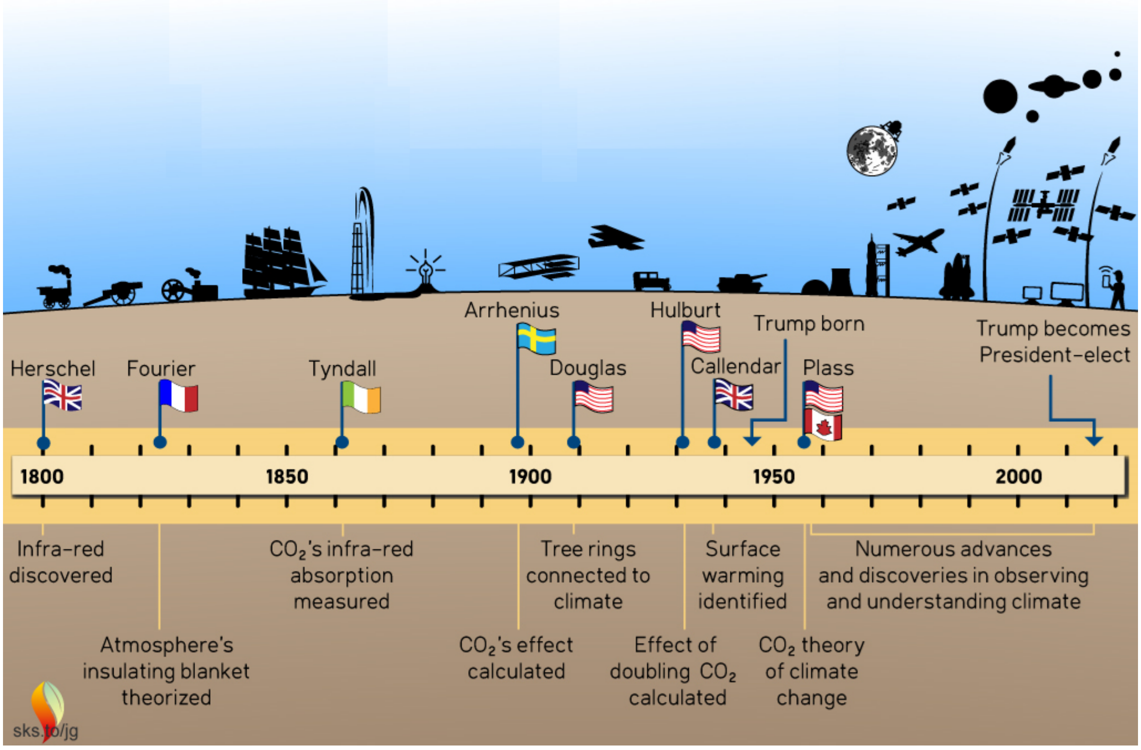
Chapter 1

Introduction

It is in the interest of any organism to gain knowledge of the environment in which it resides so that available resources can be utilised efficiently, and potential disasters mitigated against. Even a humble seedling could grow its root network and branches optimally if it could only collect, process, and act upon the necessary information about the state of its surroundings in a suitable way. Fortunately, as human beings we are comparatively gifted in our abilities to collect and process information, and have long been the natural masters of our surroundings as a result. Our natural intelligence — whilst not without flaws — has got us to where we are today.

However, we are in a time of great change: the finite nature of our planet is becoming ever harder to ignore as we progress into a self-inflicted climate crisis. Somewhat ironically, the burning of the same fossil fuels that have induced this crisis has also powered an explosion in technological progress (Figure 1.1) such that our ability to collect and process information digitally has evolved almost unrecognisably over the last century. Whilst our actions have been having significant negative impact on the Earth system, we have at least put ourselves in a position to recognise this.

In light of the climate crisis, the future success of humanity (and of life on Earth as we know it) now depends on our ability to act collaboratively and sensibly within the closed system of planet Earth. In some ways, as a species we now find ourselves in the same situation as the humble seedling: with nowhere to go, our future depends on how successfully we can collect and process information about our environment in order to act as close to optimally as possible. While a plant's control systems benefit from millions of years of evolutionary refinement, the same cannot be said of humanity's collective control systems: at the species scale, we've got no 'prior experience' of navigating a climate crisis — our success depends entirely on the systems we develop 'in the moment' to enable the actions we take here and now. Fortunately, we now have on our side a rapidly evolving body of information-processing technology that potentially already includes the precursors of true artificial intelligence.



Arrhenius
Hulburt
Trump born
Trump becomes President-elect
Herschel
Fourier
Tyndall
Douglas
Callendar
Plass
1800
1850
1900
1950
2000
Infra-red discovered
CO_2's infra-red absorption measured
Tree rings connected to climate
Surface warming identified
Numerous advances and discoveries in observing and understanding climate
Atmosphere's insulating blanket theorized
CO_2's effect calculated
Effect of doubling CO_2 calculated
CO_2 theory of climate change
sks.to/jg

Figure 1.1: Timeline of climate science, and general technology since 1800. It is staggering to consider how rapidly technology has evolved over the last two centuries (and decades), and yet Pierre Simon Laplace had recognised by 1812 that "The most important questions of life... are indeed for the most part only problems of probability". It is perhaps only now, with the recent data science revolution, that we are fully appreciating just how right he was. Image from Sceptical Science, sks.to

In this thesis, we explore some possible machine learning approaches for distilling environmental knowledge — in the form of predictive probability distributions of environmental variables — from large volumes of data, which may include outputs of existing models (e.g. from numerical weather prediction). We do so with the ambition that such approaches can contribute to humanity's ability to 'master our surroundings' in a 21st century sense: by improving the quality with which we can model the current and future states of our environment, we can enable ourselves — as a collective — to make decisions that are closer-to-optimal for achieving our goals. These decisions needn't all be revolutionary; even our day-to-day interactions with the environment can benefit from improved efficiency and reduced wastage, or loss, by having better information. By developing probabilistic modelling systems that can strive to learn predictive distributions of environmental variables which are as sharp (or precise) as possible subject to calibration (honesty about uncertainty), we put ourselves in a position to make decisions that are as precise as possible while remaining robust to uncertainty.

1.1 Thesis content and structure

This thesis covers probabilistic modelling examples in temporal, spatial, and spatio-temporal contexts, using data-sets and use-cases from both the UK Met Office and the British Geological Survey, both of whom deal with some of the UK's most complex environmental modelling challenges in meteorology and geology respectively. Motivation for the work has largely come from the need within weather forecasting to be able to assimilate and digest for specific purposes the wealth of information provided by increasing numbers of increasingly complex physics-based numerical weather prediction models. However, there is much commonality between the challenges of post-processing gridded forecasts from numerical weather prediction models (in relation to point-sampled ground-truth observations), and of incorporating information from gridded auxiliary variables (e.g. terrain elevation data, satellite imagery) into environmental models in general. As a result, this thesis explores this area of commonality and proposes architectures for probabilistic machine learning systems that are suitable for dealing with these conceptually similar data integration problems.

Specifically, Chapter 3 presents a Quantile Regression Forest (QRF) approach to site-specific post-processing of numerical weather forecasts using the example of Met Office road surface temperature forecasting (or MORST). This serves to illustrate the complexity of numerical model output data that is available nowadays to help inform predictions of weather variables even on a site-by-site basis, and therefore conveys the necessity for developing machine learning systems that can deal with this scale of data. We then expand our thinking to address spatial modelling in chapters 4 and 5, in which we develop a Bayesian neural network based approach for combining spatial interpolation and computer vision capabilities in order to automatically extract relevant information from gridded auxiliary variables, such as those provided by satellite imagery or numerical weather prediction grids (although we actually develop the concept using national geochemical survey observations accompanied by a national-scale terrain elevation grid). In Chapter 6 we address spatio-temporal modelling, this time building on the Bayesian deep learning architecture of chapters 4 and 5 to develop a deep mixture model for robust spatio-temporal interpolation of crowd sourced weather observations.

The initial goal of this trajectory of research was to develop a suitable probabilistic machine learning system for full spatio-temporal post-processing of numerical weather forecasts. As things stand, this ‘ultimate’ full post-processing application is not demonstrated in this thesis, in part due to the disruption of the coronavirus pandemic and related difficulties in assembling the necessary data-sets, but also due to the increasingly crowded nature of the AI weather forecast post-processing research space (largely since Rasp and Lerch (2018)). Therefore, in an effort to adjust the project design in the light of unforeseen problems, the final chapter of this thesis — Chapter 7 — explores the implications of our research to the discipline of geological mapping. This is an area in which the author has background expertise, but which has remained largely isolated from mathematical modelling efforts over the decades (and centuries) since its conception. As Chapter 7 explains, the Bayesian deep learning approaches developed during this PhD, inspired by the challenges and solutions of weather forecasting, may turn out to hold the key to a long overdue probabilistic revolution in geological mapping practice.

All the core research chapters of this thesis (chapters 3 to 7) have been published along the way (albeit that chapters 4 and 6 are currently just publicly available pre-prints on arXiv). In addition, some of the content of Chapter 2’s literature review has been published as part of an internationally-coauthored review on AI for weather forecast post-processing, but as I was not the lead author of that publication I have included material from it only sparingly. For a full list of publications associated with this thesis, see the previous section ‘Associated publications’.

1.2 Modelling philosophy

1.2.1 Environmental modelling and probability

While the pressing issues for our species have changed over time, human beings have always modelled their environment to some degree. Being skilful at predicting where the nearest heard of edible creatures might currently be found, or when and where the most nutritious fruit will crop, would always have provided an advantage — and such basic mental modelling abilities clearly predate us as a species by many eons. In fact, even bacteria live their lives according to (simple) environmental models: by the process of chemotaxis the movement of bacteria is biased towards environments which contain higher concentrations of beneficial chemicals (e.g. food), or lower concentrations of toxic chemicals (Wadhams and Armitage, 2004). Implicit behind this evolved behaviour is a crude model of the environment which says that good things can be found, and bad things avoided, simply by following chemical gradients.

We can imagine that any agent, or collection of agents, with a set of goals can increase their chances of achieving these goals by improving their environmental knowledge. Even the simple bacterium who follows chemical gradients could in theory, if given perfect information about its environment, make a direct beeline to the chemically-optimal location (potentially going against beneficial gradients to get there — the local optimum may not be the global optimum) and thereby live out a happy and easy life. As human beings in the face of a climate crisis, if we had perfect information about the current and future states of the Earth (and how our different possible actions would affect this), then we could simply make the optimal set of actions to mitigate or even perhaps overt the crisis. The problem, of course, is that we can never have perfect information: not everything is directly observable everywhere, and even things that can be observed can only be observed with some level of error. This introduces uncertainty into our knowledge of even the current state of the Earth. This uncertainty is then amplified when it comes to forecasting future states, as even in a deterministic universe small changes in initial state can lead to enormous changes in future states (Lorenz, 1963). This is summarised by Edward Lorenz's comment on chaos: "when the present determines the future, but the approximate present does not approximately determine the future".

In the 1700s, Pierre Simon Laplace pondered what would be possible to predict given sufficient 'intellect' if we could observe the present state of the universe with absolute certainty, in an idea known as Laplace's demon (Laplace, 1814): "We may regard the present state of the universe as the effect of its past and the cause of its future. An intellect which at a certain moment would know all forces that set nature in motion, and all positions of all items of which nature is composed, if this intellect were also vast enough to submit these data to analysis, it would embrace in a single formula the movements of the greatest bodies of the universe and those of the tiniest atom; for such an intellect nothing would be uncertain and the future just like the past would be present before its eyes".

The content of this PhD thesis generally concerns the prediction of environmental variables through space and through time. Suppose we have some variable, y, whose value varies with position in space, s, and in time, t. Given Laplace's demon, we would be capable of making predictions of y — here denoted $\hat{y}$ — which are perfect and without error, such that

$$\hat{y}_{s,t} = y_{s,t} \tag{1.1}$$

and therefore our predictions at any position in space and time would perfectly match reality.

If we had Laplace's demon at our disposal, and could use it to foresee *with certainty* the outcomes of different decisions that we could make (if we are free to make decisions in such a deterministic universe), then it would be relatively straightforward to simply make whatever decisions best achieve our chosen goals. However, there are several reasons why Laplace's demon is unrealistic, even as a hypothetical thought experiment: Laplace himself knew that "while the human mind offers, in the perfection which it has been able to give to astronomy, a feeble idea of this intelligence ... it will always be infinitely removed [from it]".

More recently, Gödel's incompleteness theorem (Gödel, 1931) showed that "Any consistent formal system F within which a certain amount of elementary arithmetic can be carried out is incomplete; i.e., there are statements of the language of F which can neither be proved nor disproved in F" (Raatikainen et al., 2020) and hence that there are inherent limitations to what can be know in any formal system, and thus Laplace's demon is logically impossible to achieve even in a deterministic universe. In addition, chaos theory (Lorenz, 1963) tells us that, even in deterministic systems, slight differences in initial conditions can evolve into considerably different states, imposing a practical time-horizon on the predictability of such systems. Therefore, regardless of the power of our computers the predictions we make will never be reliably perfect and error free, i.e. the reality for us is that

$$\hat{y}_{s,t} \neq y_{s,t}. \tag{1.2}$$

Since Laplace's demon will always be unachievable, our predictions will always be imperfect and uncertain, it therefore makes sense to adopt a probabilistic approach in order to issue predictions in the form of probability statements, rather than as (almost certainly incorrect) absolutes. The alternative, of simply issuing absolute predictions that we hope are 'close enough' to reality, can be dangerous because users of those predictions cannot take into account the risks associated with reality turning out to be different from the prediction. Instead, we can aim to issue a prediction as a distribution

$$p(y_{s,t}) \tag{1.3}$$

in which the probabilities (or probability densities, if y is continuous) that we assign to different possible values of $y_{s,t}$ represent our uncertainty about what the true value of $y_{s,t}$ is in reality.

Given that $y_{s,t}$ will have some true value in reality, it could be argued that issuing a probabilistic prediction is also ‘incorrect’, because there is only one reality that we can observe in practice, not a distribution of realities. Ideally we would be able to predict the true value of $y_{s,t}$ without error (i.e., as in Laplace’s demon), but this is unachievable — there will always be uncertainty due to the impossibility of observing everything (and even more-so to observe everything without error). The key benefit of issuing a probability distribution as a prediction is that doing so enables us to convey the uncertainty associated with the prediction, such that users of the prediction — downstream decision makers — can account for the probabilities of encountering different scenarios in reality, therefore enabling them to mitigate appropriately against adverse outcomes.

1.2.2 Frequentist vs Bayesian methods

In order to issue our probabilistic prediction $p(y_{s,t})$ for all points in space and time in which we might be interested, it makes sense to predict a probability density for y conditional on some inputs or predictor variables, $x_{s,t}$, which describe each location or position in space and time, and to do so according to some model m. Our predictions would therefore take the form

$$p(y|x_{s,t}, m). \tag{1.4}$$

Changing the model, m, will change the resultant predictive distribution, $p(y|x_{s,t})$, and, as model choice is subjective, so too is the resultant predictive distribution. Clearly it is important to use the 'right' m — in an ideal world we could use Laplace's Demon, but since that is not an option, there are two opposing schools of thought about the best approach; Frequentism and Bayesianism.

In the frequentist approach, we tend to aim to find (or discover, learn) a model, $\hat{m}$, which has the highest likelihood (Le Cam, 1990) of having generated all of the data we observe, $\boldsymbol{y}$. This is to say, by some procedure of model selection and parameter optimisation, arrive at a 'best' model, $\hat{m}$, which maximises $p(\boldsymbol{y}|m)$ so that

$$\hat{m} = \underset{m}{\operatorname{argmax}}\, p(\boldsymbol{y}|m). \tag{1.5}$$

If the likelihood, $p(\boldsymbol{y}|m)$, is sufficiently high then we may be inclined to believe in the suitability of our chosen model, $\hat{m}$. However, we should beware that

$$p(\boldsymbol{y}|m) \neq p(m|\boldsymbol{y}) \tag{1.6}$$

and therefore the model that maximises the likelihood is not necessarily the most probable model given the data. To calculate this 'inverse probability', $p(m|\boldsymbol{y})$, requires adopting a Bayesian approach (Bayes, 1763; Laplace, 1820) using Bayes' theorem

$$p(m|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|m)p(m)}{p(\boldsymbol{y})}. \tag{1.7}$$

where $p(m|\boldsymbol{y})$ is the posterior probability of the model given the data, $p(\boldsymbol{y}|m)$ is the probability of the data given the model (the likelihood), $p(m)$ is the prior probability of the model, and $p(\boldsymbol{y})$ is the evidence, or marginal likelihood, equal to $\int_m p(m)p(\boldsymbol{y}|m)dm$ in the case that the space of possible models is continuous, or $\sum_m p(m)p(\boldsymbol{y}|m)$ in the case that the space of possible models is discrete.

Omitting $p(\boldsymbol{y})$, the (subjective, user specified) posterior probability of the model given the data is proportional to the prior probability of the model multiplied by the likelihood:

$$p(m|\boldsymbol{y}) \propto p(\boldsymbol{y}|m)p(m). \tag{1.8}$$

Using Bayes theorem unlocks the capability of basing model selection and parameter optimisation on maximising $p(m|\boldsymbol{y})$ rather than $p(\boldsymbol{y}|m)$, which is arguably more logical. This is known as maximum a posteriori estimation (MAP), which returns a 'best' model, $\hat{m}$, as

$$\hat{m} = \underset{m}{\operatorname{argmax}}\, p(m|\boldsymbol{y}). \tag{1.9}$$

MAP estimation is similar to maximum likelihood estimation in the sense that it aims to find a single 'best' model, but it incorporates the prior beliefs of the modeller in the form of $p(m)$.

The real power of the Bayesian approach however comes in modelling the full distribution of possible models $p(m|\boldsymbol{y})$. That is to say, rather than committing to a single 'best' model (whether based on maximal likelihood, maximal posterior probability, or some other criteria) we instead entertain — and use to predict — all possible models according to their posterior probability $p(m|\boldsymbol{y})$. This posterior distribution of possible models becomes particularly useful when making predictions out-of-sample, where the spread between predictions from different models that could each reasonably fit our in-sample data, $\boldsymbol{y}$, is likely to increase. Using a single best model for out-of-sample prediction is therefore likely to be inherently overconfident; because the resultant predictions would be based on the strong assumption that we know for sure that our single model is the right one.

The Bayesian approach provides predictions in the form of the posterior predictive distribution (here assuming a continuous space of models)

$$p(Y_{s,t}|x_{s,t},\boldsymbol{y}) = \int_m p(Y_{s,t}|x_{s,t},m)p(m|\boldsymbol{y})dm. \tag{1.10}$$

which we can approximate using Monte Carlo integration by repeating a two step process of first sampling a possible model from the posterior distribution $p(m|\boldsymbol{y})$, and then, given the model, sampling from the likelihood distribution $p(Y_{s,t}|x_{s,t},m)$ — so sampling a possible value for our target variable at different positions in space and time. In this manner, the Bayesian posterior predictive distribution captures our uncertainty about the form of the 'true' model, by predicting from a distribution over possible models according to their posterior probability.

By contrast, predictions from a maximum likelihood, or MAP, single-model approach simply take the form

$$p(Y_{s,t}|x_{s,t},\boldsymbol{y}) = p(Y_{s,t}|x_{s,t},\hat{m}) \tag{1.11}$$

where $\hat{m}$ is our single 'best' model with either maximal likelihood $p(\boldsymbol{y}|m)$ or maximal posterior probability $p(m|\boldsymbol{y})$, depending on the chosen estimation. The single model can only account for uncertainty using its likelihood function. While this is useful for representing uncertainty in the data — or aleatoric uncertainty — it fails to capture uncertainty in the model itself — or epistemic uncertainty. Intuitively, making predictions using only a single model in situations where multiple models are possible will lead to predictions being underdispersive, so that they do not convey the full spread of environmental scenarios expected to be observed in reality. By modelling a distribution over possible models, the Bayesian approach captures both aleatoric *and* epistemic uncertainty.

The Bayesian definition of probability is as a degree of belief (Lee, 1989), and so in Bayesian modelling the posterior distribution over parameters, or possible models, represents our belief (between 0 and 1) in the parameter values, and the models they represent, being true. The posterior probabilities, or probability densities, are proportional to our prior belief in those parameter values multiplied by their likelihood. Conversely, the frequentist definition of probability is as a relative frequency of events in a long run of repeated experiments or trials, but for situations where the experiment can only happen once (e.g., the outcome of an election) this definition of probability makes little sense.

Even in situations where we are not interested in making fully probabilistic predictions, there are still benefits to adopting a Bayesian approach. For example. if we are interested in predicting just the mean of our target variable through space and time, in the Bayesian approach this prediction becomes more stable, because the discrepancies between the predictions of different possible models are averaged out (i.e., as is the case for the MAP estimate). In this sense, the Bayesian approach behaves similarly to regularisation in frequentist methods (and in fact equivalently in some instances, for example ridge regression can be derived as the MAP estimate resulting from using an independent Gaussian prior; Hastie et al., 2009; Vladimirova et al., 2019), which is used to avoid 'overfitting' to the training dataset at the expensive of the model's ability to predict out-of-sample observations — which is what matters in the real world.

1.2.3 Model performance on out-of-sample data

We should beware that any single model with a high number of parameters may perform well on the sample of data on which it is trained (the training sample; Figure 1.2), while potentially performing much less well on out-of-sample data that it has not seen during training (the test sample; Figure 1.2). This phenomenon is referred to as over-fitting, because the model has 'over-fit' to the training data, at the expense of its ability to predict out-of-sample observations well. This can similarly be considered as over-fitting to noise, at the expense of fitting to signal; a model that is more complex than the true data generating process will not generalise well to new observations (whereas a model that is less complex than the true data generating process may perform about equally well on in-sample and out-of-sample data, despite not capturing the full complexity of the problem).

Due to the risk of overfitting, simply maximising the likelihood with regard to the training data, $p(\boldsymbol{y}_{train}|m)$, is therefore no guarantee of arriving at a model with maximal (or even high) probability of generating the test observations, $p(\boldsymbol{y}_{test}|m)$. In order to reduce the discrepancy between the model's performance on in-sample and out-of-sample data (such that in-sample performance metrics become a

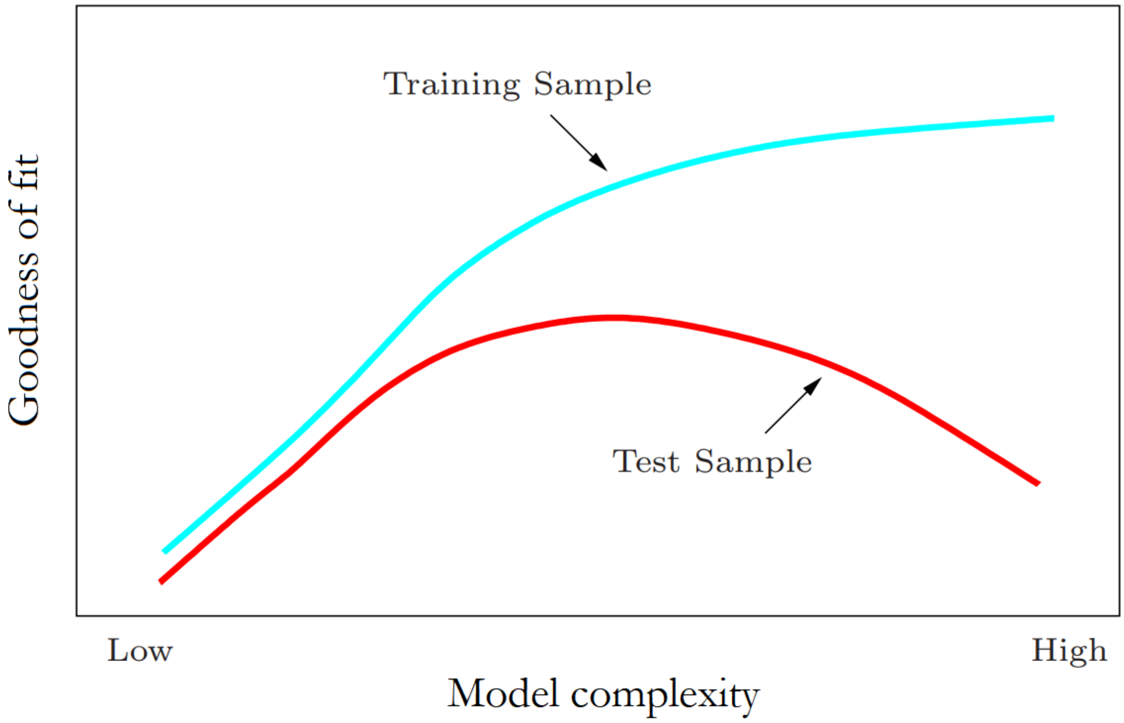


Figure 1.2: Goodness of model fit as a function of model complexity for in-sample (training) and out-of-sample (test) data. Adapted from Hastie et al. (2009).

better indicator of out-of-sample performance), we can use a penalised likelihood approach, or regularisation, where we aim to maximise not the likelihood itself but the likelihood plus some penalty for complexity.

For example, ridge regression includes a quadratic penalty on the values of a model's parameters, θ, (excluding the intercept, so that the result does not depend on the origin of the observations, $\boldsymbol{y}$). The solution to ridge-penalised maximum likelihood then becomes

$$\hat{m} = \underset{m}{\operatorname{argmax}}\, p(\boldsymbol{y}|m) - \lambda \sum_{\substack{j=1 \\ j \neq intercept}}^{M} \theta_j^2. \tag{1.12}$$

where $\lambda \geq 0$ is a user-specified complexity parameter that controls how heavily complexity is penalised. The larger the value of λ the more the coefficients are shrunk towards zero, thereby inducing a 'simpler' model with lesser tendency to overfit.

While complexity penalties help guide us away from models that overfit the training data, it still makes sense to assess the performance of our models on out-of-sample data rather than in-sample data, because this is the performance that matters in the real world — the ability to make good predictions in contexts that have not been seen during training.

If ample data is available, we can assess out-of-sample performance by splitting our dataset into separate folds, then training the model on different folds to those used to assess the model's performance. We adopt this approach for the applications in this thesis, for which we generally have ample data available. Specifically, we have tended to adopt a dataset splitting approach, whereby a dataset is split into three separate folds — train, validate, and test — at a ratio of approximately 80:10:10 from which a model is trained on the 'train' set (80% of total), while being tweaked to maximise performance on the 'validate' set (10% of total), with final model fit metrics being reported for the 'test' set only (another 10% of total). Doing so largely nullifies the risk of overfitting, on the assumptions that model performance on the 'validate' set should be similar to that on the 'test' set, and that performance on the 'test' set provides adequate approximation of the 'out of sample' performance of the model in general.

There is some controversy over the validity of randomly splitting data into training and testing folds in situations where the resultant folds may be non-independent. For example, folds produced by random sampling of spatial data may exhibit spatial autocorrelation, thus making them more similar, and easier to predict, than folds consisting of data points at geographically opposite ends of a study area. This may therefore lead to an optimistic bias when evaluating the performance of spatial models using randomly sampled folds.

Block cross-validation, in which folds are sampled as spatially distinct blocks, rather than spatially-interdispersed random samples, has been suggested as a solution (Roberts et al., 2017). However, through experimentation Wadoux et al. (2021) found 'that spatial cross-validation strategies resulted in a grossly pessimistic map accuracy assessment, and gave no improvement over standard cross-validation'. Therefore, despite the potential issue of non-independence within randomly sampled folds of spatial data, evaluation using randomly sampled test observations seems a viable approach, given that we generally wish to evaluate the performance of a map-producing model within the spatial extent of data on which it was trained (i.e, to assess its interpolation performance).

If we instead wish to assess a spatial model's extrapolation performance, e.g., the ability of a model trained using UK data to make predictions in France, then spatial cross-validation would provide a more suitable proxy for this, but as Wadoux et al. (2021) found, it would produce a pessimistic view of the interpolation performance of the model. Therefore it seems most appropriate to evaluate model performance using held out test data, or folds, that best reflect the intended real-world use case of the model. For that reason, we have evaluating the spatial models within this thesis using randomly sampled test sets.

In situations where less data is available, so that splitting the dataset becomes overly wasteful, an alternative approach to avoid selection of an overfitted model is to use a criterion that again involves penalising the likelihood for model complexity. For example, the Akaike Information Criterion (Akaike, 1974) chooses the model for which the quantity

$$\ln p(\boldsymbol{y}|m) - M \tag{1.13}$$

is largest, where M is the number of adjustable parameters in the model. However, in practice this tends to lead to favouring overly simple models (Bishop and Nasrabadi, 2006), and is therefore perhaps less useful selection criterion for modern high-M machine learning methods.

1.2.4 Assessing the quality of a probabilistic prediction

There are a number of ways that we might choose to assess the quality of a probabilistic prediction $p(y_{s,t})$. All the approaches mentioned here can be considered different aspects of 'model checking', which is to some extent a subjective process for sense-checking model output and measuring performance on a range of metrics that we deem to be important for assessing a given model's output.

1.2.4.1 Keep it simple: pretend it's deterministic

Thinking deterministically, which is perhaps the simplest place to start, we might hope that the mean of our predictive distribution, $E(y_{s,t})$, will be close to observed values of $y_{s,t}$. It would be typical to assess this using the mean squared error,

$$MSE = \sum_{s=1}^{n} (y_{s,t} - E(y_{s,t}))^2, \tag{1.14}$$

or, in the original units of the variable y, the root mean squared error,

$$RMSE = \sqrt{\sum_{s=1}^{n} (y_{s,t} - E(y_{s,t}))^2} \tag{1.15}$$

both of which will be zero if our the mean of our predictive distribution, $E(y_{s,t})$, perfectly matches observations of y (as Laplace's demon would achieve).

1.2.4.2 Calibration and sharpness

However, given that quantifying uncertainty is the key benefit of issuing probabilistic predictions, it makes sense to assess the quality of this uncertainty quantification, rather than just the closeness of the predictive mean to reality. To do so, we can think in terms of the *calibration* of our predictive distribution. To quote Donald Rubin: "A Bayesian is calibrated if his [or her] probability statements have their asserted coverage in repeated experience ... Consequently, the probabilities attached to Bayesian statements do have frequency interpretations that tie the statements to verifiable real world events" (Rubin, 1984).

Gneiting, Balabdaoui and Raftery (2007) propose a framework for defining calibration in terms of three modes: *probabilistic calibration*, *exceedance calibration*, and *marginal calibration*, each of which corresponds to a different measure of comparison between a predictive probability distribution (or forecast) F and nature's own probability distribution G from which observations are drawn. Considering times $t = 1, 2, ...$, nature picks a probability distribution G_t and the forecaster or modeller chooses a probabilistic predictive distribution F_t. While we talk here in terms of forecasts and outcomes for different times, t, the concepts are just as applicable to predictions and outcomes for different points in space, s.

Following Gneiting, Balabdaoui and Raftery (2007), let $(F_t)_t = 1,2,...$ and $(G_t)_t = 1,2,...$ denote sequences of continuous and strictly increasing cumulative distribution functions (CDFs) at each timestep t. We can think of $(G_t)_t = 1,2,...$ as the true data-generating process and of $(F_t)_t = 1,2,...$ as the associated sequence of probabilistic forecasts. Gneiting, Balabdaoui and Raftery (2007) then define the three modes of calibration as follows:

a) The sequence $(F_t)_t = 1,2,...$ is *probabilistically calibrated* relative to the sequence $(G_t)_t = 1,2,...$ if

$$\frac{1}{T}\sum_{t=1}^{T} G_t\big(F_t^{-1}(p)\big) \to p \quad \text{for all } p \in (0,1). \tag{1.16}$$

b) The sequence $(F_t)_t = 1,2,...$ is *exceedance calibrated* relative to the sequence $(G_t)_t = 1,2,...$ if

$$\frac{1}{T}\sum_{t=1}^{T} G_t^{-1}\big(F_t(x)\big) \to x \quad \text{for all } x \in \mathbb{R}. \tag{1.17}$$

c) The sequence $(F_t)_t = 1,2,...$ is *marginally calibrated* relative to the sequence $(G_t)_t = 1,2,...$ if the limits

$$\bar{G}(x) = \lim_{T\to\infty}\left\{\frac{1}{T}\sum_{t=1}^{T} G_t(x)\right\}$$

and

$$\bar{F}(x) = \lim_{T\to\infty}\left\{\frac{1}{T}\sum_{t=1}^{T} F_t(x)\right\}$$

exist and equal each other for all $x \in \mathbb{R}$, and if the common limit distribution places all mass on finite values.

If all of the above calibration modes are satisfied (i.e., it is probabilistically calibrated, exceedance calibrated, and marginally calibrated) then the sequence $(F_t)_t = 1,2,...$ is said to be *strongly calibrated* relative to $(G_t)_t = 1,2,....$

These notions of calibration all require averaging over units of time (or positions in space). Thus the calibration of a forecast, or of the predictive distribution of a model in general, does not necessarily indicate the accuracy with which predictions are made at the scale of individual points; rather it indicates how well probabilistic predictions correspond to reality on average. In theory, the best situation we can possibly hope for is that

$$F_t = G_t \quad \text{for all } t \tag{1.18}$$

so that our predictive distribution F_t is equal to nature's proposal distribution G_t from which observations x_t are drawn. Given that in reality we cannot directly observe G_t (only the observations that are drawn from it) (Gneiting, Balabdaoui and Raftery, 2007) propose the paradigm of maximising the *sharpness* of the predictive distribution subject to calibration — where sharpness simply refers to the concentration of the predictive distribution — as a reasonable way to get as close as possible to the ideal forecast. The more concentrated the predictive distributions are, the sharper the forecasts, and the sharper the better, subject to calibration.

We can illustrate this principle with the following simple example: Figure 1.3 and Figure 1.4 both show identical observations of some variable y made at different values of some covariate x (which in this case runs from 0 to 365, and is perhaps most easily imagined to be day of the year). In Figure 1.3, the shaded ribbon depicts a mean-centred 95% prediction interval for a model which does not account for the effect of x on y; as a result the predictive distribution remains constant for all values of x, but it is probabilistically calibrated, with observations falling within the 95% prediction interval 95% of the time (and for all other intervals in the appropriate corresponding proportions). In Figure 1.4, the shaded ribbon again depicts a mean-centred 95% prediction interval, but in this case for a model whose predictive distribution does account for the effect of x on y. As a result the predictive distribution varies with x, in this simulated example it equals the true

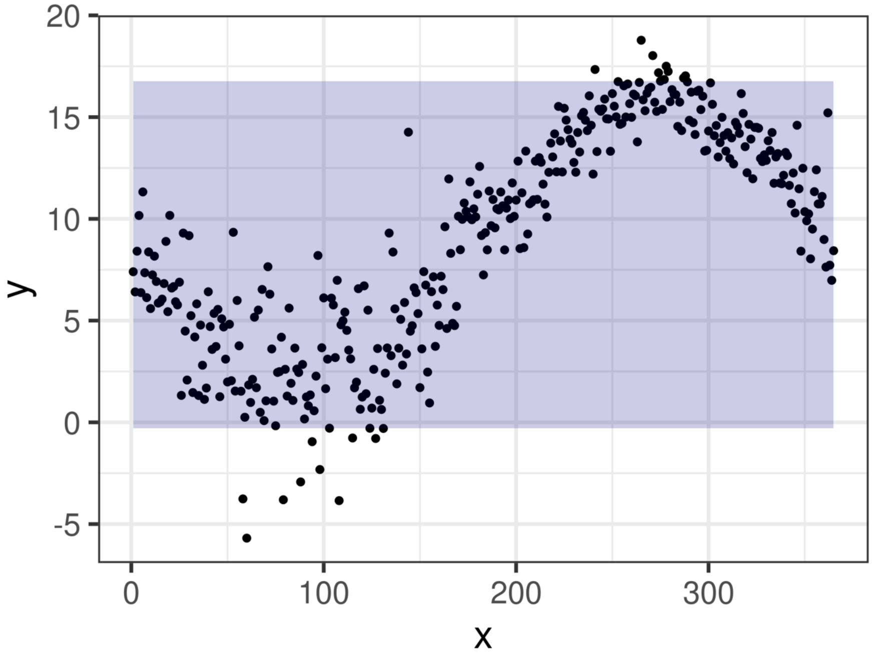


Figure 1.3: Simulated observations of some variable y plotted against a covariate x, which could be imagined to be day of the year. The grey ribbon depicts a mean-centred 95% prediction interval for a model which does not take account of the effect of x on y, and is thus not very sharp.

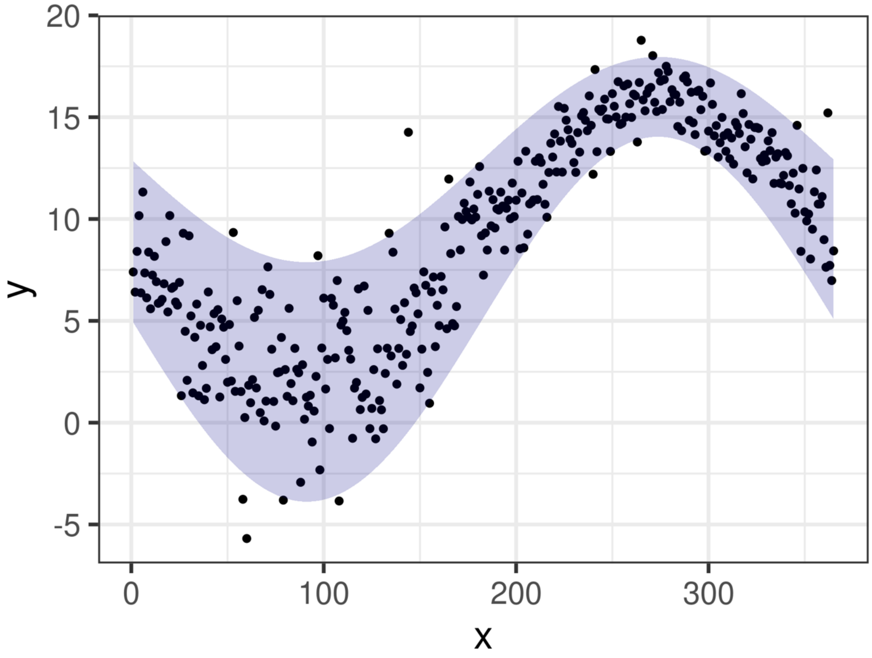


Figure 1.4: Simulated observations of some variable y plotted against a covariate x, which could be imagined to be day of the year. The grey ribbon depicts a mean-centred 95% prediction interval for a model which takes account of the effect of x on y, and is thus sharper (and is in fact equal to the true data generating distribution in this case, making it the ‘ideal forecast’).

distribution from which the observations are generated, i.e. the model depicted in Figure 1.4 is the ideal forecast. Gneiting, Balabdaoui and Raftery (2007) conject that following the paradigm of maximising the sharpness of the predictive distribution subject to calibration leads us towards this ideal forecast by favouring models that are as precise as possible while remaining honest about uncertainty.

1.2.4.3 Implications for decision making

As Tim Palmer (2017) explains: “A deterministic forecast provides the user with the simple decision strategy: take protective action when the event E is forecast. By contrast, a probabilistic forecast provides a more refined strategy: take protective action when the risk of the event pL $<$ C. Here p is an estimate of the probability of E, based on the ensemble forecast [and L is the loss caused by the event, and C is the cost of taking action to avoid this loss]. For example, according to this strategy, the user should almost always take protective action if C $\ll$ L. However, when C is comparable with L, he or she should only take protective action when it is almost certain that E will occur.” Our ability to take action to prevent loss therefore depends upon the ‘quality’ of probabilities issued.

Intuitively it seems sensible that maximising the sharpness of the predictive distribution subject to calibration will be helpful when it comes to decision making, because doing so narrows down as far as possible (within the constraints of our ignorance, i.e. while remaining honest about uncertainty) the spread of scenarios that we can expect to encounter. As a result, well-calibrated and sharp models are able to reveal opportunities to exploit gains / minimise losses which less sharp models will fail to capture. We illustrate this below with a fictitiously simple example. More on decision theory can be found in e.g., Pratt, Raiffa, Schlaifer et al. (1995) and Smith (2022).

If we imagine that y is road surface temperature, if (as in Figure 1.5) our predictive distribution fails to capture the time-of-year effect we might believe that the probability of sub-zero temperatures was a constant 0.05 per day. If gritting

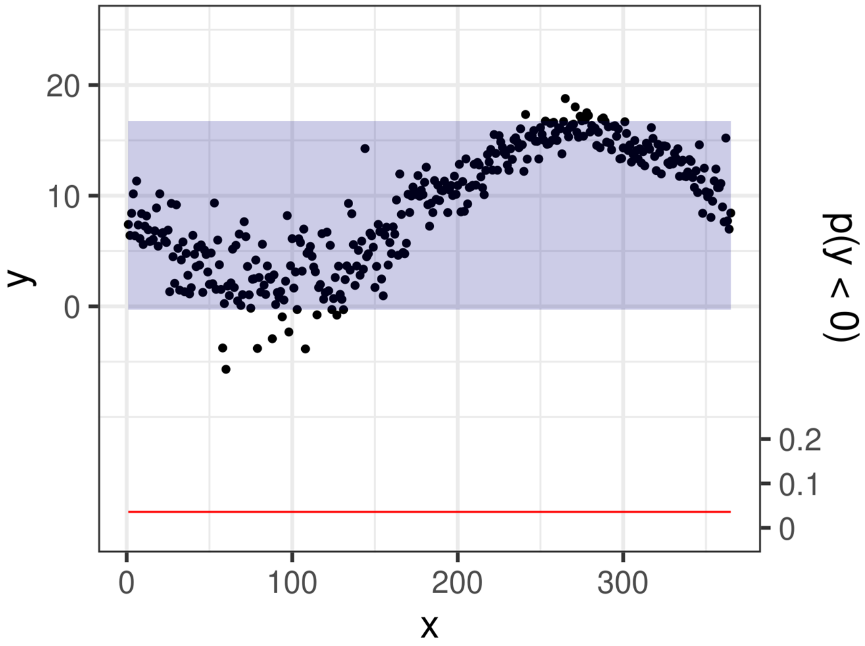


Figure 1.5: Simulated observations of some variable y plotted against a covariate x, which could be imagined to be day of the year. The grey ribbon depicts a mean-centred 95% prediction interval for a model which does not take account of the effect of x on y, and is thus not very sharp. The red line shows the probability of y falling below zero given the model - in this case it is constant.

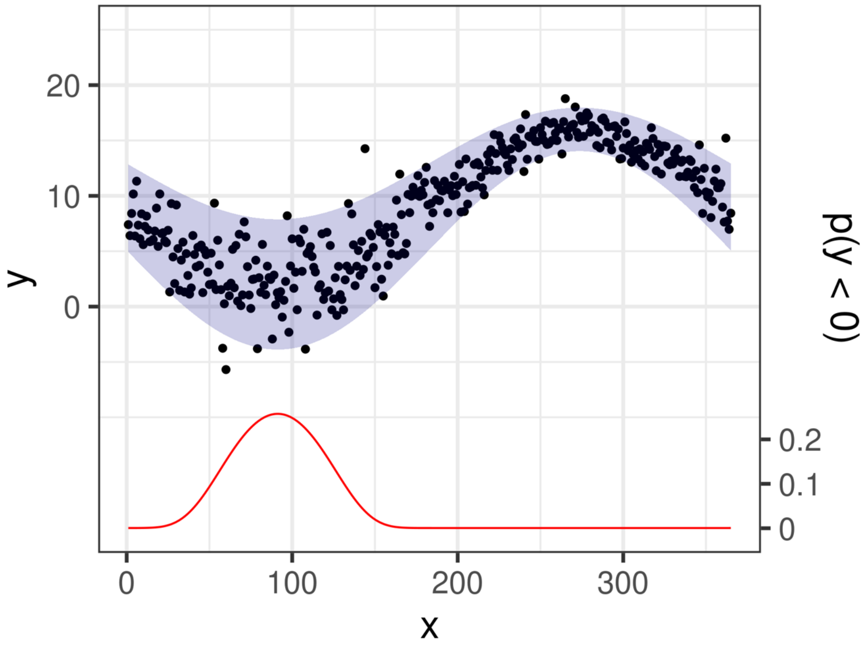


Figure 1.6: Simulated observations of some variable y plotted against a covariate x, which could be imagined to be day of the year. The grey ribbon depicts a mean-centred 95% prediction interval for a model which takes account of the effect of x on y, and is thus sharper (and is in fact equal to the true data generating distribution in this case, making it the ‘ideal forecast’). The red line shows the probability of y falling below zero given the model - the sharpness of the model focuses the probability mass around the period when p(y $<$ 0) is highest in reality.

costs £10 000 per day, but untreated frozen roads cost £100 000 per day in damage (through accidents, road deterioration etc), then although our expected daily loss to frozen roads is 0.05 x 100 000 = £5 000, it would not be worth spending £10 000 a day on gritting to prevent this, because over time our loss to constant expenditure would simply match the loss caused by occasional frozen roads. In this situation, as a result of being too ignorant to effectively mitigate our losses, we lose an expected £1.825m (£5000 x 365) per year – it's simply not viable to mitigate against the risk given how little we know.

If, on the other hand, we have a well calibrated and sharp predictive distribution (bottom figure), we can be more precise and efficient in how we mitigate risks (and pursue rewards). With the benefit of the 'ideal forecast' (whose predictive distribution is closer to, or in this case matches, the true data generating distribution; Figure 1.6) we find that our highest risk day has $p(y < 0) = 0.25$, giving an expected loss for the day of 0.25 x 100 000 = £25 000. Therefore, by gritting on this day at a cost of £10 000 we can expect to save £15 000 on average (£25 000 – 10 000 = £15 000).

If we choose to grit on every day that we expect on average to save more than the cost of gritting (i.e, using cost-loss decision making; Murphy, 1966; Palmer and Richardson, 2014), we will end up gritting the 80 highest-risk days at a cost of £800 000, but this will reduce our expected loss from damage from £1.825m to just £200 000; a reduction of £1.625m. With an annual gritting cost of £800 000, and an annual expected damage loss of £200 000, our total expected loss for the year is now £1m instead of the original £1.825m; utilising the sharper (and still well-calibrated) model for decision making saves us £825 000 per year.

While the above example concerns prediction of some variable y through time, the benefits of sharpness are perhaps best visualised via the prediction of variables through space; i.e. in the modelling of spatial maps. A previous study of my own — " machine learning approach to geochemical mapping" Kirkwood et al. (2016a) — compared spatial interpolation using ordinary kriging to a regression-on-covariates approach using a quantile regression forest for the purpose of predicting chemical element concentrations in soils. This application illustrates the difference that sharpness can make to map outputs, with the regression-on-covariates machine learning approach accounting for the effects of auxiliary variables in order to increase the precision of predictions over a purely spatial autocorrelation based interpolator.

Increasing the sharpness of spatial and spatiotemporal model output has the potential to reveal opportunities (to maximise gain or minimise loss) that less sharp models would not reveal. For a meteorological example, knowing precisely when and where rain will cause surface flooding would allow for precise mitigation measures to be rolled out — such as evacuation of vulnerable people and setup of flood barriers — in all the places that this is required (but none that it is not, thereby maximally reducing loss for minimal cost). For a geological example, knowing precisely where critical metals are concentrated allows us to optimise how we extract them. As explained in subsection 1.2.1, as the world faces a climate crisis the importance of maximising the efficiency and minimising the waste of all our activities is amplified, and sharp well-calibrated environmental models are the key to interacting with the Earth with the levels of precision that the situation demands.

1.2.4.4 Proper scoring rules

Having made the case that it is desirable to construct probabilistic models whose predictive distributions achieve maximal sharpness subject to calibration, it is important to have suitable metrics by which this can be assessed. We can do so using scoring rules, and in particular, proper scoring rules. As Gneiting and Raftery (2007) explain “scoring rules assess the quality of probabilistic forecasts, by assigning a numerical score based on the predictive distribution and on the event or value that materialises. A scoring rule is proper if the forecaster maximises the expected score for an observation drawn from the distribution G if he or she issues the probabilistic forecast $F = G$, rather than $F \neq G$. It is strictly proper if the maximum is unique. In prediction problems, proper scoring rules encourage the

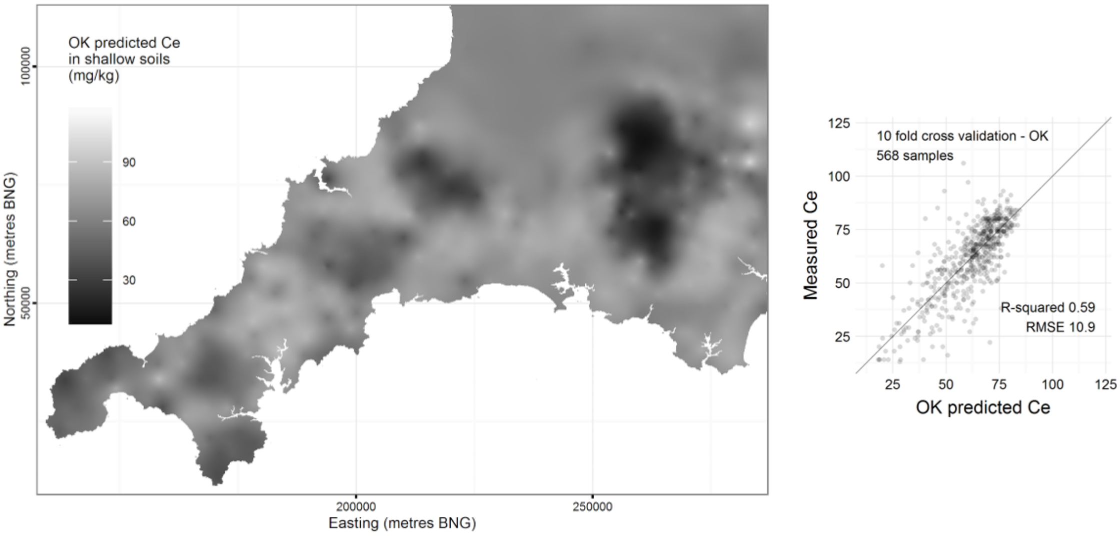


Figure 1.7: Soil cerium concentrations in south west England modelled using ordinary kriging (OK), such that predictions are made on the basis of spatial autocorrelation alone, resulting in a 'blunt' predictive distribution which lacks precision (note, calibration not shown, but for the sake of illustration we assume that both maps are similarly well calibrated). From Kirkwood et al. (2016a).

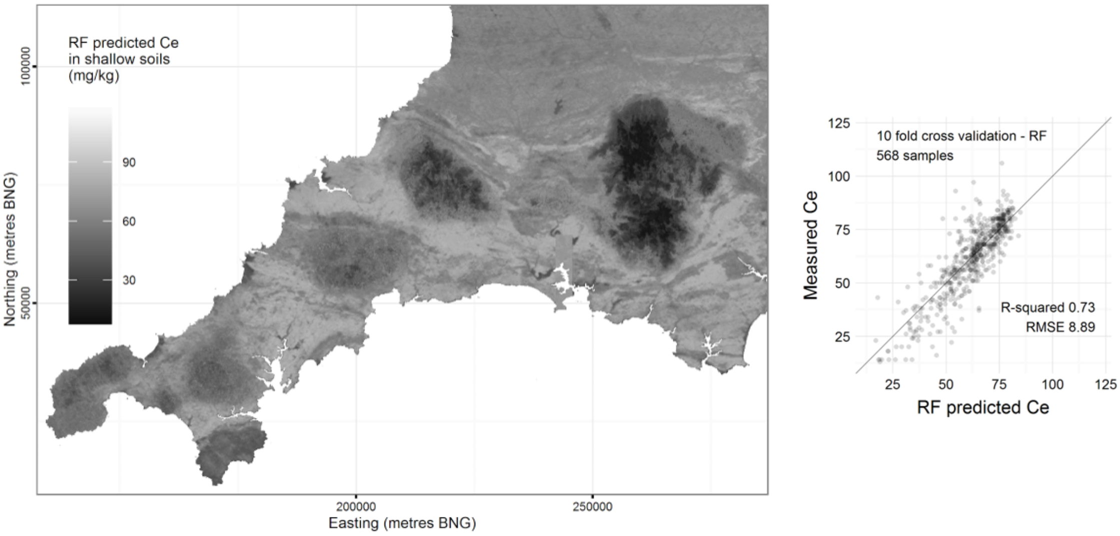


Figure 1.8: Soil cerium concentrations in south west England modelled using a regression-on-covariates quantile regression forest approach (RF), such that predictions are dependent on the values of auxiliary variables at each location, resulting in a sharp predictive distribution (note, calibration not shown here, but for the sake of illustration we assume both maps are similarly well calibrated). From Kirkwood et al. (2016a).

forecaster to make careful assessments and to be honest".

In this thesis, we use the Continuous Ranked Probability Score (CRPS) as our (strictly) proper scoring rule,

$$CRPS = \int_x \left(P(y_{pred} \leq x) - P(y_{obs} \leq x)\right)^2 dx \tag{1.19}$$

which is the mean squared error (MSE) between the predicted and observed cumulative distribution functions (CDFs; Figure 1.9), such that a CRPS of zero means that the distribution of predictions and the distribution of observations is the same. In the language of Gneiting and Raftery (2007), this would mean that the forecaster has successfully issued the probabilistic forecast F (which is the same distribution from which observations are drawn).

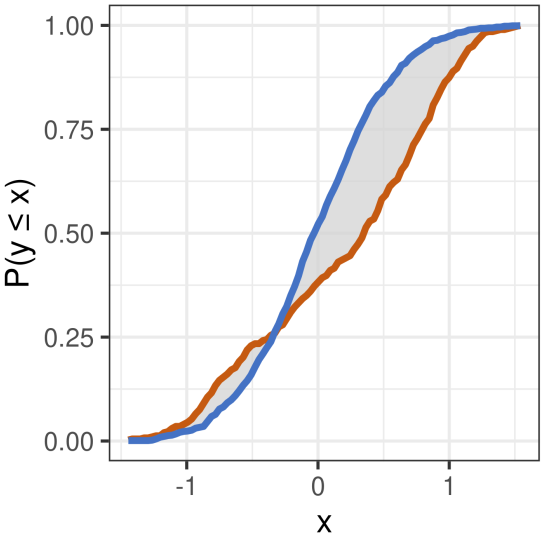


Figure 1.9: Illustration of the Continuous Ranked Probability Score (CRPS). The curves show the cumulative distribution functions (CDFS) of a set of predictions (blue) and observations (orange). The CRPS measures the mean squared erorr (MSE) between the two CDFS, and would be zero if the two distributions match, i.e. if the distribution issued by the forecaster matches the distribution from which observations are drawn in reality.

The Continuous Ranked Probability *Skill* Score (CRPSS) is favoured by the European Centre for Medium Range Weather Forecasting (ECMWF) for comparisons between the quality of forecasts. It enables the CRPS of a forecast to be compared with the CRPS of some benchmark (for example climatology; the historic distribution of weather conditions for a given location at a given time of year).

$$CRPSS = 1 - CRPS_{forecast}/CRPS_{benchmark} \tag{1.20}$$

such that: when CRPSS = 1 the forecast has perfect skill compared to the benchmark / climatology; when CRPSS = 0 the forecast has no skill compared to the benchmark / climatology; when CRPSS = a negative value the forecast is less accurate than the benchmark / climatology.

As Palmer and Richardson (2014) explain through a similar demonstration of cost-loss decision making as Section 1.2.4.3 in this thesis, “the CRPSS is simply a normalised measure of the potential economic value of a forecast system (typically an ensemble forecasting system) for a family of users which span the possible range of cost-loss ratios and for weather events which span the range of possible rainfall thresholds. That is to say, CRPSS is perhaps the simplest single measure of the overall value of a forecasting system for decision-making! ... If a perfect forecasting system (the oracle) would save the European economy €100 billion, then a forecasting system with a CRPSS of 0.2 would realise €20 billion of this saving. Doubling the CRPSS from 0.1 to 0.2 means doubling the savings from €10 billion to €20 billion.”

Murphy (1993) makes the case that “forecasts possess no intrinsic value. They acquire value through their ability to influence decisions made by users of the forecasts”. These words are equally valid in all contexts of making probabilistic predictions; the value of proper scoring rules like the CRPS — because of their relation to decision making — is not limited to the context of forecasting atmospheric variables, despite that being the use-case in which they have gained popularity.

1.2.4.5 Higher-order properties

The concepts of calibration and sharpness allow us to compare a forecast (or predictive) distribution to the true distribution (or observations drawn from it) on a point-wise basis. This enables the average quality of the predictive distribution *at each point of comparison* to be assessed, but disregards assessment of any covariance that may exist between the point-wise values of individual samples from which the predictive distribution is composed.

We can illustrate this invariance to the covariance of predictive samples by simulating samples from a multivariate normal distribution with two different covariance matrices. Firstly, by simulating from a multivariate normal distribution with diagonal covariance (i.e., all values of the covariance matrix are zero apart from the diagonal — which dictates just the variance — which is set to 5 in this case) we recover samples whose values at each point in space are independent (Figure 1.10). Secondly, by exchanging the diagonal covariance matrix for a Matern(5,4) covariance matrix, we introduce covariance to the simulates samples (Figure 1.11).

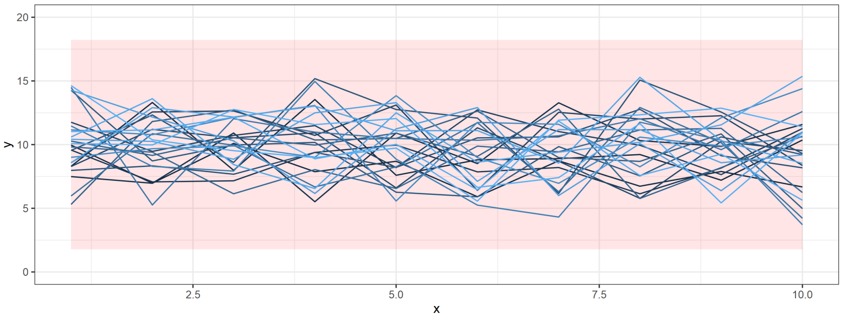


Figure 1.10: Illustration of samples drawn from a 10-dimensional multivariate normal distribution with diagonal covariance matrix such that the values of each sample are independent from point to point (e.g. across space or time). The shaded area indicates a 90% prediction interval.

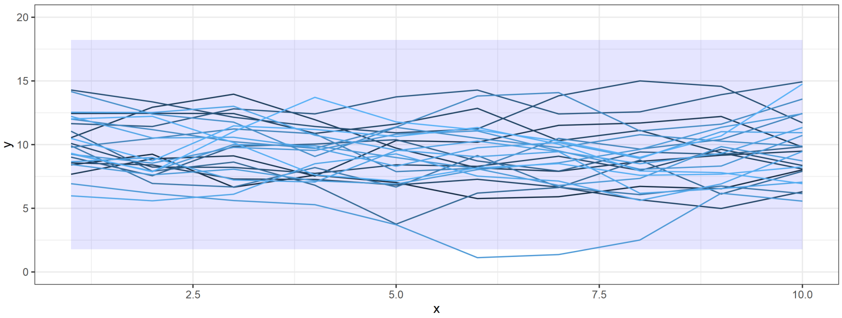


Figure 1.11: Illustration of samples drawn from a 10-dimensional multivariate normal distribution with Matern(5,4) covariance matrix, such that the values of each sample are dependent from point to point (e.g. across space or time); and exhibit spatial or temporal autocorrelation. The shaded area indicates a 90% prediction interval.

In both cases the calibration and sharpness of the predictive distribution is the same (see for example that both Figure 1.10 and Figure 1.11 show a shaded ribbon indicating a mean-centred 90% prediction interval, and that these are identical) but the covariance of the samples of which they are composed is very different. When does this matter? If the output of the model is only of interest on a point-wise basis, then the covariance of its individual samples is not important. However, if the output of the model is to be used to provide information about areas or volumes (in space and / or time), then the 'realism' of the covariance of the predictive samples becomes important because of its effect on the variance of (areal) integrals of the predictand, as shown in Figure 1.12 and Figure 1.13.

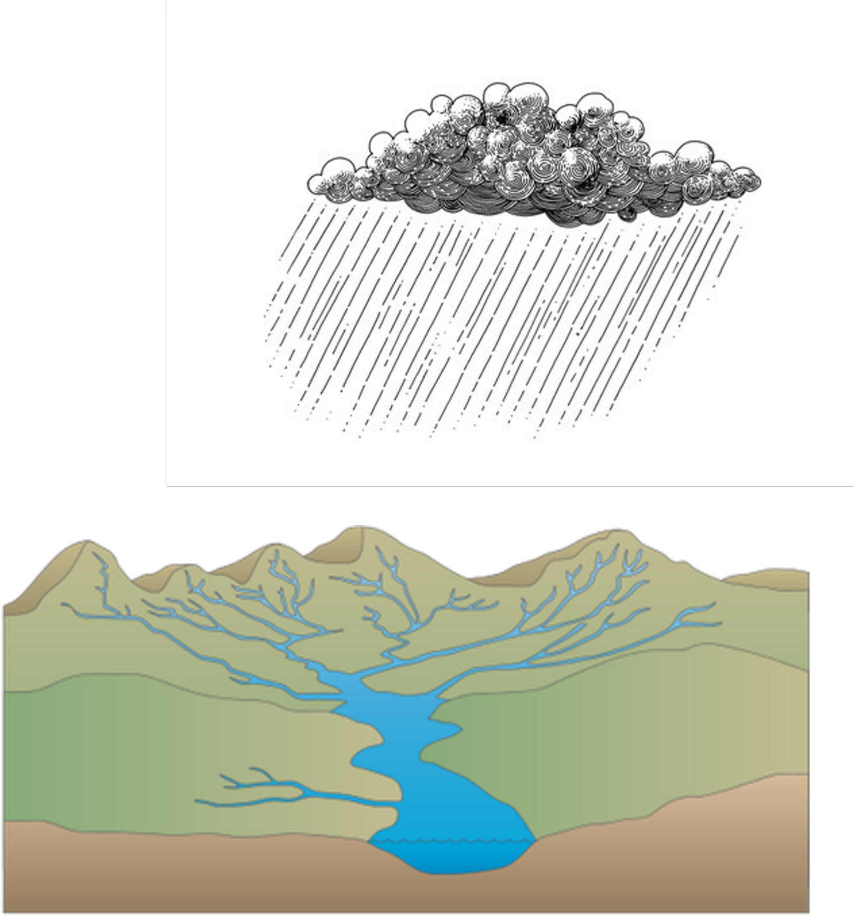


Figure 1.12: Rainfall accumulating in a river basin is one example where the autocorrelation properties of the predictand (precipitation in this case) can have an important effect on the variance of estimates of the sum total (or areal integral) of the predictand; i.e., the volume of water flowing out of the catchment. Simulations with longer length scales lead to higher variance in total rainfall accumulation (see Figure 1.13).

Auto-correlation within predictive samples causes areal integrals of the predictand to have higher variance than those calculated from unstructured samples /

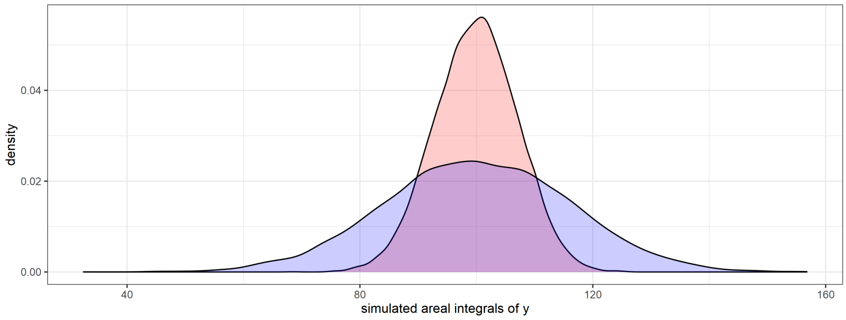


Figure 1.13: Density plots produced by integrating over x for 10000 samples from a multivariate normal distribution with diagonal covariance (red; Figure 1.10) and Matern(5,4) covariance (blue; Figure 1.11). Note how the auto-correlated samples cause the integral of the predictand (e.g., rain accumulated by a drainage basin) to have higher variance.

simulations whose predictand values are independent from one another. We can think for example of models of rainfall over a river catchment, where independent predictions of rainfall at each point within the catchment cause estimates of the total volume of accumulated water to be quite stable. Conversely, highly auto-correlated predictions of rainfall across the catchment cause estimates of the total volume of water accumulated to have much higher variance, because simulations featuring low rainfall everywhere, or high rainfall everywhere, are much more likely. Another example would be in the context of resource estimation for mining, where the variance of the estimated total tonnage of some commodity metal within a volume of rock will again be dependent on the spatial auto-correlation of the individual predictive samples, with more spatially auto-correlated samples resulting in estimates with higher variance.

The spatial auto-correlation of predictive samples is just one example of a 'higher-order property' of predictive distributions that goes beyond the typical assessments of calibration and sharpness, and highlights the fact that model checking is not a one-size-fits-all exercise; the ways that we assess the quality of our models should depend on the purposes for which they are used. For example, weather forecasters have historically prioritised the physical plausibility of their predictive samples by developing models that simulate the atmosphere using the rules of physics. As we shall see, this tends to come at the expense of the calibration of the predictive distributions they can provide, and hence statistical post-processing is necessary to correct this (see Chapter 3; A framework for probabilistic weather forecast post-processing across models and lead times).

The ‘ideal’ environmental models would generate predictive distributions that are perfectly well-calibrated, as sharp as possible, *and* consist of physically plausible samples with covariance / auto-correlation properties that match our observations of the real-world. However, the likely expense of such models, both in terms of development time and computational requirements, means that such ‘perfect’ aims are not always sensible in practice. This thesis covers a range of probabilistic machine learning approaches that satisfy the differing requirements of the various applications we look at. A general theme is the need to model large environmental datasets in useful ways using methodologies which are relatively straightforward to implement in practice. As George Box (1919-2013; Wasserstein, 2010) is famously quoted as saying: “all models are wrong, but some are useful”.

1.2.5 The primacy of doubt

> “He believed in the primacy of doubt: not as a blemish on our ability to know, but as the essence of knowing.”
>
> - James Gleick (1993) on theoretical physicist Richard Feynman

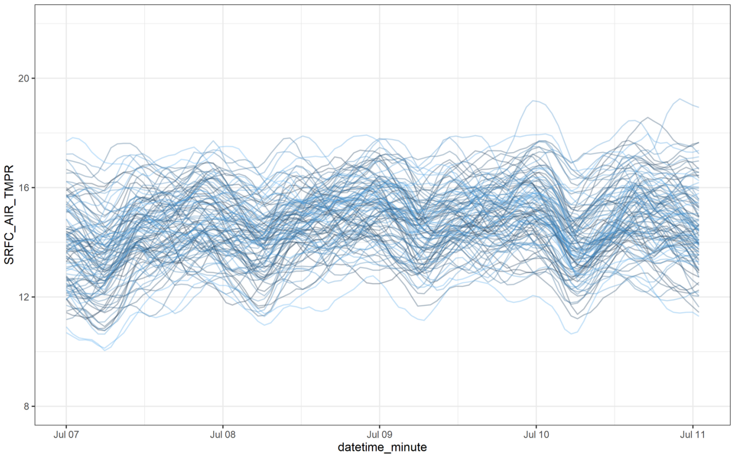
In the context of weather forecasting, Tim Palmer (2017) makes the case that “the ability to quantify uncertainty is not a “bolt-on” extra, but rather a sine qua non [an essential requirement]. Quantifying uncertainty is less about predicting the skill of some “central” prediction, as estimating probability distributions of future weather”. Meanwhile, Andrew Gelman et al. (2013) explains that “the guiding principle [of using probability as a measure of uncertainty] is that the state of knowledge about anything unknown is described by a probability distribution.”

The power of probabilistic modelling therefore comes from its ability to formally represent our *state of knowledge*, which would otherwise be buried — perhaps only loosely assembled — within our own minds. By embracing the primacy of doubt, we can construct probabilistic models that are honest reflections of our uncertainty, and use this uncertainty as a guide towards improving our state of knowledge.

1.2.5.1 Probabilistic models as knowledge - a visual example

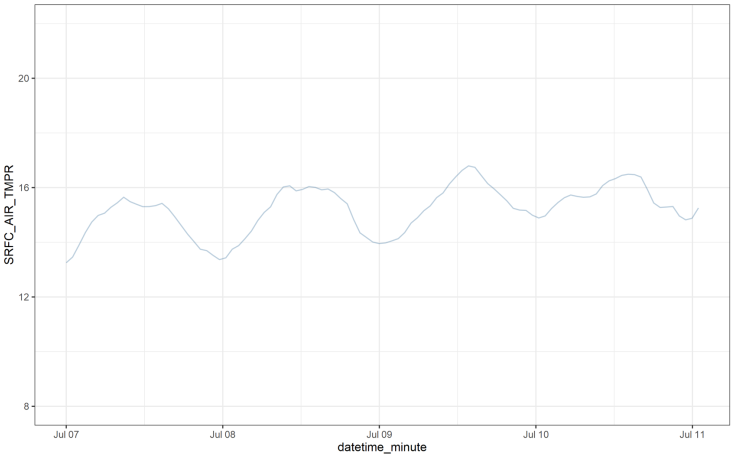
To demonstrate the concept of probabilistic modelling as a representation of our knowledge, we provide here an illustrated toy example of adopting the Bayesian philosophy to model the surface air temperature observed at a random site in the UK during July 2021. We start with our prior distribution $p(H)$, which is a distribution over hypotheses that we think could plausibly generate these type of observations.

It seems reasonable to believe that the temperature observations would be generated by some smooth function that varies in some way according to the time of day, averaging about 15℃. As such, our prior distribution $p(H)$ is a distribution over these kind of functions, representing our degree of belief in plausible hypotheses for the generation of temperature observations (although it is worth noting that we could also have beliefs simply about what the temperature will be, without proposing a data generating model):

$p(H) =$ 

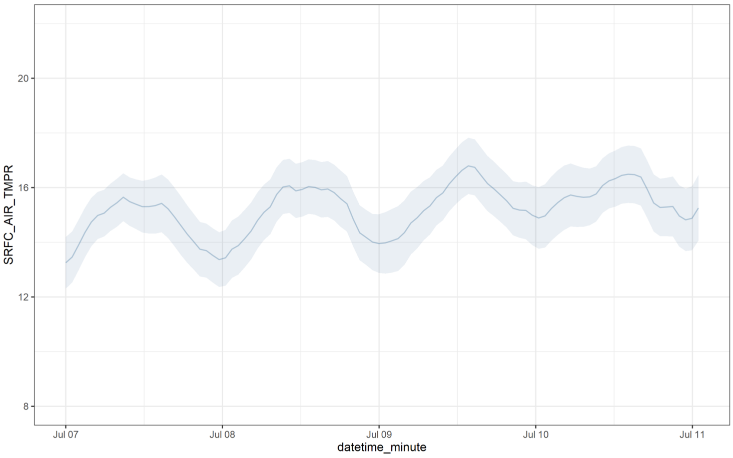
. (1.21)

In this case, each hypothesis is just a smooth function that varies through time and by time of day:

$H_x =$ 

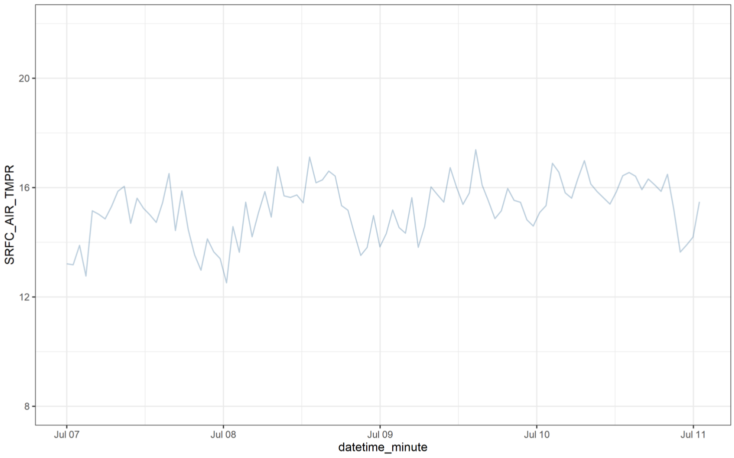
. (1.22)

However, we also assume that each hypothesis can generate data from a Gaussian distribution centred around it (e.g., in order to capture measurement error):

$p(Y|H_x) =$ 

. (1.23)

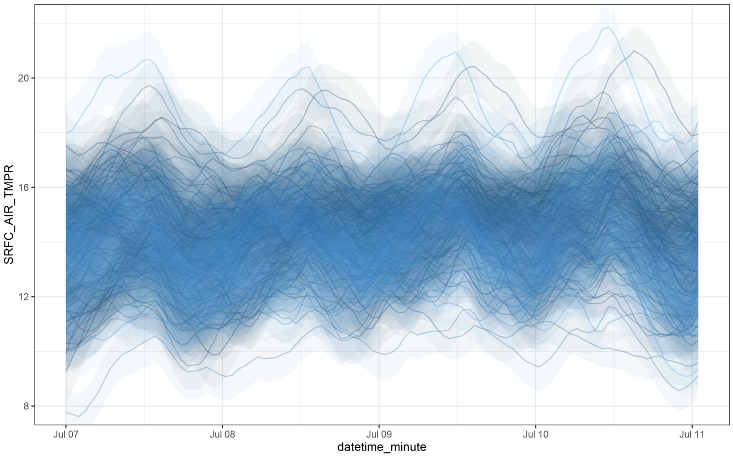
This Gaussian distribution provides the likelihood, $p(Y|H_x)$, from which we can simulate hypothetical sets of observations that we may expect to observe if H_x was true:



$\sim p(Y|H_x)$. (1.24)

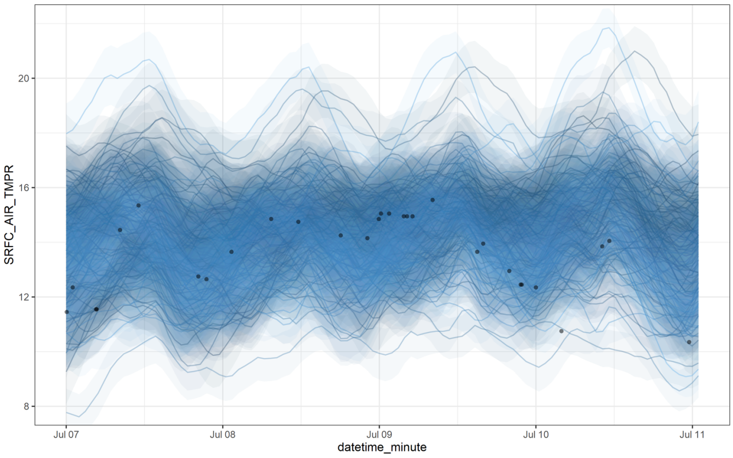
Integrating over our prior data-generating hypotheses gives us our prior predictive distribution (density of temperature values we think we might observe *a priori*). This represents both data uncertainty (i.e., by the variance of each hypothetical function's Gaussian likelihood) and model uncertainty (i.e., the spread between different hypotheses):

$$\int_H p(Y|H)p(H)dH = \quad . \tag{1.25}$$



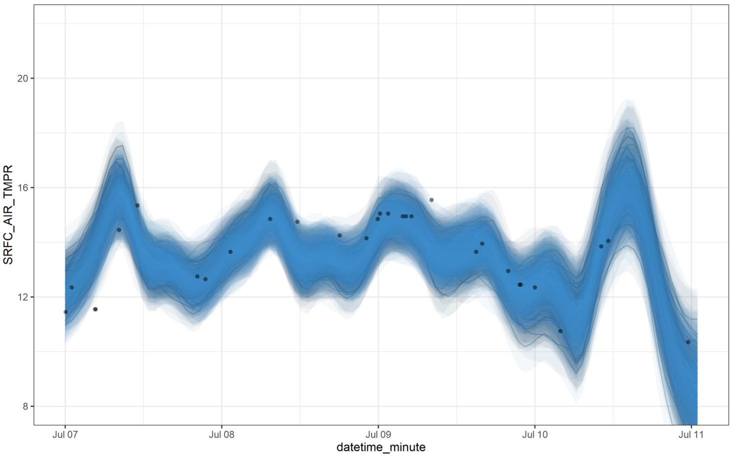
Now, if we collect some observations, it becomes apparent that our prior predictive distribution is overly cautious; it represents an outdated state of knowledge which should be updated given the new observations:

$$\int_H p(Y|H)p(H)dH = \quad . \tag{1.26}$$



Updating our state of knowledge requires computing the posterior distribution of hypotheses, i.e. the distribution of hypothesis given the data, $p(H|\boldsymbol{D})$. This illustration uses the Monte Carlo dropout method for approximate Bayesian inference in deep neural networks (and so results will differ from exact inference, but the principle remains):

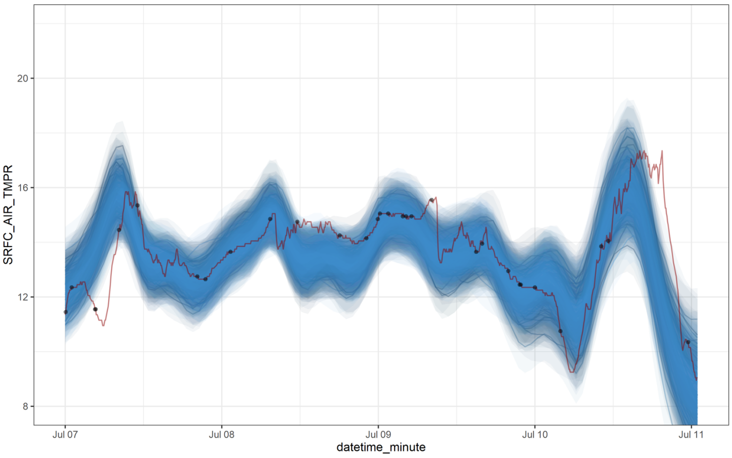
$$\int_H p(Y|H)p(H|\boldsymbol{D})dH =$$



$$. \tag{1.27}$$

As we have seen, a good probabilistic model should generate a well-calibrated (and sharp) predictive distribution, such that new observations will fall within a 95% prediction interval 95% of the time (and likewise for all other intervals). The red line below shows the true observations, from which our sample observations were taken. It is reasonably well represented by our posterior predictive distribution:

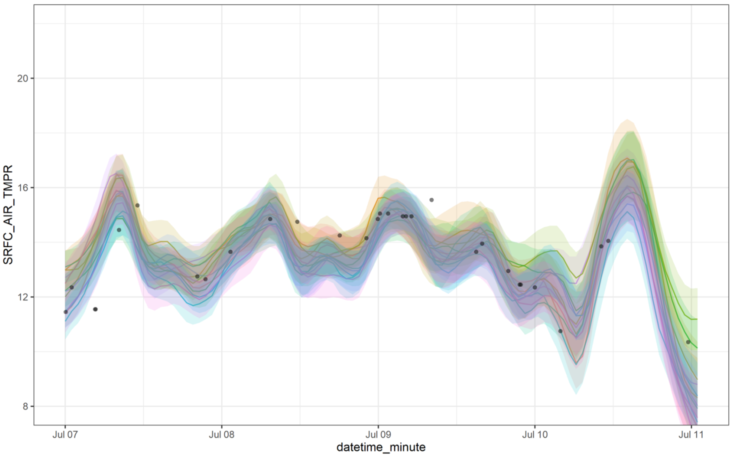
$$\int_H p(Y|H)p(H|\boldsymbol{D})dH =$$



$$. \tag{1.28}$$

If we were to reduce the number of Monte Carlo samples used to estimate our posterior predictive distribution, the quality of calibration is likely to suffer, especially in the tails of the predictive distribution (where infrequent but potentially important events may occur). Here we display just a handful of hypotheses sampled from the posterior, each uniquely coloured:

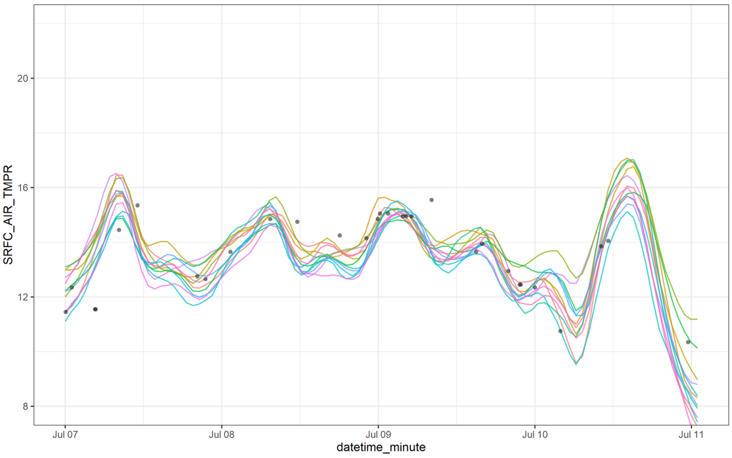
$$\int_H p(Y|H)p(H|\boldsymbol{D})dH =$$



. (1.29)

This, as we shall see, takes us into similar territory to the outputs of numerical weather prediction ensembles, which tend to be limited in the number of ensemble members (effectively posterior predictive samples) that they produce. In addition, it is the experience of this thesis that weather forecasting ensembles do not include 'data uncertainty' i.e., $p(y|H)$, but instead provide only samples of $p(H|D)$:

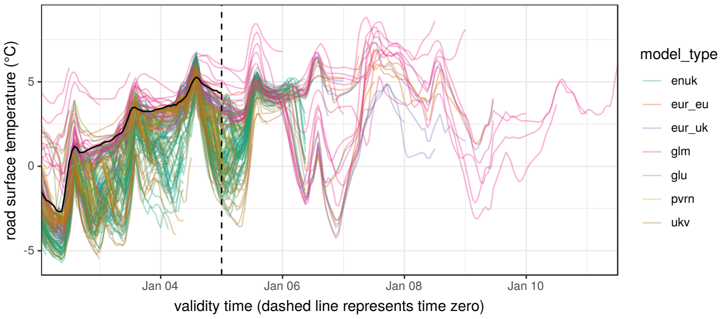
$$\int_H \cancel{p(Y|H)}p(H|\boldsymbol{D})dH =$$



. (1.30)

Thus, despite our Bayesian inference example here being for a temporal interpolation problem rather than a forecast of the future, the similarity between a sparsely-sampled Bayesian posterior distribution and the output from numerical weather prediction should be quite apparent (see more details in Chapter 4; A framework for probabilistic weather forecast post-processing across models and lead times):

$$\int_H \cancel{p(Y|H)}\, p(H|\boldsymbol{D})\, dH =$$



.

(1.31)

And it is of note that the lack of sampling for the data uncertainty, $p(y|H)$, for each hypothesis / ensemble member is in itself a reason why Numerical Weather Prediction (NWP) ensembles tend to be underdispersive; i.e., they tend to underestimate uncertainty (Raftery et al., 2005).

Similar limitations apply, although to an even greater extent, to the practice of geological mapping, in which maps are traditionally drawn by hand with no ability to convey uncertainty — only a single hypothesis is portrayed, and the uncertainty of data around this hypothesis is not considered (Figure 1.15). This is the antithesis of embracing the primacy of doubt, and Chapter 7; Geological mapping in the age of artificial intelligence, proposes a Bayesian solution to it.

It is apparent then, that despite continual development in the practices of environmental modelling — including for both weather and geology — there is a logical need to embrace the primacy of doubt, and to be confident in issuing probabilistic predictions which may themselves be far from confident. The real-world risk-abating benefits of probabilistic modelling are clear in comparison to the pit-falls that await those who would disregard the primacy of doubt in a show of false confidence.

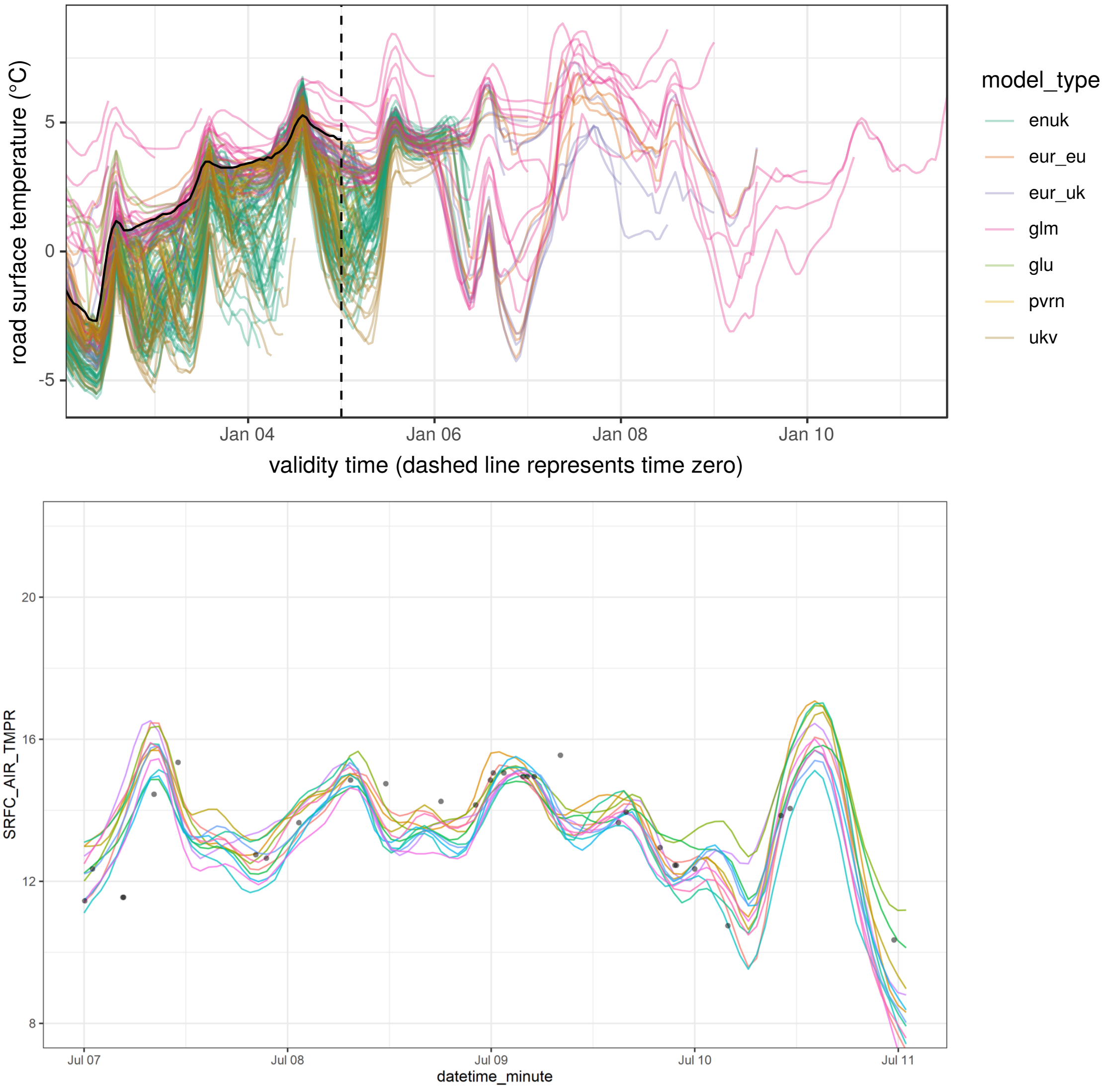


Figure 1.14: The similarity between weather forecast ensemble members (**top**), and samples from a Bayesian posterior distribution (**bottom**).

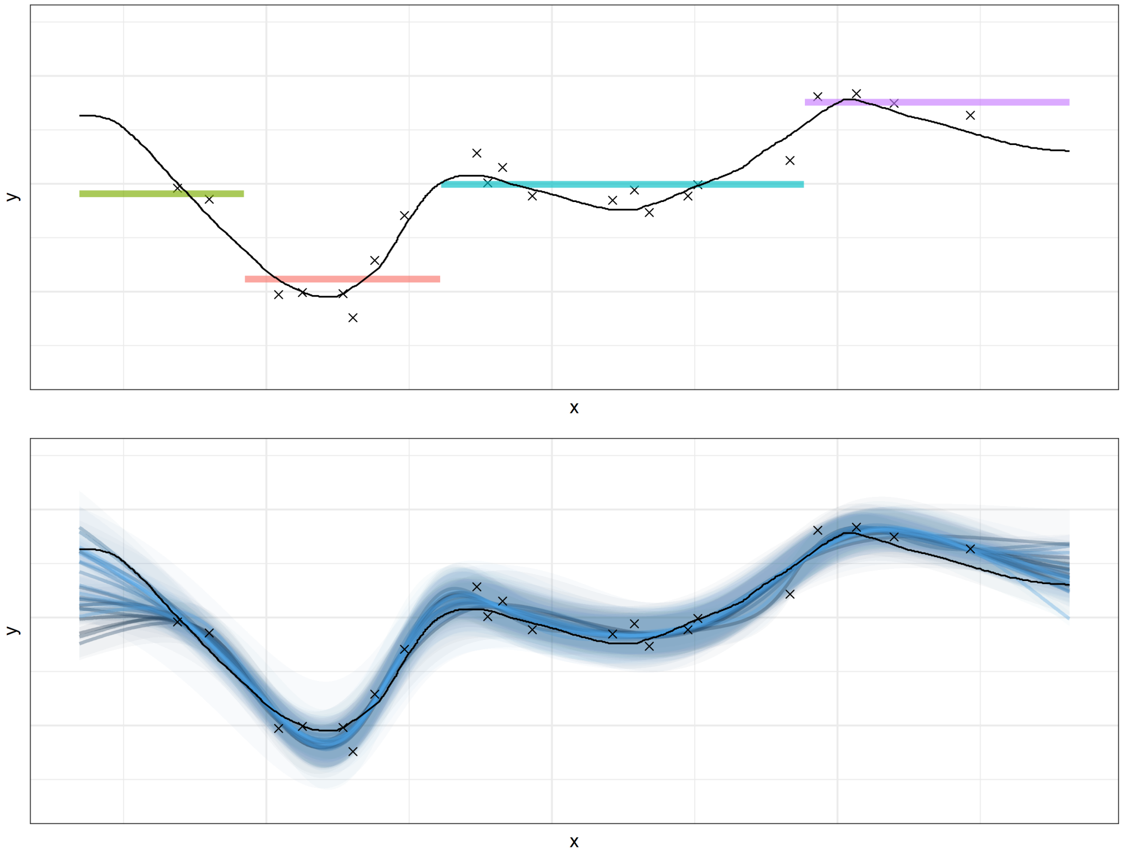


Figure 1.15: Conceptual comparison of traditional geological mapping (**top**) and geological mapping using a Bayesian neural network (**bottom**). The black line shows the true value of some geological property, y, through space, x. Black crosses are the observations of this property, with some error, which the two mapping approaches utilise. While mapping using a single set of discrete classes fails to represent continuous geological properties, and cannot convey uncertainty, mapping ‘properties first’ using Bayesian AI methods brings the potential to obtain skilful probabilistic maps of any geological properties of interest. See more in Chapter 7.

Chapter 2

Background

As each of the subsequent research chapters (3 - 7) includes its own review of relevant background literature, this standalone background chapter provides only a brief summary of the history and progress of the disciplines of weather forecasting and geological mapping through the lens of statistics / probabilistic machine learning. There is generally very little overlap between these two disciplines in the literature, which makes sense on account of the stark differences in properties and behaviour between the atmosphere and the lithosphere, and yet, through a data science lens, the aims of both — to make useful predictions of yet-to-be-observed conditions through space (and time) — are rather similar.

As such, it is not surprising that some statistical methodologies, for example those of geostatistics, do have a history of being applied within both disciplines. It seems likely that in the future, as machine learning methods become increasingly integrated within the sciences, that the fundamental commonalities between weather forecasting and geological mapping will provide new opportunities for the development of models which are inspired by work within both disciplines. The later chapters of this thesis (4-7) detail the development of a probabilistic deep learning technique — Bayesian deep learning for spatial (and spatio-temporal) interpolation in the presence of auxiliary information — that has properties that make it reasonable for learning models of both atmospheric and lithospheric variables. It could be that this approach therefore contributes a step in a direction of progress towards the development of artificial intelligence systems for probabilistic modelling of the environment as a whole.

2.1 The rise of machine learning

The last few years has seen something of an explosion in the adoption of machine learning methods in both industry and academia, often producing state-of-the-art results in a wide range of applications. The kernel for this explosion can be traced back to the 2012 ImageNet image classification competition, which was won by AlexNet: a deep convolutional neural network developed by Geoffrey Hinton's team (Krizhevsky, Sutskever and Hinton, 2012). This was the first convincing demonstration that bottom-up feature learning could outperform manual top-down feature engineering for solving complex modelling problems. This result more-or-less made 20 years worth of computer vision research in custom feature extraction redundant (e.g. see review by Tian, 2013), and in time sent ripples across the wider scientific community. It is now increasingly being shown that, given sufficient data, self-learning function approximators (whether statistical or machine learning methods by name) are capable of outperforming manually constructed models for solving complex tasks. At the time of writing this sentence (Sept 2022), Krizhevsky, Sutskever and Hinton's AlexNet paper has been cited 115099 times (Figure 2.1). Incredibly, this is up from 35680 citations in Feb 2019, when I first noted the figure down in the early days of my PhD.

The following sections provide a brief review of applications of data science techniques to weather forecasting (and then to geological mapping), and explores some opportunities where this technology may yet be applied. Methods from both

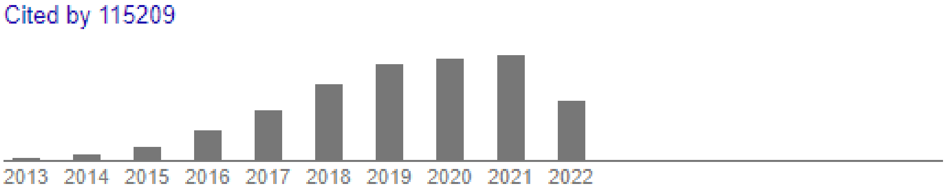


Figure 2.1: Google scholar citations of Krizhevsky, Sustkever, and Hinton's breakthrough work achieving a new state of the art in image classification using deep neural networks in 2012 Krizhevsky, Sutskever and Hinton, 2012. From 342 citations in 2013, to over 20 000 citations each year from 2019, this timeline depicts the explosion in interest in artificial intelligence methods over the last 10 years.

the statistical modelling and the machine learning literature are considered, and in fact these fields are seen as more-or-less synonymous for the purposes of this literature review — both fields provide methods for learning (to approximate functions) to represent the relationships between variables in order to predict the value or class assignment of new observations.

For weather forecasting, the output variables are those that describe the weather at a given location and time e.g, air temperature, precipitation, wind speed and direction. Having good predictions of such variables is useful for decision making in our everyday lives (e.g., should I take an umbrella to work?) but also underpins a huge range of applications; energy(Zhang, Wang and Wang, 2014), transport(Berrocal et al., 2010), defence(Miller, 1967), agriculture(Calanca et al., 2011), hazard mitigation (Dale et al., 2014) and more (Katz and Murphy, 2005). Extreme weather events such as hurricanes and flooding are among the most damaging natural hazards that we face. Effectively mitigating the risk of such hazards requires having well-calibrated and sharp probabilistic forecasts that can allow us to make decision which are closer to optimal, and it is here that statistical and machine learning methods, with foundations in probability theory, have much to offer.

For geological mapping, the output variables are those that describe the geology (the ground beneath our feet) at a given location. For example, age, texture, and composition are variables on which traditional rock classification schemes are constructed, but there are additional specific variables which may be of interest, for example the concentrations of commodity elements or potentially harmful elements, and the values of engineering properties such as shear strengths. Just as for weather forecasting, generating well-calibrated and sharp probabilistic predictions of these properties allows us to improve the outcomes of our interactions with the lithosphere.

2.2 Weather forecasting

Weather forecasting has existed as a quantitative science for over a century. Following from initial synoptic-map based manual forecasting efforts by Robert Fitzroy in the mid 1800s (Hughes, 1988), it was around the year 1890 that Cleveland Abbe, Chief Meteorologist of the United States Weather Bureau first proposed that meteorology should be “essentially the application of hydrodynamics and thermodynamics to the atmosphere” (Lynch, 2008). In his subsequent publication *The physical basis of long-range weather forecasting* Abbe, 1901 described how physical equations could be used to represent atmospheric processes, and conveyed his hope that atmospheric scientists would “take up our problems in earnest and devise either graphical, analytical, or numerical methods” with which to solve these equations.

Soon after, Norwegian physicist Vilhelm Bjerknes, who had been in correspondence with Abbe (Friedman, 1993), published his work on *The problem of weather forecast, viewed from the standpoint of physics and mechanics* (Bjerknes, 1904). Bjerknes defined seven equations to represent the seven variables of pressure, temperature, density, humidity and three components of velocity, and devised a qualitative graphical method to solving them. These graphical methods allowed Bjerknes to progress weather charts forwards through time in steps based on the rules of physics, however Bjerknes was unsatisfied with the accuracy of his method — a problem confounded by the lack of available observations — and, given the technology of the time, could see no practically useful way to implement his work (Lynch, 2008).

Later, in 1913, British scientist and mathematician Lewis Fry Richardson joined the UK Meteorological Office as Superintendent of the Eskdalemuir Observatory in Scotland (Lynch, 2008). Richardson had little prior experience of meteorology, but had previously worked across scientific disciplines, including publishing work on using finite differences to solve physical problems involving differential equations

(Richardson and Glazebrook, 1911). Richardson recognised the suitability of his finite difference method to provide approximate solutions to the equations proposed by the likes of Abbe and Bjerknes, and went ahead to develop the first numerical methods for weather prediction, which he communicated in his book *Weather Prediction by Numerical Process* (Richardson, 1922).

Despite these key works of the late 19th and early 20th century, the first implementation of numerical modelling for weather prediction on an electronic computer did not take place until 1950, when Charney, Fjörtoft and von Neumann of the Meteorology Project at Princeton, USA succeeded in implementing hindcasts on the Electronic Numerical Integrator and Computer (ENIAC) (see Charney, Fjörtoft and Neumann, 1950). Their successes made numerical weather forecasting an operational reality within five years, with the first real-time forecasts being made in Stockholm in 1954 (see Bolin, 1955).

It was still not until the 1970s that developments in supercomputing made solving the full set of equations proposed by Abbe and Bjerknes feasible (Lynch, 2008), but the near-exponential increases in computing power over the years (Moore first hit upon his law in 1965), combined with continued improvements in data collection, model parametrisation, and numerical solution methods, have resulted in forecasting skill increasing at a rate of about one additional day per decade (i.e. 6-day forecasts today are as accurate as 5-day forecasts were ten years ago). However, these methods, which have come to be referred to ubiquitously as numerical weather prediction (NWP), are essentially the continuation of the ideas of Abbe, Bjerknes and Richardson.

Prior to the practical implementation of Numerical Weather Prediction in the 1970s, statistical methods rather than dynamical modelling were being used to provide weather forecasts as part of a push towards ‘objective weather forecasting’ (Byers et al., 1951) and an effort to move away from reliance on meteorologists’ subjective expert judgement as a way to issue forecasts (e.g. Brier, 1946; Gringorten, 1955). As Glahn (1982) writes: “Statistical weather forecasting, in its broadest sense, has undoubtedly been practiced for thousands of years. All that is necessary is for someone to collect some data, someone to process it, and someone to use the results to make a forecast. Ancient man, seeing a dark cloud approaching and thinking that rain was likely, would be practising statistical weather forecasting even if he had no knowledge of the physical processes involved".

Further, Glahn (1982) wrote that "In the early years of operational numerical weather prediction, competition rather than cooperation dominated the relationship between those individuals engaged in developing statistical models and those researchers concerned with developing numerical models. Each group thought that its approach was the best way to proceed and that the other branch of objective weather prediction was not necessary. Even though the barriers between the two groups have not yet vanished, each group has become much more tolerant of the other group's viewpoint. Statistical modelers now use the results from (rather than compete with) numerical models, and numerical modelers recognise the usefulness of properly applied statistical procedures."

2.2.1 Applications and opportunities

In their review *the quiet revolution of numerical weather prediction* Bauer, Thorpe and Brunet (2015a) explain that, so far, the greatest increases in predictive skill have been obtained from improvements in three areas — physical process representation, ensemble forecasting, and model initialisation. Bauer, Thorpe and Brunet also anticipate that these three areas will continue to provide the greatest improvements to NWP in the next decade, but that additional challenges come from the high-performance computing required to run NWP at ever higher resolutions.

So where does machine learning fit in? The ultimate application of machine learning to weather forecasting would be in the implementation of an end-to-end architecture: a system that could predict the future values of weather variables based on their current values by learning to model the relationships between past and future states using historic observations. Krizhevsky, Sutskever and Hinton's deep neural network for classifying images was one such system — it was able to learn a function to map images to their labels through example alone. When I started this PhD, no such example had been demonstrated in the space of weather forecasting, however given the commercial attention that machine learning, and in particular deep learning, has been getting, it was perhaps not surprising that large corporates, such as NVIDIA, would step up to the mark and develop end-to-end deep learning systems for weather forecasting (Kurth et al., 2022; Pathak et al., 2022). Deep learning based forecasting systems are now being developed outside of these corporations too with the help of GPU donations (to hardware accelerate neural network training) from NVIDIA (e.g., Weyn et al., 2021). In the latter half of this PhD, the UK's European Centre for Medium Range Weather Forecasting (Düben et al., 2021), Natural Environment Research Council (NERC, 2022), and Met Office (Met Office, 2022) have all published strategies specifically for developing and incorporating machine learning and data science methods into their operations. In the case of the Met Office, this includes the launch in 2020 of the University of Exeter - Met Office partnered Joint Centre for Excellence in Environmental Intelligence. These organisational strategies do not necessarily revolve around deep learning only, but are an indication of the technological (and philosophical) changes that we are currently living through and which provide the backdrop for this PhD.

At the start of this PhD, end-to-end deep learning systems had already been developed to solve tasks within a range of fields outside of weather forecasting, for example Amodei et al. (2016) presented an end-to-end deep learning solution for the transcription of audio to text, stating that "because it replaces entire pipelines of hand-engineered components with neural networks, end-to-end learning allows us to handle a diverse variety of [conditions]". The same technology powers the translation and voice recognition functionalities of modern smartphones, and is a likely candidate for the realisation of self driving cars (Bojarski et al., 2016). However, while replacing pipelines of hand engineered components certainly sounds applicable to current NWP workflows, unlike the above tasks, accurate weather forecasting could never be considered learnable by one individual human, and the challenge of matching or outperforming 100 years of NWP development is likely to be somewhat tougher, with more stringent requirements.

When Lewis Fry Richardson first joined the UK Meteorological Office in 1913, the meteorologists used an *Index of Weather Maps* to look up maps of past weather scenarios that were most similar to the one presently being observed. Predictions were then made by eye, on the assumption that “what the atmosphere did then, it will do again now”. Richardson was sceptical of this technique, writing “why then should we expect a present weather map to be exactly represented in a catalogue of past weather?” (Lynch, 2008). Ironically, the concept of the *Index of Weather Maps* approach is essentially the same as that of machine learning, although machine learning systems are specifically evaluated on their ability to make accurate predictions in situations that they have not previously encountered, so as to demonstrate their competence in interpolating within the space of a task, rather than simply regurgitating the nearest known examples as a best guess (i.e., K nearest neighbours algorithm with K = 1). Perhaps today Richardson could be convinced by a suitable demonstration of modern machine learning methods.

Taking a step back from end-to-end approaches (most of which, it should be said, have not so far been concerned with the modelling of uncertainty - self driving cars being the exception), less grandiose applications of self-learning function approximators (e.g., statistics, machine learning, AI) to weather forecasting require the integration of these methods with existing NWP frameworks. Recapping Bauer, Thorpe and Brunet’s three key areas for improvements going forward: physical process representation, ensemble forecasting, and model initialisation (and the fourth, speed), it seems that none are beyond the reaches of potential benefits from machine learning approaches.

In terms of physical process representation, Gentine et al. (2018) and Rasp, Pritchard and Gentine (2018) find that neural networks are able to adequately learn to represent subgrid processes in atmospheric models, reducing computational complexity in the process. O'Gorman and Dwyer (2018) apply random forests in a similar fashion to learn to parameterise moist convection, and conclude that the use of machine learning is promising, although they note issues with failing to generalise to different climates (partly because decision trees cannot extrapolate).

The use of statistical methods in conjunction with ensemble forecasting has a longer history. Ensemble forecasting is NWP's answer to uncertainty quantification, given that each individual NWP model run is deterministic and provides only a point forecast. Ensembles can be run using a range of model types — including with parameters perturbed to capture model uncertainty — and a range of initial conditions, where initial conditions are perturbed to capture uncertainty in the observed state (for an overview of the Met Office's ensemble forecasting see `https://www.metoffice.gov.uk/research/approach/modelling-systems/unified-model/weather-forecasting`). Running ensembles of forecasts with these perturbations produces a distribution of point forecasts which provides an indication of forecast uncertainty, although the high computational expense of the numerical model runs limits the number of ensemble members that can be obtained. By using statistical post-processing on the limited number of NWP ensemble members' output we can correct biases and improve calibration (and sharpness) by learning from the historic performance of the numerical models compared to real observations. Statistical post-processing therefore has the potential to increase forecast skill; improving predictions of the most likely outcomes and quantifying the probabilities of extreme events.

The principle established methods for the statistical post-processing of NWP ensembles are Ensemble Model Output Statistics (EMOS; Gneiting et al., 2005) and Bayesian Model Averaging (BMA; Raftery et al., 2005). These two approaches, both co-authored by Tilmann Gneiting and Adrian Raftery (and others) in the same year, emerged as solutions to the problem of calibrating ensemble forecasts, given that ensemble forecasting first became an operational reality around this time (including that year at the Met office; Bowler et al., 2008).

The EMOS approach (Gneiting et al., 2005) proposes using multiple linear regression to predict a univariate weather quantity y, such that

$$y = a + b_1 X_1 + ... + b_m X_m + \varepsilon \tag{2.1}$$

where a and $b_1, ..., b_m$ are regression coefficients, and ε is an error term. In order to capture the spread-skill relationship, by which an increase in spread between ensemble members will tend to correspond to increasing uncertainty, (Gneiting et al., 2005) model the error term, ε, as a linear function of the ensemble spread. That is,

$$Var(\varepsilon) = c + dS^2, \tag{2.2}$$

where S^2 is the ensemble variance, and where c and d are nonnegative coefficients. Combined, these equations yield the Gaussian predictive distribution

$$N(a + b_1 X_1 + ... + b_m X_m, c + dS^2) \tag{2.3}$$

whose mean derives from the regression eqution and forms a bias-corrected weighted average of the ensemble member forecasts, and whose variance depends linearly on the ensemble variance (Figure 2.2).

Through this task-tailored multiple linear regression approach, EMOS combines information from numerical ensemble members into a single bias-corrected unimodal predictive distribution. While this is simple, it is also a potential weakness, as it fails to account for any inherent multimodality that the numerical ensemble may legitimately exhibit. BMA on the other hand is able to capture such multimodality, as explained below.

Where EMOS combines the numerical ensemble into a single statistical model,

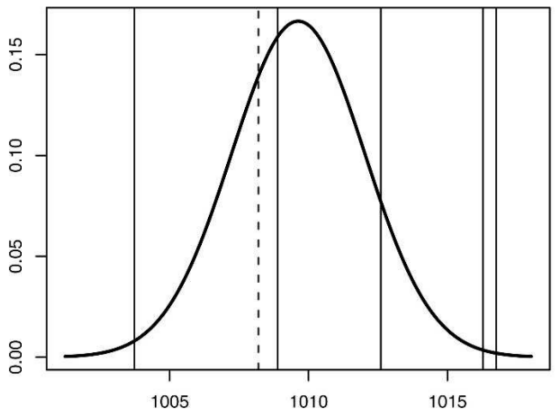


Figure 2.2: Example predictive probability density function produced by the EMOS post-processing approach for a 48-h forecast of sea level pressure. Also shown are the five ensemble member forecasts (solid vertical lines) and the verifying observation (broken vertical line). From (Gneiting et al., 2005).

BMA treats each ensemble member as a statistical model, and then (as the name suggests) applies Bayesian Model Averaging to combine them. By doing so, BMA captures the between-model uncertainty better than EMOS can, and thus is less likely to produce under-dispersive output.

As (Raftery et al., 2005) explain, in the case of a quantity y to be forecast on the basis of training data Y using K statistical models $M_1,...,M_K$, the law of total probability tells us that the forecast PDF, $p(y)$, is given by

$$p(y) = \sum_{k=1}^{K} p(y|M_k)p(M_k|Y), \tag{2.4}$$

where $p(y|M_k)$ is the forecast PDF based on model M_k alone, and $p(M_k|Y)$ is the posterior probability of model M_k being correct given the training data, and reflects how well the model M_k fits the training data. The BMA probability density function, $p(y)$, is a weighted average of the conditional PDFs given each of the individual models, weighted by their posterior model probabilities $p(M_k|Y)$.

To extend this from the averaging of statistical models, to the averaging of numerical ensemble members requires that each ensemble member f_k provides its own PDF $g_k(y|f_k)$, which is not naturally the case since numerical weather prediction ensemble members provide a deterministic point-prediction, rather than a probability distribution. If each ensemble member provides its own PDF $g_k(y|f_k)$, then the BMA predictive distribution becomes

$$p(y|f_1,...,f_k) = \sum_{k=1}^{K} w_k g_k(y|f_k), \tag{2.5}$$

where w_k is the posterior probability of the forecast k being the best one and is based on forecast k's performance in the training period.

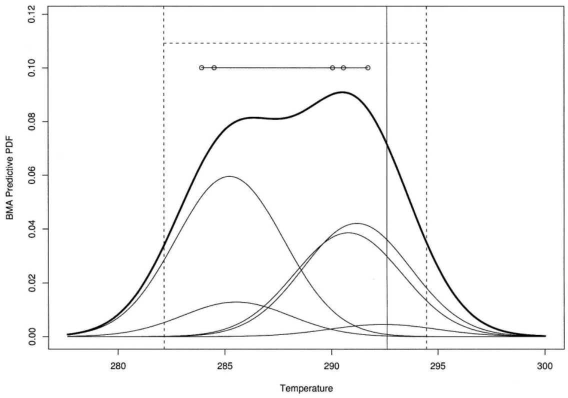


Figure 2.3: Example predictive probability density function (thick curve) and its five components (thin curves) produced by the BMA post-processing approach for a 48-h forecast of temperature. Also shown are the ensemble member forecasts and range (solid horizontal line and bullets), the BMA 90% prediction interval (dotted lines), and the verifying observation (solid vertical line). From (Gneiting et al., 2005).

For some weather variables (e.g. temperature and pressure) it is reasonable to assume a Gaussian distribution for the errors around a forecast ensemble member, such that each ensemble member's conditional PDF $g_k(y|f_k)$ should be a normal distribution with mean $a_k + b_k f_k$ and variance σ^2:

$$y|f_k \sim N(a_k + b_k f_k, \sigma^2). \tag{2.6}$$

The model parameters a_k, b_k, w_k for $k = 1,...,K$, and σ^2 require estimation, which is achieved on the basis of a training dataset consisting of ensemble forecasts and verifying observations. Following (Raftery et al., 2005) we denote space and time by subscripts s and t so that f_{kst} denotes the kth forecast for place s and time t, and $y_{s,t}$ denotes the corresponding verification.

First, a_k and b_k are estimated by simple linear regression of y_{st} on f_{kst} using the training data. This can be viewed as a simple bias correction of each forecast / ensemble member. Then, w_k for $k = 1, ..., K$ and σ^2 are estimated by maximum likelihood using the same training data. The log-likelihood function for the model (Equation 2.5) is

$$\ell(w_1, ..., w_k, \sigma^2) = \sum_{s,t} \log \left(\sum_{k=1}^{K} w_k g_k(y_{st}|f_{kst}) \right). \tag{2.7}$$

In summary, BMA for ensemble weather forecast post-processing first performs a simple bias correction of each ensemble member, and then ‘dresses’ each ensemble member with its own conditional PDF, with shared variance σ^2 which, along with the mixing weight, w_k, of each ensemble member, is optimised to maximise the likelihood of the resultant blended PDF in relation to training observations. This process is performed separately for each lead time (Raftery et al., 2005).

Because the variance of each ‘dressed’ forecast in (Raftery et al., 2005, ’s) BMA approach is controlled by a single shared parameter σ^2 which is optimised to maximise the likelihood, calibration is to some extent built in as a natural trade-off between intra-model (within-model) and inter-model (between-model) variance. However, as we move into the age of artificial intelligence it is likely that it will become commonplace to generate statistical forecasts that inherently have their own PDF, $p(y|f)$, rather than just point predictions. In this case, as we see in the next chapter, if the individual forecasts are themselves well-calibrated, then the use of BMA without control of the variance of the individual forecasts tends to produce an over-dispersed predictive distribution, because adding inter-model variance to forecasts with pre-calibrated intra-model variances leads to overdispersion. A different approach may therefore be more appropriate (which we propose in Chapter 3).

While EMOS and BMA are the established core of NWP post-processing methods, other methods have been and continue to be proposed, which go beyond linear regression setups (Hagedorn, Doblas-Reyes and Palmer, 2005) in order to capture more complex effects, which may include relationships to covariates outside of the forecast ensemble itself. For example, neural networks, (Shamseldin, O'Connor and Liang, 1997; Rasp and Lerch, 2018), gene expression programming (Zaherpour et al., 2019), and decision tree ensemble techniques Taillardat et al. (2016) have been proposed. Deep learning based generative models such as variational autoencoders (VAEs - Kingma and Welling, 2013), and generative adversarial networks (GANs - Goodfellow et al., 2014) also have great potential due to their ability to learn potentially complex sampling distributions.

We can perhaps take a step back and say that the ideal weather forecast would be a well-calibrated and sharp-as-possible predictive distribution composed of physically plausible samples or simulations. Existing numerical weather prediction ensembles, which are physically plausible in the sense that they operate on physical equations (although with some unrealistic constraints, such as their discretisation into coarse grids) therefore get close to satisfying requirements of physical plausibility, but this tends to be at the expense of calibration. Numerical weather prediction systems are also highly computationally expensive, which limits the number of samples (ensemble members) that they can generate. Therefore, it is computationally more efficient to use the output of numerical weather prediction to *inform* statistical post-processing systems which can generate many more samples from a well-calibrated and sharp predictive distribution. In the extreme, statistical post-processing systems may even utilise just a single deterministic numerical forecast to good effect (e.g. Chapman et al., 2019). To achieve physically-plausible predictive samples from the statistical post-processing system is perhaps the hardest part of the task, but cheaper approximations may be acceptable depending on the application.

2.2.2 Summary

The concepts behind numerical weather prediction (NWP) have remained largely unchanged for the last century, although increasing data availability due to satellites, combined with innovations in data assimilation (e.g., 4D VAR; Lorenc and Rawlins, 2005) have resulted in continuous steady progress. Meanwhile, there have been revolutionary breakthroughs in machine learning technologies within the last few years (e.g. deep learning) which can now supersede "entire pipelines of hand engineered components" for complex modelling tasks. The first applications of these technologies to weather forecasting have shown promising results, but we are only just entering what could become a loud revolution (as opposed to the quiet revolution of Bauer, Thorpe and Brunet, 2015b) in weather forecasting methodology. It fundamentally makes sense that statistical and machine learning methods, with foundations in probability theory, have much to offer compared to deterministic numerical methods when it comes to accurately modelling systems as chaotic as the atmosphere - but how best to implement them has remained to some extent an open question.

2.3 Geological mapping

Geological mapping was not originally intended to be part of this thesis, but the conceptual commonalities between probabilistic weather forecasting and probabilistic geological mapping are clear, and thus similar statistical techniques are likely to be applicable to both. Given that geological mapping is currently quite far behind weather forecasting in terms of technological advancement — while the Met Office generates the UK's official weather forecasts using Numerical Weather Prediction run on supercomputers (as do other weather forecasting agencies around the world), the British Geological Survey draws the UK's official geological maps by hand (as do other geological surveys around the world) — there is clearly significant impact to be had in the adoption of more sophisticated mapping techniques by the geological community.

One could argue — I would say incorrectly — that geological mapping would not benefit from probabilistic modelling because we are quite certain of what our maps should look like. While it is the case that the uncertainties involved in geological mapping are probably lesser than those involved in weather forecasting (at long ranges in particular). The uncertainties involved are perhaps of a similar scale to short range weather forecasting, or nowcasting, on time scales at which chaos has only a minimal effect. However, weather forecasting benefits from a great advantage in that the atmosphere is quite observable from space, and therefore satellite observations allow our knowledge of the *current* state of the atmosphere to be relatively certain (albeit with some limitations; the information is not perfect). By contrast, geological maps, which convey our knowledge of the lithosphere, have the disadvantage of the ground being relatively impenetrable to sensing equipment (and the human eye). Seismic data can help here, but the information it provides is limited.

As such, while weather forecast uncertainty increases with time into the future, geological map uncertainty increases with depth below ground. However, the current practice of hand drawing geological maps fails to quantify uncertainty even at the surface, and we need to get that right before we can hope to quantify the larger uncertainties at depth. So, there is a need to develop probabilistic approaches to geological mapping, for which ensemble weather forecasting (including its statistical post-processing) provides good inspiration. While the meteorological community has had around a century of numerical modelling development between their original hand-drawn maps (i.e., the Index of Weather Maps; Lynch, 2008) and their adoption of artificial intelligence, the geological community (at least in the main; and represented by the British Geological Survey) stands a good chance of going straight from hand-drawn maps to probabilistic maps produced by artificial intelligence. There will be challenges along the way, but communication between meteorologists and geologists is likely to facilitate their resolution.

2.3.1 Applications and opportunities

Despite the great differences in how they are generated — with weather forecasts being the solution to the equations of a dynamical system, and geological maps being a data-based classification — we can think of a single hand-drawn geological map as being conceptually equivalent to a single deterministic NWP forecast, in that neither provides an estimate of uncertainty (the geological map is also a coarser representation owing to its reliance on classification, rather than continuous modelling of lithospheric properties akin to the continuous modelling of atmospheric properties that NWP achieves).

Similarly to the BMA approach for weather forecast post-processing, we could simply 'dress' a geological map with a PDF centred on the map (or bias-corrected) and with suitable variance. However, the variance in this case would only be intra-model variance to represent the data uncertainty, rather than the uncertainty of the model / map itself. To make geological mapping truly probabilistic (and embrace the primacy of doubt) requires devising a system to simulate possible geological maps, just as numerical weather prediction can simulate possible weather maps. This would allow the inter-model variance, or model uncertainty, to be represented too.

It has to be noted that the field of geostatistics emerged from geology (or more specifically mining engineering; Krige, 1951; Matheron, 1969) for the purposes of interpolating between and simulating from observations of geological variables. One may superficially think, therefore, that the challenge of probabilistic geological mapping is essentially a solved problem. However, this is not the case, primarily due to limitations of traditional geostatistical methods, i.e. kriging, which we will briefly outline here. Nevertheless, it is true that the development of any future probabilistic geological mapping systems should rightfully be referred to under the umbrella of 'geostatistics', although that term has become perhaps overly associated with traditional kriging.

Kriging corresponds to Gaussian process regression; which (here adopting the explanation of Gelman et al., 2013) is regression using a Gaussian process from which random functions $\mu(x)$ can be drawn at any n prespecified points $x_1,...,x_n$ from the n-dimensional normal distribution,

$$\mu(x_1),...,\mu(x_2) \sim N\left((m(x_1),...,m(x_n)),K(x_1,...,x_n)\right), \tag{2.8}$$

with mean m and covariance K. This is a Gaussian process $\mu \sim GP(m,k)$. The mean function represents an initial guess at the function, so that the Gaussian process can model deviation from this initial guess, whilst being centred on it. Traditional geostatistics, e.g. for resource estimation, would typically use a linear or quadratic mean function for this trend removal purpose (Journel and Rossi, 1989).

The function k specifies the covariance between the process at any two points, with K an $n \times n$ covariance matrix with element (p,q) corresponding to $k(x_p,x_q)$ which can be written in shorthand as $k(x,x')$. The covariance function controls the smoothness of realisations and the degree of shrinkage towards the mean (i.e., regularisation). A common choice is the squared exponential covariance function,

$$k(x,x') = \tau^2 \exp\left(-\frac{|x-x'|^2}{2l^2}\right), \tag{2.9}$$

where τ and l are unknown parameters in the covariance and $|x-x'|^2$ is the squared Euclidean distance between x and x'. Here τ controls the magnitude and l the smoothness of the function; i.e., l is the length scale. In practice one may also estimate an additional parameter, σ^2, to control the variance of zero-centred independent random noise, or 'nugget variance' which can be added to the Gaussian process to represent measurement and other independent errors. This enables a GP conditioned on observations to not have to pass directly through them:

$$\mu \sim GP(m,k) + N(0,\sigma^2). \tag{2.10}$$

In ‘model based geostatistics’ Diggle, Tawn and Moyeed (1998) extended kriging methodology to include situations in which the stochastic variation in the data is known to be non-Gaussian, in the same way that generalized linear models (Nelder and Wedderburn, 1972) extend the classical Gaussian linear model. The key point to make here though is that, whether Gaussian or non-Gaussian, the traditional ‘kriging’ geostatistical approaches require making an assumption of stationarity because the parameters of their covariance function (e.g., τ, l; Equation 2.9) are estimated globally rather than themselves being dependent on position in space (and time). This means that resultant simulations will exhibit the same spatial structure throughout the mapped area, which, depending on the variables being mapped, may well be a rather unconvincing assumption to make. In the case of geological variables, they spatial structure they exhibit is highly variable owing to the presence of differing terranes (containing different types of rock) and dislocational features such as faulting and shearing. To map geological variables in a geologically convincing way therefore requires the use of more flexible models than those to which the term ‘geostatistics’ typically pertains, although these more flexible models will by definition still technically fall within the discipline of ‘geostatistics’.

There are a variety of ways that one could add more ‘flexibility’ into the single-layer Gaussian process regression of traditional geostatics (i.e., kriging). The possibilities fall into two broad categories; either to make the kernel parameters dependent on position in space and time (Heinonen et al., 2016), for example

$$(\tau, l, \sigma^2) = f(x_{s,t}). \tag{2.11}$$

or to warp the space on which the kernel is acting, such that $|x - x'|^2$ is no longer the squared Euclidean distance in true geographic space (map space) but instead a distance in some latent space, or feature space, Z, which is itself a (potentially non-linear) transformation of position in true geographic space (e.g. Calandra et al., 2016; Wilson et al., 2016). This may be achieved using deep Gaussian processes (Damianou and Lawrence, 2013), in which the space of inputs is transformed through one or more Gaussian processes before being passed as the input to a final Gaussian process layer.

While Gaussian processes are perhaps the best way to achieve flexible regression with 'full control' (i.e., a clearly defined prior, although it is difficult to preserve and interpret the 'meaning' of this prior as depth increases), they have a downside in their computational complexity, which makes it unfeasible to use them for modelling the largest of datasets (although efficiency developments such as sparse Gaussian processes — Snelson and Ghahramani, 2005 — do improve the situation).

By contrast, neural networks with their ability to be trained using stochastic gradient descent on modern GPU hardware (for frequentist inference, but also for variational-inference and other Bayesian approximations; Wilson, 2020), are among the most scalable modelling methods available to us, and are famously suitable for flexible regression. It has also been shown that a single-hidden-layer neural network with independent and identically distributed priors over its parameters becomes equivalent to a Gaussian process in the limit of infinite layer width (Neal, 1995), and that this equivalence extends to their deep counterparts (Lee et al., 2017).

Gal and Ghahramani (2016) showed that training deep neural networks using dropout, in which the output of each node is multiplied by a Bernoulli distribution (so that sampling returns either zero [i.e., the node is 'dropped out'] or the node's original output) can be cast as approximate Bayesian inference in deep Gaussian processes. This therefore enables the use of deep neural networks for applications where deep-Gaussian-process-like behaviour is desired, but with lesser computational expense by avoiding the need to compute distances between all pairs of observations, which is required in Gaussian processes in order to compute the covariance $k(x,x')$ between observations.

We make use of this Bayesian deep neural network approximation to deep Gaussian processes, $BNN(x) \approx GP(m,k)$, in the approach we develop in this thesis for Bayesian deep learning for spatial (and spatiotemporal) interpolation in the presence of auxiliary information, and achieve results that suggest that our Bayesian deep learning approach is a suitable candidate for the 'next generation' of geostatistical models that are required in order to model geological variables in a geologically convincing manner.

There remains an open question about whether including a covariance kernel, $k(x,x')$, to explicitly model spatial autocorrelation would bring benefits over the spatial autocorrelation that our Bayesian deep learning approach already implicitly models (see the final figure of Chapter 5's appendix). This could be achieved for example by adopting a deep kernel learning approach (Wilson et al., 2016), by which the deep neural network would be used to transform the space of inputs into a suitable feature space to pass through a Gaussian process as the final layer of the deep model. Alternatively, our current Deep learning architecture could simply be used as the mean function for a single-layer Gaussian process, although this is likely to prove overly restrictive due to the stationarity assumptions of using a fixed-parameter kernel (without feature space warping). In either case, explicit modelling of spatial (and spatiotemporal) covariance will always add computational cost, and so it is interesting to investigate what Bayesian neural networks can achieve without doing so.

A crucial benefit of adopting a deep neural network based approach enables the incorporation of computer vision capabilities using convolutional layers. This brings the capability to extract task-relevant features from gridded auxiliary variables, and thus allows the model's resultant interpolations / simulated maps to capture structure (which in the case of geology may include faulting, if this is alluded to within the covariates) in a way that traditional approaches (e.g., regression kriging; Hengl, Heuvelink and Rossiter, 2007) cannot emulate without requiring prohibitive amounts of manual feature engineering. This benefit may in itself justify what may otherwise be considered a 'downgrade' in using Bayesian neural networks over exact Gaussian processes.

2.3.2 Summary

With the fundamental aims of geological mapping (the prediction of geological variables through space) being quite similar to those of weather forecasting (the prediction of atmospheric variables through space and time), there is potentially a lot of overlap between the two disciplines when viewed through the lens of statistical modelling and machine learning, and both stand to benefit equally from well-implemented probabilistic modelling.

Where the established practice for weather forecasting is to generate forecasts using numerical weather prediction (NWP) based on physical laws, the established practice for geological mapping is to generate maps through a process of mental-modelling and hand-drawing by geologists. The outputs of both can be viewed as 'forecasts' to be post-processed by statistical learning systems, but perhaps the bigger prize is the development of end-to-end statistical learning systems which can ingest the same observations as existing approaches use, and from these learn well-calibrated and sharp predictive distributions (preferably composed of realistic / physically plausible samples). After all, Bayesian data models are dependent on our choice of prior, and the prior knowledge currently used in weather forecasting (i.e., atmospheric physics) and geological mapping (i.e, geological experience), form no exception to this.

From a statistical perspective geological mapping has so far, in the main, failed to become a quantitative modelling practice because of the limitations of traditional geostatistical techniques, and the geologically unrealistic / unconvincing prior assumptions that these have historically required making (e.g., of stationarity, isotropy). With the development of the right methods, there is not reason why a new generation of geostatistical modelling techniques cannot revolutionise geological mapping in a similar manner to how (ensemble) numerical weather prediction has revolutionised (probabilistic) weather forecasting. Excitingly, the requirements of spatio-temporal weather forecast post-processing systems, which must ingest and learn from numerical weather prediction forecast grids (as well as other covariates), are perhaps surprisingly similar to those of spatial geological mapping systems, which must ingest and learn from geologically informative covariate grids. Some of these covariate grids may even be the very same; for example digitial elevation models, and satellite imagery, and so the same statistical modelling approaches may perform well in both tasks (although task-specific tweaks are likely to be beneficial).

Chapter 3

A framework for probabilistic weather forecast post-processing across models and lead times

Forecasting the weather is an increasingly data intensive exercise. Numerical Weather Prediction (NWP) models are becoming more complex, with higher resolutions, and there are increasing numbers of different models in operation. While the forecasting skill of NWP models continues to improve, the number and complexity of these models poses a new challenge for the operational meteorologist: how should the information from all available models, each with their own unique biases and limitations, be combined in order to provide stakeholders with well-calibrated probabilistic forecasts to use in decision making?

In this chapter, we use a road surface temperature example to demonstrate a three-stage framework that uses machine learning to bridge the gap between sets of separate forecasts from NWP models and the 'ideal' forecast for decision support: probabilities of future weather outcomes. First, we use Quantile Regression Forests to learn the error profile of each numerical model, and use these to apply empirically-derived probability distributions to forecasts. Second, we combine these probabilistic forecasts using quantile averaging. Third, we interpolate between the aggregate quantiles in order to generate a full predictive distribution, which we demonstrate has properties suitable for decision support. Our results suggest that this approach provides an effective and operationally viable framework for the cohesive post-processing of weather forecasts across multiple models and lead times to produce a well-calibrated probabilistic output.

3.1 Introduction

The importance of weather forecasting for decision support is likely to increase as we progress into times of changing climate and perhaps more frequent extreme conditions (Rahmstorf and Coumou, 2011). Any methodological developments that can improve our ability to make the optimal decisions in the face of meteorological uncertainty are likely to have a real impact on all areas that utilise weather forecasts.

Since the inception of meteorology as a mathematical science, driven by the likes of Abbe (1901), Bjerknes (1904), and Richardson (1922), numerical modelling has been the core methodology of weather forecasting. In 2015, Bauer, Thorpe and Brunet (2015a) reviewed the progress of numerical forecasting methods in *the quiet revolution of numerical weather prediction*, and explained how improvements in physical process representation, model initialisation, and ensemble forecasting have resulted in average forecast skill improvements equivalent to one day's worth per decade — implying that in 2020 our five day forecasts have approximately the same skill as the one day forecasts of 1980.

However, the continuation of these gains requires ever more computational resources. For example, in pursuit of higher resolution models, halving grid cell length in three dimensions requires eight times the processing power, but due to model biases and initial condition uncertainty, corresponding improvements in forecasting skill are not guaranteed. At the same time, as society progresses we are placing greater emphasis on efficiency and safety in everything we do. In order for businesses to operate efficiently and in order to keep the public safe from meteorological hazards, there should be great emphasis on improving the functionality of weather forecasts as decision support tools — and that means bridging the gap between deterministic NWP model outputs (including sparse ensembles from these) and fully probabilistic forecasting approaches suitable for supporting decision making through the use of decision theory Economou et al., 2016; Simpson et al., 2016. In essence, statistical approaches are key to optimal, transparent, and consistent decision making.

At the same time, while numerical weather prediction methodology has evolved gradually over the last century (hence *'the quiet revolution'*), the last decade has seen significant developments in machine learning and its rise into the scientific limelight, with promising results being demonstrated in a wide range of applications e.g. Gulshan et al., 2016; Silver et al., 2017; Hey et al., 2020. The catalyst for this new wave of machine learning can perhaps be attributed to the results of Krizhevsky, Sutskever and Hinton (2012) in the Large Scale Visual Recognition Challenge (ILSVRC) of 2012, who demonstrated for the first time that deep neural networks — with their ability to automatically learn predictive features in order to maximise an objective function — could outperform existing state-of-the-art image classifiers based on hand-crafted features, which had been the established approach for previous decades. The parallels between the hand-crafted features in image classification, and the human choices that are made in all kinds of data processing pipelines — including weather forecasting — have inspired exploration into new applications of machine learning. In meteorology, could these tools relieve pressure from current model development and data processing bottlenecks and deliver a step-change in the rate of progress in forecasting skill?

Initial efforts using machine learning in the context of post-processing NWP model output have shown promising results (e.g. Rasp and Lerch, 2018; Chapman et al., 2019; Taillardat et al., 2016) in both probabilistic and deterministic settings. We believe that the greatest value of machine learning in weather forecasting lies in the probabilistic capabilities of these methods: not only do they have the potential to learn to improve forecasting skill empirically, but also to bridge the gap between traditionally deterministic forecasting approaches (i.e. numerical weather prediction) and the probabilistic requirements of robust decision support tools.

To this end, in this chapter we demonstrate our framework for probabilistic weather forecast post-processing using machine learning. We have designed this framework to be suitable for use by operational meteorologists, and therefore, unlike other studies that we are currently aware of, our proposed solution incorporates forecast data from all available model solutions (i.e. multiple NWP model types, and all available forecast lead times). The framework aggregates the available forecast information into a single well-calibrated predictive distribution, providing probabilities of weather outcomes for each hour into the future. Our application is road surface temperature forecasting — a univariate output — using archived operational data from the UK Met Office. In this demonstration we use Quantile Regression Forests (QRF, Meinshausen, 2006) as our machine learning algorithm, but hope to convince readers that our overall approach — flexible quantile regression for each forecast, followed by averaging of quantiles across forecasts, and finally interpolating the full predictive distribution — provides a flexible framework for probabilistic weather forecasting, and crucially one that is compatible with the use of any probabilistic forecasting models (post-processed or otherwise).

Our framework can be seen as an overarching aggregator of forecast information, emulating part of the role of the operational meteorologist, who must otherwise develop a sense for how skillful each individual forecast is through experience, and mentally combine these forecasts in order to make probabilistic statements to inform decision making. These include judgements of uncertainty such as a ‘most likely scenario’ and a ‘reasonable worst case scenario’ (Stephens and Cloke, 2014). Figure 3.1 gives an example of how complex a task it is to make sense of the available forecast information, even for the single variable of road surface temperature at a single site.

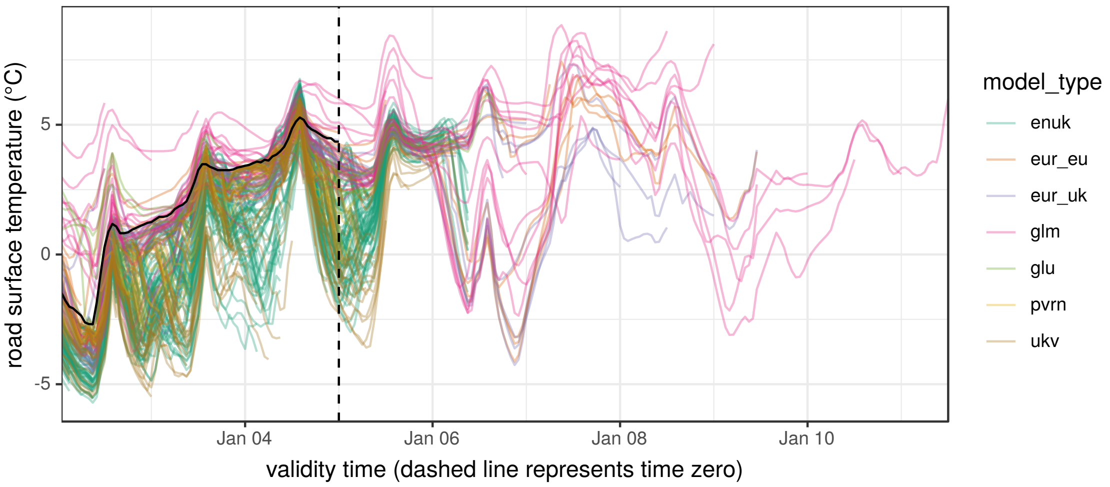


Figure 3.1: A visualisation of the information provided by numerical weather prediction (NWP) forecasts. Each coloured line represents an ensemble member from a different model type. Observations (solid black line) go as far as time zero (vertical dashed line - the 'current time', which is 00:00 on 5th of Jan in this figure) and beyond that, if a statistical approach is not used, it's down to individual meteorologists to determine the likely weather outcomes based on the information presented by the models.

While methods for weather forecast post-processing using more traditional statistical approaches have existed for some time (e.g. Raftery et al., 2005; Glahn and Lowry, 1972; Wilks and Hamill, 2007; Gneiting et al., 2005), we believe our machine learning based approach to be a useful contribution to the field as interest in meteorological machine learning grows. The development of our framework has been guided by the needs of operational weather forecasting, including handling sets of different weather forecasting models with their own unique ranges of lead times. Increasingly these forecasts may not all be raw NWP forecasts, but are themselves likely to have been individually post-processed using machine learning (e.g. for downscaling), or purely statistical spatio-temporal forecasts. It is therefore a strength of our proposed framework that we can post-process any number of models of any type, and for any lead times.

3.2 Post-processing framework

The key considerations in designing our framework were that we wanted to develop an approach that was flexible, compatible, and fast. Flexible in the sense that we would like to minimise the number of assumptions made that would constrain the form of our probabilistic forecasts, and largely ‘let the data do the talking’, as tends to be the machine learning ethos. Compatible in the sense that we would like our framework to generalise to scenarios in which NWP model outputs are not the only forecast available - this is likely to become more common as machine learning becomes more commonplace. And fast, because weather forecasting is a near-real-time activity and any post-processing approach has to be able to keep up.

There are many possible approaches for post processing individual weather forecasts, and indeed many possible approaches for producing forecasts in the first place (for example spatio-temporal statistical models (Hengl et al., 2012), or more recently neural network based approaches (Asanjan et al., 2018), in addition to the traditional NWP models). By using quantiles as the basis on which we combine multiple forecasts, our approach is compatible with any forecast from which well-calibrated predictive quantiles can be obtained, either from the forecast model directly (if probabilistic), or through uncertainty quantification of deterministic models, as we demonstrate in this chapter. The three stages of our framework’s methodology are explained in the following subsections.

3.2.1 From deterministic to probabilistic forecasts

For our application to road surface temperature forecasting, the available forecasts come from a set of NWP models, as is commonly the case. Our model set spans from long range, low resolution global models (glu, glm) through medium range, medium resolution European models (eur_eu, eur_uk) to shorter range, high resolution UK specific models (ukv, enuk) including a six-hour nowcast (pvrn). Apart from the ‘enuk’ model, which itself provides an ensemble of 12 members on each run, the other models provide single deterministic forecasts. While all of these models provide spatial forecasts, in this study we post-process the forecasts for specific sites in order to focus on the probabilistic aspects. Figure 3.1 shows a snapshot of the set of model forecasts for a single site.

While the final output of our framework is a full predictive distribution summarising the information contained in the entire set of NWP model output, the first step is to convert each deterministic forecast into an individually well-calibrated probabilistic forecast. We do this by using machine learning to model the error profile of each deterministic forecast conditional on forecasting covariates. The error is defined as:

$$\varepsilon_{t,m} = y - x_{t,m} \tag{3.1}$$

where $x_{t,m}$ is a NWP model forecast for model type m (e.g. 'eur_uk') and lead time t while y is the corresponding observation. For our surface temperature data, lead times range from 0 hours to 168 hours. Predictions of future data points are then obtained by

$$\hat{y}_{t,m} = x_{t,m} + \varepsilon_{t,m} \tag{3.2}$$

Modelling the forecast errors rather than y was empirically found to produce better predictions using significantly less training data. An explanation for this is that $x_{t,m}$ is used as a complex trend removal function (e.g. for seasonality and other non-stationary effects), thus allowing us to treat $\varepsilon_{t,m}$ as a time-invariant (stationary) variable — the stochastic relationship between bulk model error and lead time is quite stable across absolute time (see Figure 3.2). This simplifying assumption may not hold up in every case, and we would recommend checks before applying it to other variables and forecasting tasks. Modelling the forecast errors, ε, also has the benefit of providing many more unique $\varepsilon_{t,m}$ observations for training than is provided by the absolute temperature observations $y_{t,m}$. This is because, while y_t is identical for all m (only one absolute temperature observation is made per time step), ε is unique for each t,m pair because each unique NWP forecast produces its own unique error. The recent work of Taillardat and Mestre (2020), and Dabernig et al. (2017) before them, shows that we are not alone in successfully using an error modelling approach.

Figure 3.2 shows $\varepsilon_{t,m}$ for $m =$ glm (global long range forecast) and $t = 0,1,\dots,168$. Note the expected general increase in variance with increasing lead times and the increase in the location of the mean of the distribution (red line) indicating a systematic bias in the forecast. There is also a cyclic trend caused by the interaction between lead time and model initialisation time. This particular model is initialised at 00:00 and 12:00 hours, so we see increased errors on a 12 hour cycle starting from initialisation. This is because temperature errors tend to be larger in the early hours of the afternoon (when effects of inaccurately modelled cloud coverage on solar irradiance are most pronounced) compared to the early evening and morning.

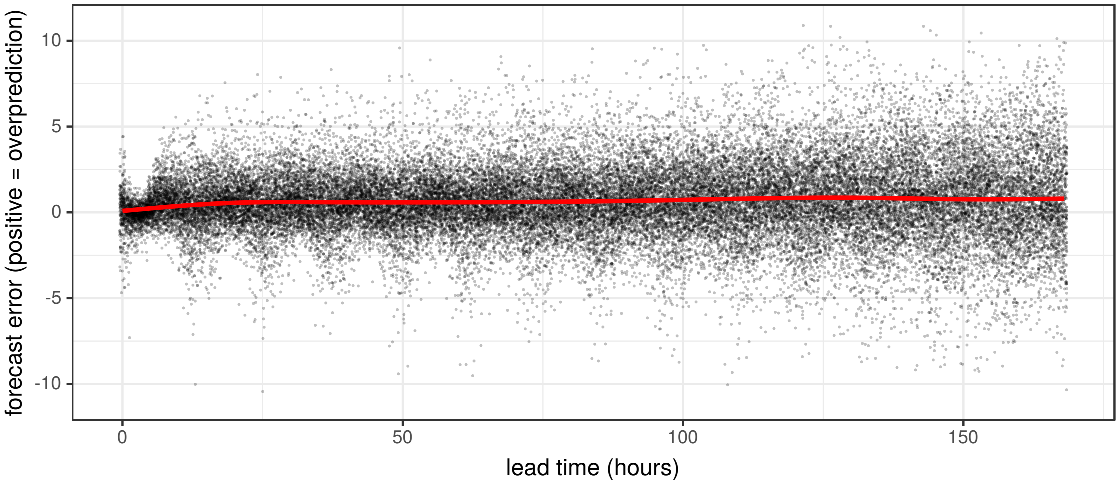


Figure 3.2: Plot of $\varepsilon_{t,m}$ for $m =$ glm against lead hour $(1,2,\ldots,168)$ for a random sample of our dataset (spanning multiple months of absolute time). Each point is $\varepsilon_{t,m}$ at a single hourly time step. The red line is a smooth estimate of the mean.

In order to learn the error distribution of each NWP model type, we use Quantile Regression Forests (QRF, Meinshausen, 2006) as implemented in the 'ranger' package in R (Wright and Ziegler, 2017). While many other data modelling options are possible, QRF has a number of desirable properties. First, it has the flexibility to fit complex functions with minimal assumptions. For data rich problems such as ours, not specifying a parametric distribution allows us to capture the true complexity of the error distribution. Second, it is very fast in both training and prediction, and suitable for operational settings avoiding user input such as convergence checks (e.g. MCMC or gradient descent based methods). Third, it is relatively easy to understand the algorithm and has only a few hyper-parameters to tune, which makes getting reasonably good results in new problems quite straightforward.

For a detailed explanation of the QRF algorithm see Athey, Tibshirani and Wager (2019) or Taillardat et al. (2016) for a more weather oriented description. For regression problems like ours, the QRF algorithm (a variant of the popular random forest algorithm) consists of an ensemble of regression trees. A regression tree recursively partitions the space defined by the covariates into progressively smaller non-overlapping regions. A prediction is then some property/statistic of the observations contained within the relevant region. Conventionally for each tree the prediction is the sample mean of the observations in the partition corresponding to new input data. Suppose for instance that a regression tree is grown on the data in Figure 3.2 and that our aim is to predict the mean forecast error at 100 hours. Suppose also that the tree had decided to group all observations in $t \in [98, 106]$ into the same partition. Then the prediction for $t = 100$ would simply be the mean of all observations between 98 and 106 hours. For a QRF however, the same tree would instead return the values of all the observations between 98 and 106 hours as an empirical distribution from which quantiles are later derived.

The predictive performance of random forests is sensitive to how the covariate space is partitioned. The splitting rule, which governs the placement of partitioning splits as each tree grows, is therefore an important parameter, as are tunable hyper-parameters that we discuss in the next paragraph. Here we use the variance splitting rule, which minimises the intra-partition variance within the two child partitions at each split. A key aspect of the random forest and QRF algorithm is that each tree in the ensemble is grown on its own unique bootstrapped random sample of the training data. This produces a forest of uncorrelated trees, which when aggregated (called bootstrap aggregation or ‘bagging’) results in an overall prediction that is less prone to over-fitting than an individual decision tree, while retaining the ability to learn complex functions. To produce quantile predictions, the QRF returns sample quantiles from all observations contained within the relevant partition of each individual tree in the forest. In doing so it behaves as a conditional (on the covariates) estimate of the CDF.

Extending from the earlier work of Breiman (2001) who proposed the random forest algorithm, Meinshausen (2006) provides explanation of the workings of the quantile regression forest approach as follows (initially the random forest is explained, and then the quantile regression forest). Let θ be the random parameter vector that determines how a tree is grown. The corresponding tree is denoted by $T(\theta)$. Let B be the space in which X lives, that is $X : \Omega \rightarrow B \subseteq \mathbb{R}^p$, where $p \in N_+$ is the dimensionality of the predictor variable. Every leaf $l = 1,...,L$ of a tree corresponds to a rectangular subspace of B. Denote this rectangular subspace by $R_l \subseteq B$ for every leaf $l = 1,...,L$. For every $x \in B$, there is one and only one leaf l such that $x \in R_l$ (corresponding to the leaf that is obtained when dropping x down the tree). Denote this leaf by $l(x,\theta)$ for tree $T(\theta)$.

The prediction of a single tree $T(\theta)$ for a new data point $X = x$ is obtained by averaging over the observed values in leaf $l(x,\theta)$. Let the weight vector $w_i(x,\theta$ be given by a positive constant if observation X_i is part of leaf $l(x,\theta)$ and 0 if it is not. The weights sum to one, and thus

$$w_i(x,\theta) = \frac{{}^{1}\{X_i \in R_{l(x,\theta)}\}}{\#\{j : X_j \in R_{l(x,\theta)}\}} \tag{3.3}$$

The prediction of a single tree, given covariate $X = X$, is then the weighted average of the original observations $Y_i, i = 1,...,n$,

$$\text{single tree: } \hat{\mu}(x) = \sum_{i=1}^{n} w_i(x,\theta) Y_i. \tag{3.4}$$

Using random forests, the conditional mean $E(Y|X = x)$ is approximated by the averaged prediction of k single trees, each constructed with an i.i.d. vector$\theta_t, t = 1,...,k$. Let $w_i(x)$ be the average of $w_i(\theta)$ over this collection of trees,

$$w_i(x) = k^{-1} \sum_{t=1}^{k} w_i(x, \theta_t). \tag{3.5}$$

The prediction of random forests in then

$$\text{Random Forests: } \hat{\mu}(x) = \sum_{n=1}^{n} w_i(x) Y_i. \tag{3.6}$$

The approximation of the conditional mean of Y, given $X = x$, is thus given by a weighted sum over all observations. The weights vary with the covariate $X = x$ and tend to be large for those $i \in \{1, ..., n\}$ where the conditional distribution of Y, given $X = Xi$, is similar to the conditional distribution of Y, given $X = x$.

One could expect that the weighted observations may deliver not only a good approximation of the conditional mean, but also to the full conditional distribution. The conditional distribution function of Y, given $X = x$, is given by

$$F(y|X = x) = P(Y \leq y|X = x) = E(1_{\{Y \leq y\}}|X = x). \tag{3.7}$$

The last expression is suited to draw analogies with the random forest approximation of the conditional mean $E(Y|X = x)$. Just as $E(Y|X = x)$ is approximated by a weighted mean over the observations of Y, define an approximation to $E(1_{\{Y \leq y\}}|X = x)$ by the weighted mean over the observations of $1_{\{Y \leq y\}}$,

$$F(y|X = x) = \sum_{n=1}^{n} w_i(x) 1_{\{Y_i \leq y\}}, \tag{3.8}$$

using the weights $w_i(x)$ as for random forests, defined in (3.5). This approximation is at the heart of the quantile regression forest algorithm.

Pseudocode The algorithm for computing the estimate $\hat{F}(y|X = x)$ can be summarised as:

(a) Grow k trees $T(\theta_t), t = 1, ..., k$, as in random forests. However, for every tree, take note of all the observations in this leaf, not just their average.

(b) For a given $X = x$, drop x down all trees. Compute the weight $w_i(x, \theta_t)$ of observation $i \in \{1, ..., n\}$ for every tree as in (3.3). Compute weight $w_i(x)$ for every observation $i \in \{1, ..., n\}$ as an average over $w_i(x, \theta_t), t = 1, ..., k$, as in (3.5).

(c) Compute the estimate of the distribution function as in (3.8) for all $y \in \mathbb{R}$, using the weights from Step b).

For modelling NWP surface temperature errors, the tuning of QRF hyper-parameters as well as the selection of input covariates was conducted manually with the aim of achieving good out-of-bag quantile coverage (a QRF proxy for out-of-sample performance) across all lead times. This was achieved using visual checks such as Figure 3.3, which indicates that on average, prediction intervals are close to the ideal coverage across lead times, i.e. 90% of the time observations will fall within the 90% prediction interval. However for operational setups it may be preferable to use a more formal optimisation procedure, such as Bayesian optimisation. We found that using just lead time, t, and model type, m, as covariates gave the best calibration results, presumably aided by the parsimonious nature of this simple representation. The chosen hyper-parameters were: mtry = 1 (this is the number of covariates made available at random to try at each split), min.node.size = 1 (this limits the size of the terminal nodes / final partitions of each tree - in this case there is no limit on how small these can be), sample.fraction = 128/nrow(training data) (this is the size of the bootstrap sample of the training data provided to each tree), and num.trees = 250 (this is the number of trees in the forest). The use of a relatively small sample size (128 observations for each tree, out of a total of around 50,000 observations in a 14 day run-in period) and a minimum node size of one (trees grown to full depth) was found to produce the best out-of-bag coverage at a minimal run time. Our mtry setting meant that one of our two covariates (t and m) was made available at random to each tree at each split. If another objective had been prioritised (e.g. to minimise mean squared error, rather than optimise coverage) the optimal hyper-parameters would be different.

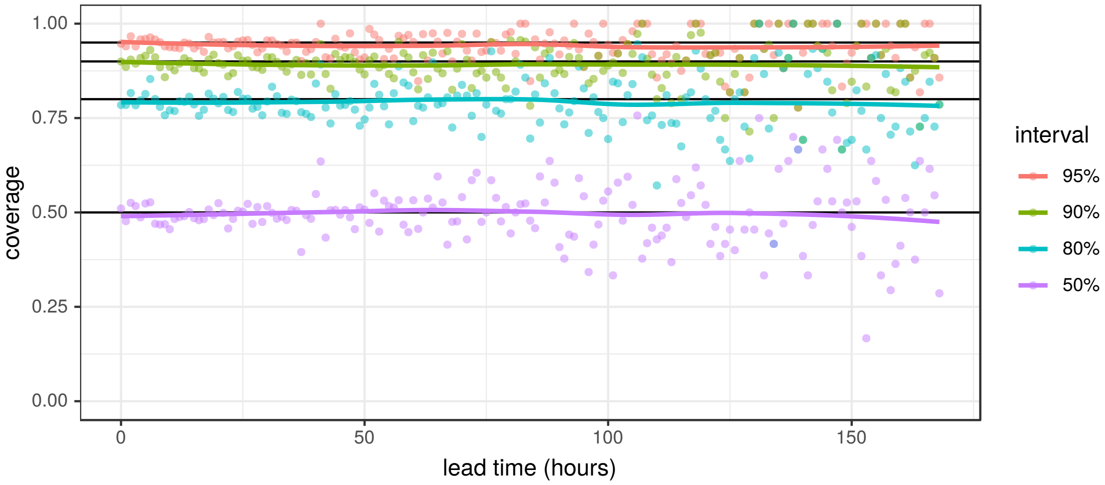


Figure 3.3: Coverage of the 50%, 80%, 90%, and 95% QRF prediction intervals on out-of-bag data from one training scenario (though the picture is indicative of other scenarios). The coverage is the proportion of observations that fall within each prediction interval, and should match the interval (i.e. with 95% of observations falling within the 95% prediction interval) in a well-calibrated setup.

Once the QRF has been trained, each NWP forecast can be converted to a probabilistic forecast by adding to it the predicted error distribution (3.2). Unlike the deterministic NWP forecast, the prediction is now a probability distribution, constructed through a conditional bootstrap of $\varepsilon_{t,m}$ via the QRF algorithm. Prediction intervals are obtained as quantiles of this distribution as illustrated in Figure 3.4.

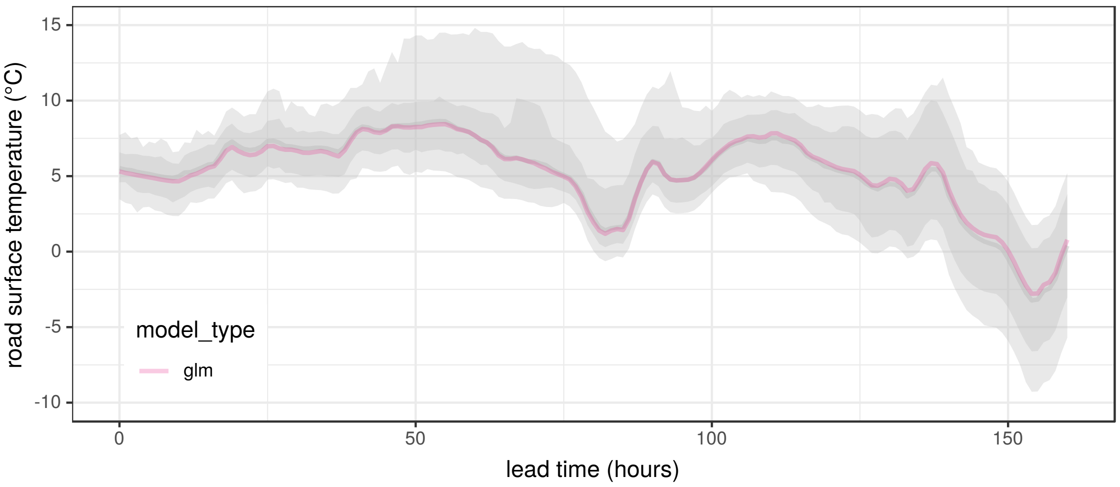


Figure 3.4: A deterministic NWP forecast for $m =$ glm that has been converted to a probabilistic forecast using equation (3.2). The 80% and 95% prediction intervals are shown as overlain grey ribbons, while the solid grey line is the median (which differs little from the NWP forecast here).

3.2.2 Combining probabilistic forecasts

The next step is to combine these predictive distributions from each NWP model output into a single distribution that is suitable for use in decision support. The challenge is to combine the forecasts in a probabilistically coherent manner, with the goal of producing a single well-calibrated and skillful predictive distribution.

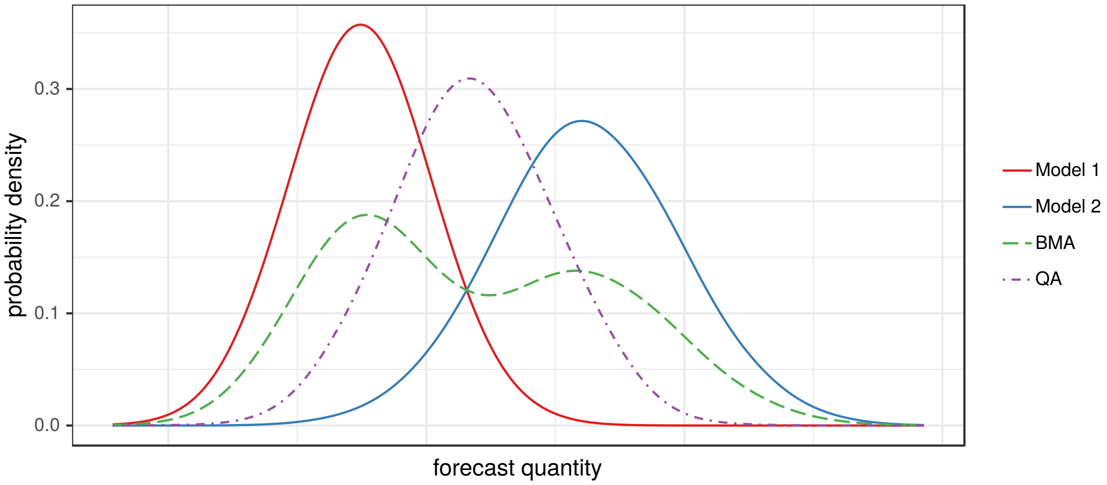


Figure 3.5: Synthetic example of combining two probabilistic forecasts using Bayesian Model Averaging (BMA) and Quantile Averaging (QA), after Schepen and Wang, 2015.

A popular approach for combining probabilistic models is Bayesian Model Averaging (BMA), and its use in the statistical post-processing of weather forecasts has precedent (e.g. Hoeting et al., 1999; Raftery et al., 2005; Gneiting and Ranjan, 2013). Basic BMA produces a combined distribution as a weighted sum of PDFs. However, in order to satisfy the requirements of our framework, we propose an alternative approach using quantile averaging, whereby each quantile of the combined distribution is taken as the mean of the same quantile estimated by each individual model. An illustrative comparison of equal-weighted BMA and quantile averaging is shown in (Figure 3.5). For the purposes of our framework, we found BMA to be unsuitable for the following three reasons: 1) Achieving good calibration of the combined distribution produced by BMA requires optimisation of the intra-model variance, i.e. the spread of each individual model's error profile. In our case, where each model's error profile has been learned independently by QRF, and is already well-calibrated, combining these through BMA produces an over-dispersed predictive distribution due to the inclusion of the inter-model variance in addition to the already calibrated intra-model variances. 2) In turn, this makes BMA rather incompatible with input models that are individually well-calibrated (e.g. statistical nowcasts), and therefore incompatible with a general framework like ours. 3) The use of BMA across all models and lead times is complicated by the fact that there are not an equal number of forecasts available for each lead time. This means that the inter-model variance is intrinsically inconsistent across lead times, even dropping to zero at our longest ranges, where only a single deterministic forecast is available (e.g. Figure 3.1). This decrease in inter-model variance with increasing forecast range trends opposite to the true uncertainty, which intuitively should on average increase with forecast range. This is a quirk of NWP forecast availability and one that probabilistic post-processing must overcome.

Our framework overcomes this instability in inter-model variance by using quantile averaging (also known as the 'Vincentization' method (Genest, 1992; Vincent, 1912) to combine forecasts that are already well-calibrated for coverage (owing to their QRF error profiles, in our case). Using this approach, we construct our combined forecast distribution from the quantile predictions of our individual QRF post-processed forecasts. To produce each predicted quantile of the combined distribution, Vincentization simply takes the mean of the set of estimates of the same quantile by each individual forecast. As explored by Ratcliff (1979), Vincentization produces a combined distribution with mean, variance, and shape all approximately equal to the average mean, variance, and shape of the individual distributions (as we see in Figure 3.5). Vincentization therefore provides similar functionality to parameter averaging of parametric distributions, but for non-parametric distributions such as ours. Within our framework, Vincentization effectively integrates out the inter-model variance (by taking the mean across models), and in doing so preserves the calibration of the individual QRF post-processed forecasts, avoiding the underdispersion issues that would be introduced by BMA in this situation, since each individual forecasts is already pre-calibrated by the QRF post-processing step, and so including the inter-model variance in the model blending step would overinflate the total variance of the combined predictive distribution. Vincentization, quantile averagin, is therefore one possible solution to the issue of combining pre-calibrated probability distributions without loss of calibration Gneiting and Ranjan, 2013, and has previously been found capable of providing sharper and better calibrated forecasts (Lichtendahl Jr, Grushka-Cockayne and Winkler, 2013). However, the method by which probability distributions are combined can have important implications for decision-support forecasting, and while quantile averaging satisfies our general requirements for this framework, we do not discount that alternative approaches may be preferable depending on the application.

Our quantile averaged forecast benefits from stability owing to the law of large numbers — any quantile of the forecast distribution represents an average of the estimates of that quantile across the available individual forecasts. This approach is therefore more akin to model stacking procedures, as used in ensemble machine learning to improve prediction accuracy by reducing prediction variance (Ren, Zhang and Suganthan, 2016). Indeed, this same logic is behind the bootstrap aggregation (‘bagging’) procedure of the random forest algorithm: by averaging the predictions of multiple individual predictors — each providing a different perspective on the same problem — the variance of the aggregate prediction is reduced, resulting in improved prediction accuracy at the expense of some increased bias (Belkin et al., 2019). Crucially for our framework, unlike a BMA approach which retains the inter-model variance, the calibration of our quantile averaged output is invariant to the number of forecasts available at each timestep. This is key for temporally coherent forecast calibration across all lead times.

Our error modelling approach does require one extra-step of processing in order to handle model types which themselves have multiple interchangeable ensemble members. The 'enuk' model (Figure 3.1) is our example of this, having twelve non-unique members. In such cases, the apparent error profile for the model type as a collective gets overinflated by the inter-member variance. Our solution to this is to label each ensemble member by its rank (at each time step). This splits our 12-member 'enuk' ensemble into 12 unique model types in the eyes of the QRF. This approach produces well-calibrated error profiles (though with significant offset bias in the extreme ranking members, as would be expected).

3.2.3 Simulation from the full predictive distribution

While quantile averaging provides an effective way of combining multiple probabilistic forecast distributions, it leaves us with only a set of quantiles rather than the full predictive distribution. This distribution is desirable because it allows us to (a) answer important questions such as 'what is the probability that the temperature will be below $0°$C?' and (b) evaluate the skill of the probabilistic forecast using a range of proper scoring rules (although, depending on the end use, some proper scoring rules could be calculated directly from quantile predictions, e.g. the quantile score Bentzien and Friederichs, 2014 or the interval score Gneiting and Raftery, 2007).

To obtain the full predictive distribution, we interpolate between the quantiles of our combined forecast in order to construct a full CDF using the method of Quiñonero-Candela et al. (2006), which has previously been applied to precipitation

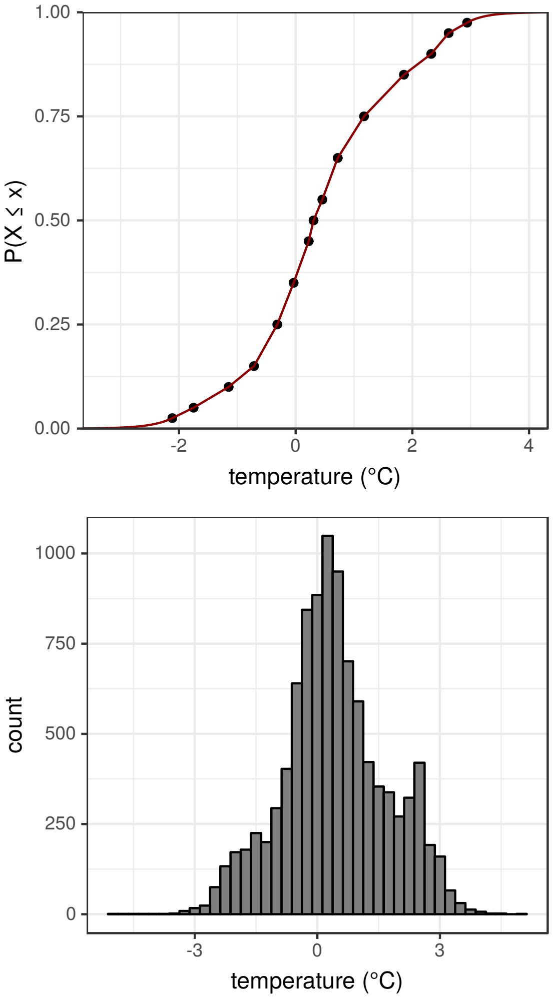


Figure 3.6: Interpolated CDF of the combined predictive distribution (top), and corresponding road surface temperature simulation (bottom) for a particular 50-hour ahead forecast.

forecasting Cannon, 2011 and is available in the R package qrnn Cannon, 2011. The method linearly interpolates between the given quantiles of the CDF (our combined quantiles from Vincentization), and, beyond the range of given quantiles, extrapolates down to $P(X \leq x) = 0$ and up to $P(X \leq x) = 1$ assuming tails that decay exponentially with a rate that ensures the corresponding PDF sums to one (Figure 3.6 top, for details see pages 8 and 9 of Quiñonero-Candela et al. (2006)). Using this approach allows us to construct a full predictive distribution from the Vincentized quantiles of our individual QRF post-processed forecasts. Depending on the application at hand, suitable forecast information might be obtained by querying the CDF of the predictive distribution directly at each time step, but in our application here, we go the extra step of simulating temperature outcomes at each timestep by randomly sampling from the CDF (Figure 3.6 bottom). This is the final step of our framework — taking us from a set of disparate NWP forecasts to a full predictive distribution of weather outcomes.

3.3 Results

To evaluate our framework, we applied it to 200 randomly time-sliced and site-specific forecasting scenarios extracted from our UK Met Office road surface temperature dataset, which we have aggregated to hourly time steps. Each scenario has its own training window of 14 days, providing approximately 50 000 data points of $\varepsilon_{t,m}$ to train the QRF, immediately followed by its own evaluation window extending as far as the longest range NWP forecast (up to 168 hours / 7 days), which is akin to the area to the right of the vertical dashed line in Figure 3.1. While there are only 336 hours in a 14 day training window, the number of NWP models and their regular re-initialisation schedule, means that approximately 150 forecasts are made for any hour by the time it is observed. While we only use the current forecasts from each model type to generate our predictions, the training benefits from every historical forecast within the window.

Figure 3.7 shows an example prediction of up to 168 hours into the future for a particular scenario. This is just one of the 200 random scenarios used in our overall evaluation. Although the prediction at each hour ahead is a full probability distribution, here we present prediction intervals as well as a simulation of 1000 temperature values from it. The samples were used to derive the probability of the temperature being below $0°$C as the proportion of values less than zero. Different stakeholders will require their own unique predictive quantities, and by providing a full predictive distribution, our framework should cater for a wide variety of requirements.

Various metrics could be used to evaluate the skill of our probabilistic forecasts over multiple scenario runs. From the perspective of decision support, the ideal metric to evaluate would be the change in loss resulting from using our forecasts to make real world decisions, such as about when to grit roads in our case.

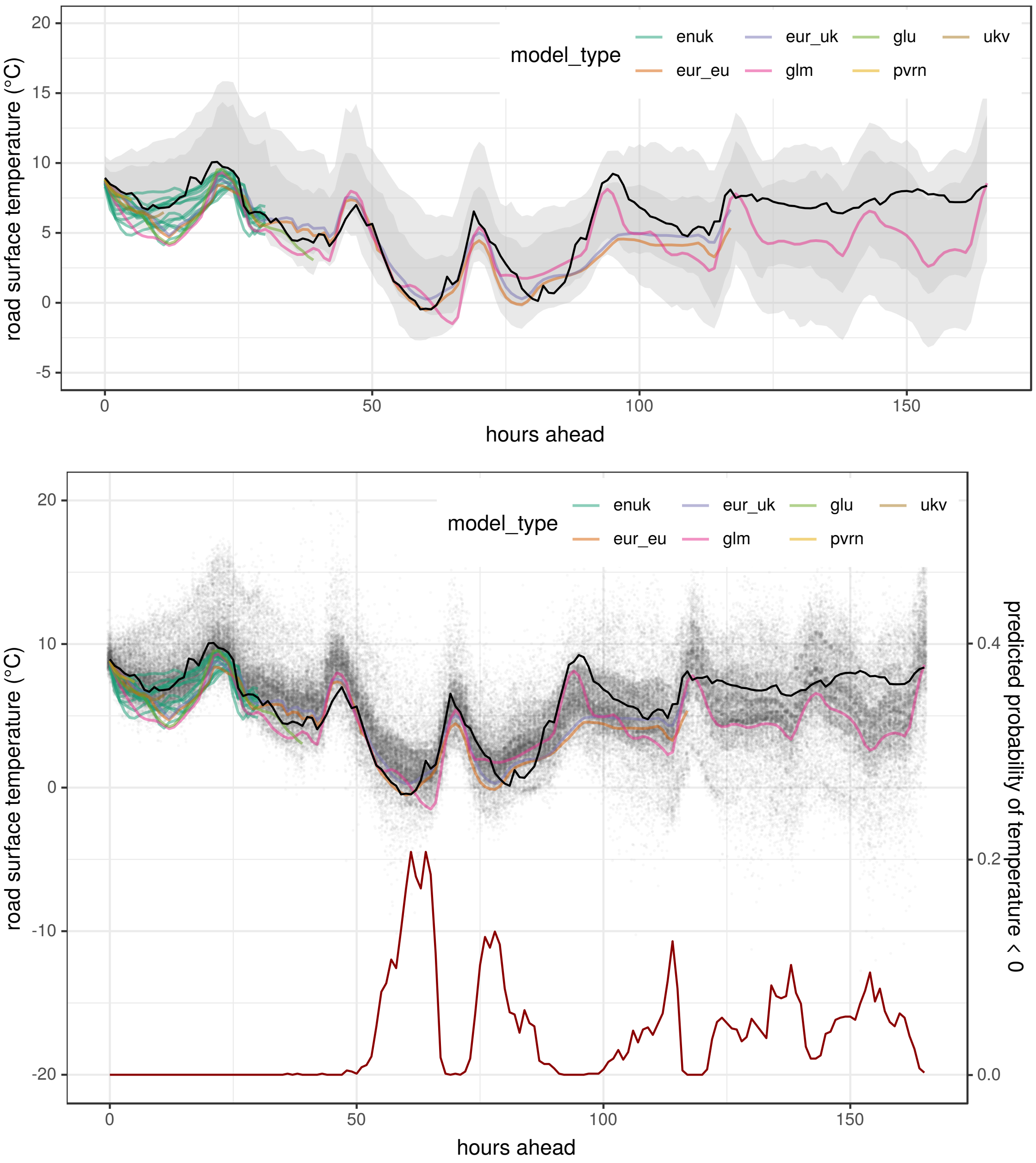


Figure 3.7: An example of the output of our post-processing framework. Top: the probabilistic forecast is visualised by the 80% and 95% prediction intervals. Bottom: simulations from the full predictive distribution as grey dots, while the red line (right-hand y-axis) shows the probability of temperature being $< 0°$C. NWP model forecasts are shown by coloured lines, and the true observed temperature (not known at time of forecasting) is shown by a solid black line.

However, in the interest of a more general analysis we use a range of standard metrics. These are: prediction interval coverage (Figure 3.8 left), the mean-absolute-error (MAE) of the median (Figure 3.8 right, because sometimes a single ‘best’ deterministic forecast is still desired), as well as the continuous ranked probability score (CRPS) and logarithmic score of our probabilistic forecast (both in Figure 3.9).

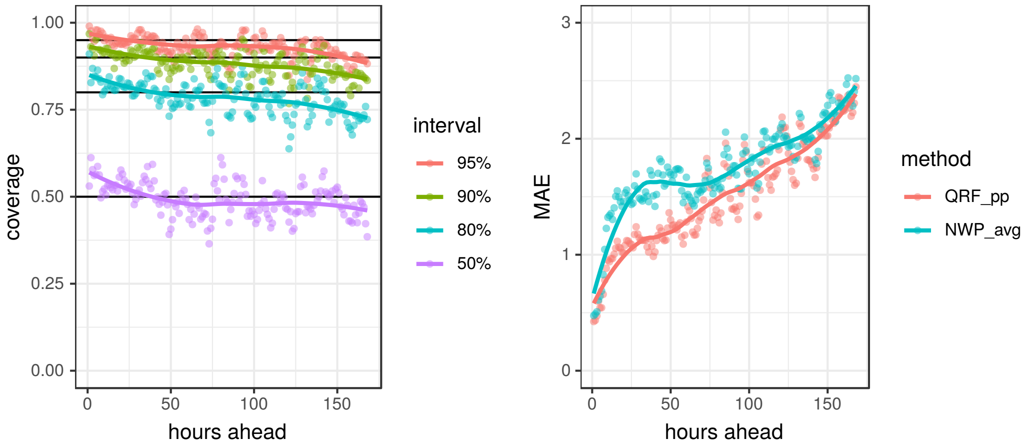


Figure 3.8: Evaluation metrics from 200 forecast scenarios. On the left, coverage of the prediction intervals of the combined probabilistic forecasts. On the right, the MAE achieved by the median of the combined probabilistic forecast (QRF_pp) compared to taking the median of the available NWP forecasts (NWP_avg).

Figure 3.8 indicates that coverage is good overall, with 94.7% of observations falling within the 95% prediction interval, although there is some over-dispersion of our forecast at the shortest ranges and under-dispersion at the longest ranges. This is an indication that, despite producing near perfect results on out-of-bag training data (Figure 3.3), the QRF performance diminishes slightly when applied to new data. The range dependent over- and under-dispersion may be due to the partitioning process on which the forest is grown - by necessity the partitions that represent the extremes of forecast range must extend some distance towards the middle of the range, and in doing so end up capturing an empirical error distribution that is slightly biased towards the average empirical error distribution, rather than perfectly representing the distribution at the extremes of covariates. It may be the case that other data modelling approaches could do better in this respect.

Although deterministic performance was not our focus, the QRF median prediction does outperform the median of the available NWP models across the entire forecast range in terms of MAE. While only a conceptual benchmark, this can be taken as some indication that we have not ‘thrown away’ deterministic performance in pursuit of probabilistic calibration. Figure 3.8 also indicates that our method results in a monotonically increasing error with forecast range, unlike the median of the original NWP forecasts. Similarly, we see a monotonic increase in both the

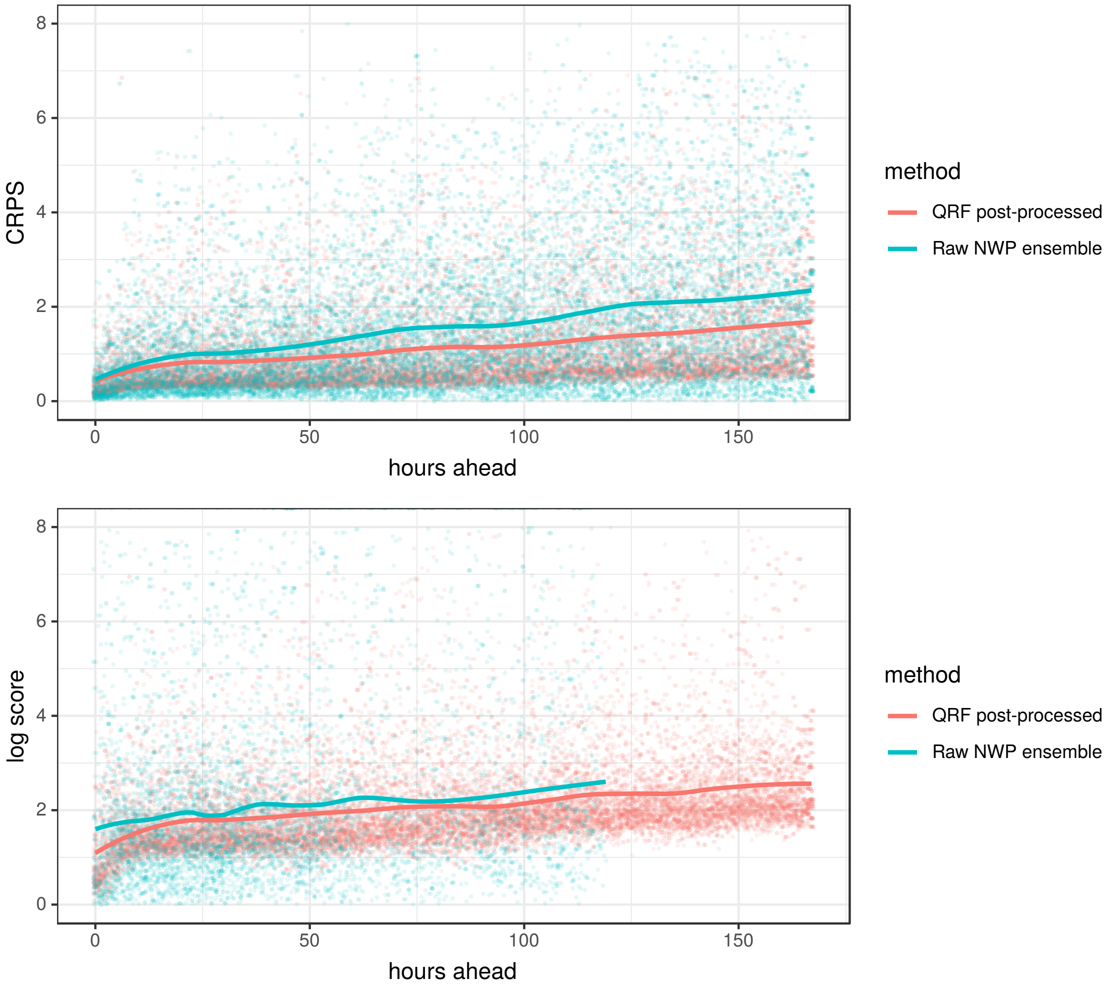


Figure 3.9: Evaluation metrics of our post-processing framework across all lead times on 200 random forecast scenarios. We compare our QRF post-processed output to the raw NWP ensemble in terms of continuous ranked probability score (CRPS, top) and logarithmic score (bottom).

CRPS and the logarithmic score with increasing forecast range (Figure 3.9, top and bottom), and, when compared to the performance of the raw NWP ensemble on the same metrics, find our QRF post-processing approach to perform better. In the case of CRPS, our QRF post-processing approach reduces the rate at which forecasting skill decreases with forecast range. Also, by looking at the spread of performance across individual forecasting scenarios (represented by individual points in Figure 3.9, rather than the lines, which trace the mean) we can see that our QRF post-processing approach reduces the variance in forecasting skill across different forecasting scenarios, making it a more consistent forecast than raw NWP. In the case of logarithmic score (Figure 3.9, bottom) we see again that the forecasting skill provided by the QRF post-processed output is more consistent than that of the raw NWP ensemble, although the difference in the mean performance is less pronounced. The logarithmic score of the NWP ensemble cannot be obtained at longer ranges as only a single deterministic forecast is available. The authors recognise that comprehensive comparisons of our approach to other probabilistic post-processing approaches (in addition to raw NWP output) will be important to consider when choosing the best approach for any operational setup. While we do not offer such comparisons in this study, we have made our dataset openly accessible as one of several benchmark datasets compiled by Chapman et al. (n.d.) and Haupt et al. (2021) at `https://doi.org/10.6075/J08S4NDM` in the hope that it will facilitate comparison of different post-processing approaches on common benchmarks in the future. In terms of speed, training the QRF for each forecast scenario takes between just three and four seconds on an i7-8550U laptop, and so the implementation of this framework can be expected to add very little overhead to a typical operational NWP forecasting setup.

3.4 Discussion and conclusions

The conversion of disparate forecasts into a cohesive probabilistic output is important. A key function of weather forecasts is to support decision making, but current numerical methods do not provide the well-calibrated probabilistic output required to do this rigorously. By applying our framework we compensate for this shortcoming, effectively supplementing forecasts with information from their historic performance in order to combine all available deterministic inputs, for all lead times, into a single well-calibrated probabilistic forecast. Whilst our approach is by no means the first to provide probabilistic post-processing of weather forecasts, we believe the flexibility and speed provided by our use of machine learning, along with our framework's relative simplicity and ability to simultaneously deal with all available models and lead times, makes it a strong option for consideration in operational forecasting settings.

In this study we have only applied our framework to site specific forecasting, but there are no fundamental reasons why the same principles cannot be applied to spatial forecasting by providing the QRF with additional spatial covariates against which to learn its error profiles, or by adopting the standardised anomaly model output statistics (SAMOS) approach as proposed by Dabernig et al. (2017). The error modelling approach that we use seems a very effective way of minimising the amount of training data required compared to predicting absolute values. Taillardat et al. (2016), who also make use of QRF in their post-processing, initially used four years of training data for their absolute value forecasting system in 2016, but have since adopted an error modelling approach themselves (Taillardat and Mestre, 2020).

There are still several aspects of our framework that are open to further investigation. One significant aspect that we explored in preliminary experiments but have not included in our methodology here, is the opportunity to use weighted quantile averaging for combining forecasts. In our setup, where all of the inputs are recent NWP forecasts (and therefore similarly skillful), we saw negligible difference in using a weighted averaging approach, but in situations where more diverse forecast types are in use, it may prove beneficial to assign weightings according to forecast skill. A dynamic weighting approach also enables individual models to be updated without jeopardising the overall post-processed output, as the contribution of the new or updated model will be minimal until it's error profile is well understood. The QRF algorithm provides a convenient means by which skill can be estimated ahead of time, in the form of out-of-bag metrics. For example, we showed earlier the out-of-bag coverage of our trained QRF (Figure 3.3). Metrics such as the CRPS, logarithmic score, and Kullback–Leibler divergence would provide good comparisons of forecast skill on which to base quantile averaging weight, although their calculation would add some additional processing time. Yao et al. (2018) provide more detail about using such metrics for weighted model stacking, and in fact these weights can be optimised as an additional supervised learning problem (Ren, Zhang and Suganthan, 2016).

The overall strategy for combining forecasts is also open to further research. Because it retains the inter-model variance, BMA may be considered to provide a better representation of extreme outcomes at the expense of well-calibrated coverage (at least in setups where each input forecast is already well-calibrated, which is likely to become the norm). We also think that the output of BMA would be difficult to make use of in practice when applied across all lead times as in our framework, because of the discrepancy in the number of models available at each time step, and therefore the spurious inconsistency of the inter-model variance across the forecast range. Still, applications where capturing extremes is a priority may wish to investigate further. For general purposes, we are satisfied with our time-consistent and calibration-preserving quantile averaging approach.

It is our belief that, as time goes on, and the number of different forecasting models in use — along with their complexity and resolution — continues to increase, there will be increasing need for algorithmic interfaces such as ours to summarise the otherwise overwhelming sea of forecast information into decision ready output. This would consist of optimally well-calibrated probabilities of future weather outcomes given all available information. Probabilistic machine learning is a technology that can enable this, and we hope that the work we have demonstrated here will go some way in aiding progression towards this goal.

3.5 Code, data, and acknowledgements

Our dataset has been made available with permission from the Met Office and Highways England, for which we are grateful. It is available for download along with several other open weather forecast post-processing datasets collated by Chapman et al. (n.d.) and Haupt et al. (2021) at `https://doi.org/10.6075/J08S4NDM`. In addition, the code for this study can be accessed at `https://github.com/charliekirkwood/mlpostprocessing`

The lead author is grateful for the insightful discussions and community feedback that came from attending the Machine Learning for Weather and Climate

Modelling conference at the University of Oxford in 2019, and the EUMETNET Workshop on Artificial Intelligence for Weather and Climate at the Royal Meteorological Institute of Belgium in 2020. In addition CK would like to thank Thomas Voigt for offering his data engineering expertise. We are grateful to all who have helped to guide this study.

3.6 Bridge into Chapter 4 and beyond

To go beyond site-specific weather forecast post-processing requires adopting spatial methods, which can model a forecast predictive distribution through space (as well as through time). The setup of the numerical weather prediction post-processing problem; in which the *gridded* outputs of numerical models are clearly the main source of information, motivates a desire to develop statistical approaches which can ingest and learn from gridded inputs. In order to develop such approaches, we have been inspired by methods in computer vision (i.e., convolutional neural networks) and in fact incorporate these into our models. While full spatio-temporal post-processing of ensemble weather forecasts could be one of the ultimate goals of the development of these hybrid geostatistical-computer-vision methods, over the next two chapters we make use of a complex static spatial dataset (observations from the British Geological Survey's geochemical baseline survey of the environment; Johnson et al., 2005) as a test bed for developing the spatial modelling capabilities of our the hybrid geostatistical-computer-vision method that we develop over these chapters. We then apply the approach, with some modifications for outlier filtering, to spatio-temporal atmospheric data — crowd-sourced temperature observations from the Met Office's Weather Observation Website — in Chapter 6, before returning to the lithosphere in Chapter 7 to demonstrate our refined hybrid geostatistical-computer-vision method as a suitable foundation for a probabilistic revolution in geological mapping.

Thus, the model development research that this thesis presents combines inspiration from ensemble weather prediction, computer vision, and geostatistics in order to realise what could be considered a new generation of general probabilistic environmental AI methods, from which many branches of future research directions spring up - both in terms of real-world use and theoretical developments. We summarise these in Chapter 8; Summary and conclusion.

Chapter 4

Deep covariate-learning: Optimising information extraction from terrain texture for geostatistical modelling applications

Where data is available, it is desirable in geostatistical modelling to make use of additional covariates, for example terrain data, in order to improve prediction accuracy in the modelling task. While elevation itself may be important, additional explanatory power for any given problem can be sought (but not necessarily found) by filtering digital elevation models to extract higher-order derivatives such as slope angles, curvatures, and roughness. In essence, it would be beneficial to extract as much task-relevant information as possible from the elevation grid. However, given the complexities of the natural world, chance dictates that the use of 'off-the-shelf' filters is unlikely to derive covariates that provide strong explanatory power to the target variable at hand, and any attempt to manually design informative covariates is likely to be a trial-and-error process — not optimal.

In this chapter we present a solution to this problem in the form of a deep learning approach to automatically deriving optimal task-specific terrain texture covariates from a standard SRTM 90m gridded digital elevation model (DEM). For our target variables we use point-sampled geochemical data from the British Geological Survey: concentrations of potassium, calcium and arsenic in stream sediments. We find that our deep learning approach produces covariates for geostatistical modelling that have surprisingly strong explanatory power on their own, with R^2 values around 0.6 for all three elements (with arsenic on the log scale). These results are achieved without the neural network being provided with easting, northing, and absolute elevation as inputs, and purely reflect the capacity of our deep neural network to extract task-specific information from terrain texture alone. By visualising our deep-learned covariates as geographic maps, we can see that complex but general features of the surface environment and the subsurface are being captured. We hope that these results will contribute to further investigation into the capabilities of deep learning within geostatistical applications.

4.1 Introduction

Since it's inception in 1951 by mining engineer Danie Krige (Krige, 1951), the 'kriging' method has largely defined the field of geostatistics. In the beginning, kriging was a purely spatial model, utilising only the spatial autocorrelation of the target variable in order to make new predictions. The underlying logic is perhaps best summed up by Tobler's third law of geography: that "everything is related to everything else, but near things are more related than distant things" (Tobler, 1970). Kriging worked very well for its original purpose of interpolating gold grades in mines, where additional data was not available. In subsequent decades, as more data-rich problems began to be tackled, kriging evolved to include the ability to handle additional covariates in the model. There have been several somewhat muddled incarnations along the way (i.e., universal kriging; Matheron, 1969, regression kriging; Odeh, McBratney and Chittleborough, 1995, kriging with external drift; Hudson and Wackernagel, 1994) but, as has been well explained by Murray Lark (Lark, 2012), in 1999 Michael Stein (Stein, 1999) brought mathematical clarity to the situation. Stein pointed out that all varieties of kriging that aim to minimise the mean squared error (MSE) can be considered as forms of the empirical best linear unbiased predictor (or BLUP) based on the linear mixed model:

$$\mathbf{Z} = \mathbf{X}\boldsymbol{\tau} + \mathbf{u} + \boldsymbol{\varepsilon}, \tag{4.1}$$

where $\mathbf{Z}$ is a random vector corresponding to the target variable at n sites, $\mathbf{X}$ is an $n \times p$ design matrix, containing the values of any covariates, $\boldsymbol{\tau}$ are the corresponding fixed effects coefficients, $\mathbf{u}$ is a spatially correlated random variable (Gaussian process), and $\boldsymbol{\varepsilon}$ is an independently and identically distributed random variable. This formulation was significant for geostatistics as it enabled parameter estimation by maximum likelihood, with corresponding improvements over the previous method-of-moments approach (Lark, 2000). For our purposes, the formulation is significant because it allows us to separate the ‘regression on covariates’ component, $\mathbf{X}\boldsymbol{\tau}$ — on which we focus — from the spatial, u, and noise, $\boldsymbol{\varepsilon}$, components of this definitive geostatistical model. As mentioned in Chapter 2, Diggle, Tawn and Moyeed (1998) showed how kriging could also be extended to cater for non-Gaussianity (similarly to how generalised linear models extend classical Gaussian linear models) and how Bayesian inference via MCMC methods could be used to incorporate parameter uncertainty - a topic we cover in more detail in the context of deep learning in Chapter 5.

In general, obtaining measurements of the target variable for any geostatistical application is difficult. It could be said that this is the reason geostatistical models are required in the first place — if we could easily observe our target variable at any point in space, we would have little need for statistical models. At the same time, the progression of technology has lead to a vast increase in data availability in general. In the geosciences, the rise of remote sensing means that multispectral satellite imagery is readily available for the entire globe, along with elevation data (which we make use in this study), and many countries have now conducted some form of airborne geophysical survey to provide gravity, magnetic, and radiometric measurements in continuous gridded format. Although these datasets tend not to measure our target variables directly, they may contain information that can contribute to the ‘regression on covariates’ component of the typical statistical model, $\mathbf{X}\boldsymbol{\tau}$. But are we making the most of the information they provide? The typical geostatistical model, as formulated in Equation 4.1, is restricted to only being capable of capturing linear relationships between any provided covariates and the target variable. This means that the typical geostatistical approach to utilising remote sensing data has previously always had to involve manually post-processing gridded datasets in order to derive new covariates that we hope will be informative for the task at hand (i.e. that will display a linear relationship with the target variable).

In the case of terrain analysis, it is common to use a set of standard filters in order to obtain derivatives such as slope aspect, curvature and roughness. The Topographic Roughness Index (TRI) (Riley, DeGloria and Elliot, 1999) for example, has been used to identify landslides (Berti, Corsini and Daehne, 2013), model forest fire return levels (Stambaugh and Guyette, 2008) and map emerging bedrock in eroding landscapes (Milodowski, Mudd and Mitchard, 2015), among other applications. But are we to believe that the TRI provides optimal explanatory power from the terrain to any of those tasks? The range of applications that have made use of 'off-the-shelf' filters to derive covariates for their geostatistical models is huge. In fact, to the best of our knowledge, the study we present here is the first of its kind to demonstrate an approach for automatically deriving optimal task-specific covariates from gridded datasets for geostatistical modelling applications. The covariates we derive are optimal in that they have been engineered by our deep neural network to have maximal explanatory power with respect to the target variable, to which they relate linearly. This linearity ensures that our deep learned covariates are compatible for use within the fixed effects component of the typical geostatistical model, $\mathbf{X}\boldsymbol{\tau}$ in Equation 4.1.

In reality, if all the covariate information provided to the geostatistical model is being processed through our neural network, as it is in this study, then the neural network's output *is* the fixed effect component of our geostatistical model. This is because of the 1:1 relationship (plus noise, $\boldsymbol{\varepsilon}$) between our neural network's output (the covariate, with values contained in $\mathbf{X}$) and the target variable, $\mathbf{Z}$. As a result, the value of $\boldsymbol{\tau}$, the fixed effect coefficient (singular in this case), would be one. We therefore replace our typical geostatistical model formulation with a 'deep covariate-learning' geostatistical model formulation:

$$\mathbf{Z} = \mathbf{D} + \mathbf{u} + \boldsymbol{\varepsilon}, \tag{4.2}$$

where $\mathbf{D}$ is the output of our deep neural network. Because of the additive nature of these formulations, we do not give much consideration to the spatial component, u, or the noise component, $\boldsymbol{\varepsilon}$, in the rest of this chapter. They do not interact with $\mathbf{D}$, and so we focus on evaluating the explanatory power that our deep neural network output, $\mathbf{D}$, provides to our target variable, $\mathbf{Z}$, on a stand-alone basis.

4.2 Method

Our approach is inspired by the work of computer scientist Geoffrey Hinton and colleagues, who revolutionised the field of computer vision in 2012 by using deep learning to achieve a new state-of-the-art in image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krizhevsky, Sutskever and Hinton, 2012). Prior to their work, image classification problems had been solved by providing linear classification algorithms with sets of manually derived image features. Similarly to the way that 'off-the-shelf' covariates are currently used in geostatistical modelling problems, it seemed unlikely that the manually derived image features were optimal for the task at hand, but a viable alternative had yet to be proven. Deep learning changed everything by replacing the existing setup with end-to-end learning: in deep learning the classification algorithm is also the feature learner — feed raw images in, and get answers out. In 2012, the answers that Krizhevsky, Sutskever, and Hinton got out — correct labels for images — were the best that had ever been achieved (Krizhevsky, Sutskever and Hinton, 2012), and lead to the ubiquitous use of deep learning in computer vision applications.

We hope that the parallels between manually creating features for image classification and manually deriving covariates for geostatistical applications are apparent. At the same time we do acknowledge the view of Coveney, Dougherty and Highfield (2016) that 'big data need big theory too' and that while automatically learning features is valuable if it leads to improved predictions, perhaps the greatest value of all is in understanding how the features that have been learned relate to the physical processes of nature - a topic we do not get into here, but which future work should explore.

In our case, we want to learn features from terrain texture in a similar way, so that we can go beyond using ‘off-the-shelf’ terrain derivatives as covariates, and extract more explanatory power from the landscape for any specific task. To do this we use our own deep neural network, constructed from similar building blocks as used by Hinton and colleagues in 2012. The critical difference is that in our case we want to learn to do image *regression* rather that classification, because our target variables (element concentrations from geochemical survey data) are continuous. In practice this simply means giving our network a single linear output rather than using a multinomial logistic output. An additional consideration for us has been the importance of retaining spatial context amongst the terrain texture. Translation invariance is an important feature of the deep neural networks used in image classification: it shouldn’t matter where in the image the cat is, it’s still an image of a cat. For our purposes however, it seems likely that the positions of terrain features relative to our prediction point and to each other matters greatly. In geochemistry, concentrations of immobile elements can be expected to be associated in situ with certain bedrock types, while mobile elements may show spatial relationships with distance to faults and other fluid conduits along which they might be mobilised.

It should be noted at this point that deep learning has been applied to problems within the realms of remote sensing and geostatistics before, and the novelty of our study does not lie in deep learning itself but in how we use it. For some background, deep learning has seen significant use in remote sensing applications in the latter half of the last decade, applied to tasks of object detection, scene classification, image fusion, image registration, land-use classification, semantic segmentation and more (e.g. Han et al., 2014; Zou et al., 2015; Zhang, Zhang and Du, 2016; Zhu et al., 2017; Ma et al., 2019). Meanwhile, general machine learning approaches have been applied to the spatial interpolation of environmental variables (e.g. Li et al., 2011), which had traditionally been considered the preserve of geostatistics. Complex mapping tasks, such as that of landslide susceptibility (Pourghasemi and Rahmati, 2018), or mineral prospectivity analysis (Rodriguez-Galiano et al.,

2015) have always relied on being provided with good covariates in order to achieve good results. It is exactly to these complex mapping tasks, where a ground-measured target variable is modelled with the support of remotely-sensed auxiliary variables, that our deep covariate-learning approach appeals. Where previously covariates have had to be derived manually from remotely-sensed auxiliary variable grids, deep covariate-learning allows this covariate-derivation process to happen automatically and optimally. The unique contribution of this study is therefore that, to the best of our knowledge, it is the first to show how the feature-learning ability of deep learning can be used within the framework of the well-established BLUP geostatistical model (Equation 4.1), thus providing new capabilities for use in geostatistical applications.

We believe that the interface between deep learning and geostatistics is an under-explored area in general, but would like to highlight some contributions that have preceded us in this space: Wadoux (2019) and Wadoux, Padarian and Minasny (2019), and Padarian, Minasny and McBratney (2019) have demonstrated how deep learning can be used in the context of digital soil mapping, and their work has utilised the feature-learning ability of deep learning (although manually derived covariate grids are also included). However, where as we present deep learning as a way to learn optimal covariates for use in typical geostatistical models — as deep covariate-learning — these previous studies have used deep learning to replace the entire geostatistical model. As we discuss later on, there may well be benefits to such end-to-end approaches. Nevertheless, we believe our deep covariate-learning approach provides a unique contribution to this under-explored research area in that it lays bare the ability of deep learning to extract information from remotely-sensed auxiliary variable grids even in the absence of explicit spatial location information. We hope this work will contribute to continued investigation into how to combine the best of both worlds (geostatistics and deep learning) in order to advance our capabilities in modelling and mapping complex environmental phenomena.

4.2.1 Data setup

For this study, we make use of two datasets: 1) NASA's SRTM 90m gridded global elevation data (Van Zyl, 2001), from which we use deep learning to derive optimal covariates in order to map 2) element concentrations from the British Geological Survey's G-BASE stream sediment sampling program (Johnson et al., 2005). Both can be seen in Figure 4.1. The geochemical dataset contains element concentrations from 110 794 sample sites from across the UK, though the number of observations used in this study varies by element as sites containing NA values are excluded. Any element concentrations reported below the accepted lower limit of detection were set to half the lower limit of detection as in previous studies using the geochemical dataset (Kirkwood et al., 2016b). For readers whose focus is on geochemical mapping and prospectivity analysis, we would recommend using log-ratio transformations on the geochemical data to avoid issues with compositional closure (Pawlowsky-Glahn, Egozcue and Tolosana-Delgado, 2015). However, in this study our focus lies in learning terrain textural features, and so we simply use our element concentrations in their raw form, with the exception that we log transform arsenic in order to improve stability of the gradient descent process by which the neural network is trained, and to make for more eye-friendly visualisations.

While our target variables are simple element concentrations, our inputs are square images of SRTM terrain data. For our training dataset, these images each consist of a 32x32 cell window of terrain centred around their respective geochemical sample site (Figure 4.2). We use a cell size of 500m, which gives a real-world window size of 16x16km square. The size and resolution of these images can be thought of as a tunable hyper-parameter to the neural network, but in reality we chose them by our own visual judgement, believing that they should provide a reasonable amount of information without exceeding our compute capacity (an Nvidia Titan X Pascal GPU - with thanks to Nvidia's grant scheme). As is standard practice in neural network training, we normalised our input data

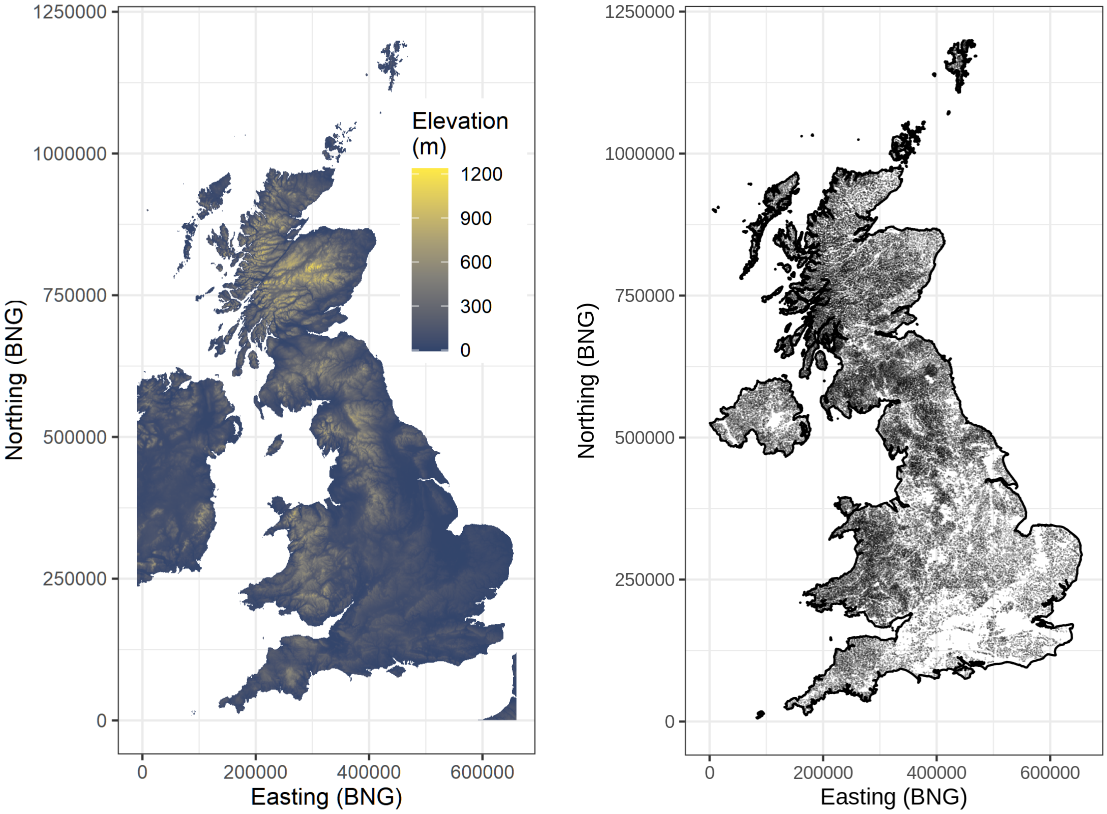


Figure 4.1: **Left -** SRTM 90m elevation data for the UK. **Right -** G-BASE stream sediment geochemistry sample collection sites (110 794 sites in total).

values. This is typically done to each input variable by subtracting the mean and dividing by the standard deviation, in order to achieve a mean of zero and a standard deviation of one. In our case we set the centre of each image to zero and divided all elevation values by the standard deviation of the UKs elevation grid. By setting the centre of each image to zero, we remove elevation as an explicit variable to the neural network. We also don't provide easting and northing to the neural network - terrain texture is all it has to make use of.

For each element, our dataset therefore consists of an element concentration vector of length n and a corresponding multidimensional array of dimensions $n \times 32 \times 32 \times 1$ that contains the images to be input to the neural network. It is worth mentioning that while our images only contain a single channel (terrain) there is no reason why our approach cannot be extended to multiple channels if other continuous covariates are available (such as from other airborne and satellite surveys).

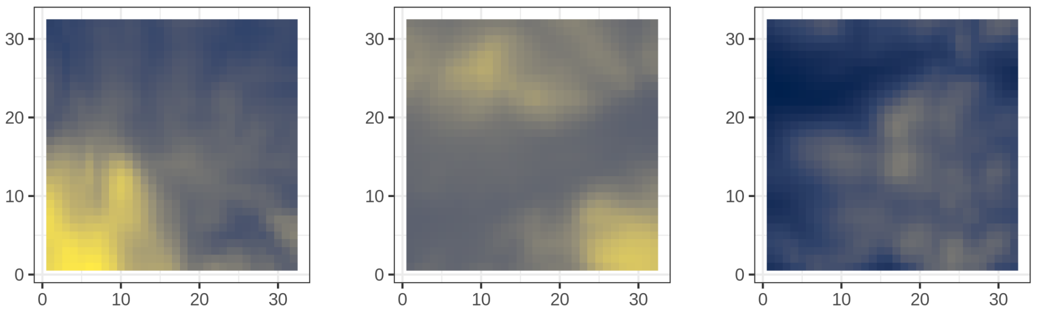


Figure 4.2: Three examples of 32x32 cell terrain input images as provided to our neural network. The colour scale is linear with cell elevation and is shared across all three images - more extreme shading variations therefore represent more extreme terrain. However, the absolute elevation of each terrain image has been normalised out, so that the central point is always at zero. This means that the neural network cannot use absolute height to ‘cheat’ - it must learn features purely from the terrain texture.

4.2.2 A deep neural network for terrain filtering

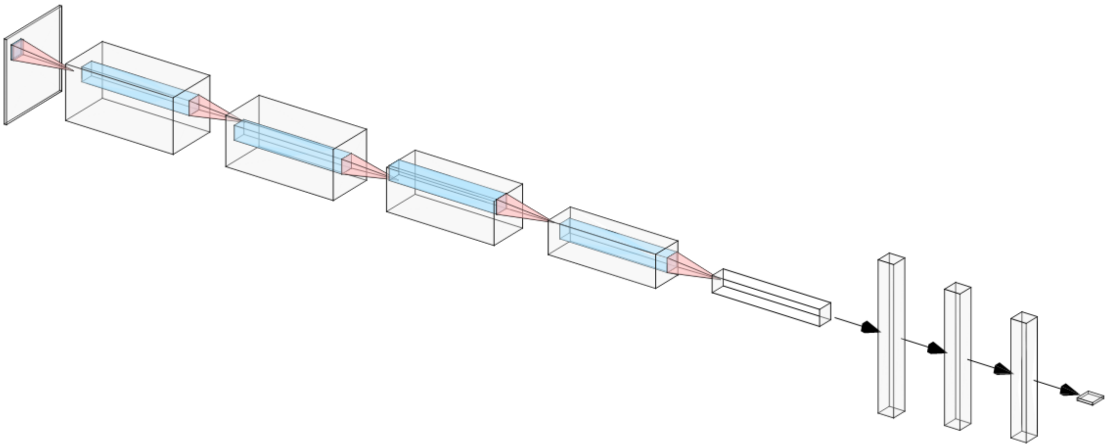


Figure 4.3: The architecture of our deep neural network. Input terrain images of size 32x32x1 (left hand side) are filtered through 5 128-channel convolutional layers and a single average pooling step to represent each image as a 4x4x128 spatial tensor. This then flattened into a vector of length 512 before being passed through two more fully connected hidden layers (256 and 128 nodes) prior to the final output - a single linear output (right hand side). The network uses dropout throughout, and a small amount of gaussian noise is added before each convolutional layer, to minimise overfit.

We implement our deep neural network using the Keras interface to Tensorflow, via the R language for statistical computing. We refer readers to our code for full details. The architecture of the network we present here represents the best performance we were able to achieve through fairly extensive trial and error. For future versions we may utilise automated procedures for architecture design and hyper-parameter tuning, but it was an enjoyable experience to gain intuition into effective neural network designs for extracting information from the terrain. The design we settled upon (Figure 4.3) consists of a series of stacked convolutional layers topped off with an average pooling layer which feeds into a fully-connected multilayer perceptron-type architecture which provides the final output (predictions of element concentrations). In total our network has just over 600 000 trainable parameters, and our objective function is to minimise mean-squared-error (MSE) in relation to the target variable. In order to prevent overfitting, we use dropout at every level in the network, and inject a small amount of gaussian noise ahead of each convolutional layer to further aid generalisation. Despite our efforts, it seems almost certain that the design we present here is *not* truly optimal (indeed, the optimal network design would be different for any given set of data) but it performs well in our application. The field of deep learning is very fast moving, and in this study we aim to share the general approach of using deep learning to derive task-specific covariates from terrain texture, rather than promote any particular network architecture.

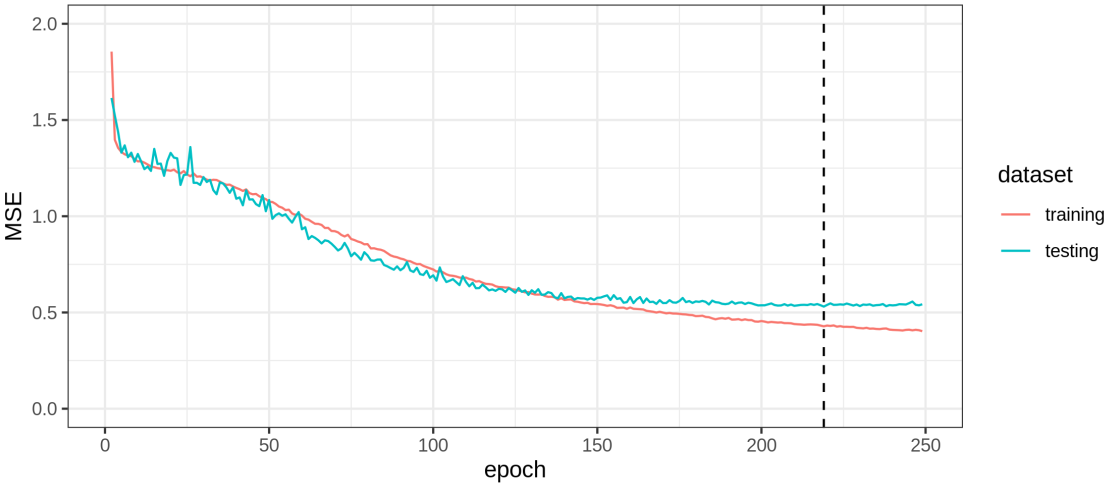


Figure 4.4: The training history of our neural network trained to predict log(arsenic) in stream sediments from terrain texture. The vertical dashed line marks the best epoch, for which the mean-square-error (MSE) on held out test data is lowest. The weights at this best epoch are the ones that are kept for subsequent use.

To train the neural network, for each element we split our dataset into 10 folds at random, and trained using 9 of them, while monitoring the mean-squared-error (MSE) of the neural network's predictions on the 10th fold to ensure that we did not overfit. We trained the neural network using the ADAM optimiser, and a batch size of 4096 observations. We ran training for up to 300 epochs (Figure 4.4), but early stopping tended to find the best fit around 200 epochs (before the MSE began to increase again on held out test data as the network began to overfit, but this was very gradual thanks to our regularisation measures). On our NVIDIA Titan X Pascal GPU each training run (one for each element) took about 10 minutes.

Once our deep neural network has been trained to predict the concentration of an element from the terrain (though this could equally be any other target variable), its output *is* the optimal terrain texture covariate that we wanted to learn. As we saw in Equation 4.2, if no other covariates are supplied to the geostatistical model, as is the case in this study, then the deep neural network output in fact becomes the entire fixed effect component of our ‘deep covariate-learning‘ statistical model. The neural network knows nothing of the spatial location at which a prediction is to be made (no easting, northing, or absolute elevation were provided) and can only extract information contained within the texture of surrounding terrain. It stands in for the role of the ‘off-the-shelf‘ covariates in the typical geostatistical model, with the aim to provide as much explanatory power as possible independently from the spatial component of the problem. The difference is that by using deep learning we are able to optimise this process of extracting information from the terrain — the neural network learns to derive the best terrain texture covariates that it can for the task at hand — providing as much explanatory power as possible with respect to the target variable.

4.3 Results

We can evaluate the explanatory power of our neural network's output by comparing its predictions to the true observed values in held-out test data (the 10th fold - not used in training, Figure 4.5). By doing so we find that our deep learning approach is able to explain a significant proportion of the variance in our target variables. It explains 61% of the variance in log(arsenic) (As), 58% of the variance in calcium (oxide, CaO), and 64% of the variance in potassium (oxide, K2O). These are fairly high degrees of explanatory power to achieve by harnessing the information contained within terrain texture alone.

For reference, previous studies in geochemical mapping have achieved out-of-sample R^2 values that top out at about 0.7 (Kirkwood et al., 2016a; Wilford, Caritat and Bui, 2016) although this of course varies between different elements, study areas, sampling strategies, and modelling approaches. In (Kirkwood et al., 2016a) ordinary kriging achieved cross-validated R^2 values above 0.6 for only 6 of 51 elements modelled. Wilford, Caritat and Bui (2016) achieved an R^2 value of 0.7 for chromium and 0.67 for sodium, both modelled using the 'Cubist' algorithm (similar to a random forest but with trees whose terminal nodes are linear functions) and a range of 'manually engineered' environmental features.

Geologists have long understood that underlying geology is reflected in the terrain, but the complexity of this relationship — requiring caveats, conditions, and qualifiers at every turn — has never lent itself to being formalised. It is somewhat remarkable then, that deep learning has been able to capture these relationships so successfully, and in less than ideal circumstances given that our neural network has never been told where it is - all it gets to see is the 16x16km window of terrain - and always centred at zero elevation.

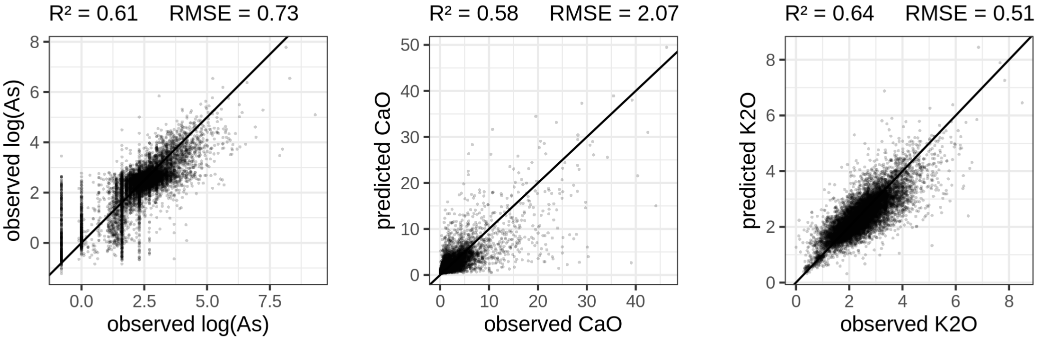


Figure 4.5: Plots evaluating the predictive performance, or explanatory power, of the output of our deep neural network trained to optimally extract terrain texture information in order to predict each of our three target variables: arsenic (As), calcium (CaO), and potassium (K2O) concentrations in UK stream sediments. These evaluations are made on held out test data that was not seen by the network during training.

To get a feel for the complexity of terrain features that the neural network has been able to learn (each in relation to the concentrations of chemical elements in stream sediments) we can generate maps of its output. We do this by making predictions from the neural network on a regular grid. For each prediction, the corresponding 16x16km terrain window is first extracted from the underlying SRTM elevation data (and elevation normalised, as explained in methods), which are then provided to the neural network so that it can make predictions for the new locations. Even though the neural network has never seen these new windows of terrain before, we take its performance on the held-out test set (Figure 4.5), which it had also never seen, as evidence of its explanatory ability on previously unseen data. The maps we produce in this manner are essentially SRTM elevation grids run through a complex non-linear filter (machine-learned from the bottom up, not designed from the top down) which maximises explanation of the target variable.

These deep-learned covariate maps (Figure 4.6, Figure 4.7, Figure 4.8) reveal a great deal of geological information, but in fact their task of explaining stream sediment geochemistry is more complex than explaining geology alone: Not only are stream sediments subject to the influence of surface processes as well as geological ones, but they also consist of mixtures of material accumulated from their upstream catchment area rather than representing any single point. This areal property makes their prediction difficult without accounting for upstream catchments during any modelling process (Kim et al., 2017). Visually (and it will be very interesting to investigate further), our deep-learned covariate maps do appear to have captured some flow-like effects. For example, in Figure 4.7 we can see patterns that appear to show the ‘washing out’ of elevated calcium concentrations from the chalk scarp that brightly trends north-east into East Anglia (the most eastern lobe of the UK). While these maps — which harness *only* the information contained within terrain texture — fall short of explaining all of the variance in our target variables (which would never be expected) it is somewhat surprising that they are able to explain so much of it (with R^2 values around 0.6 for all three elements), and a very encouraging result for the use of deep learning in

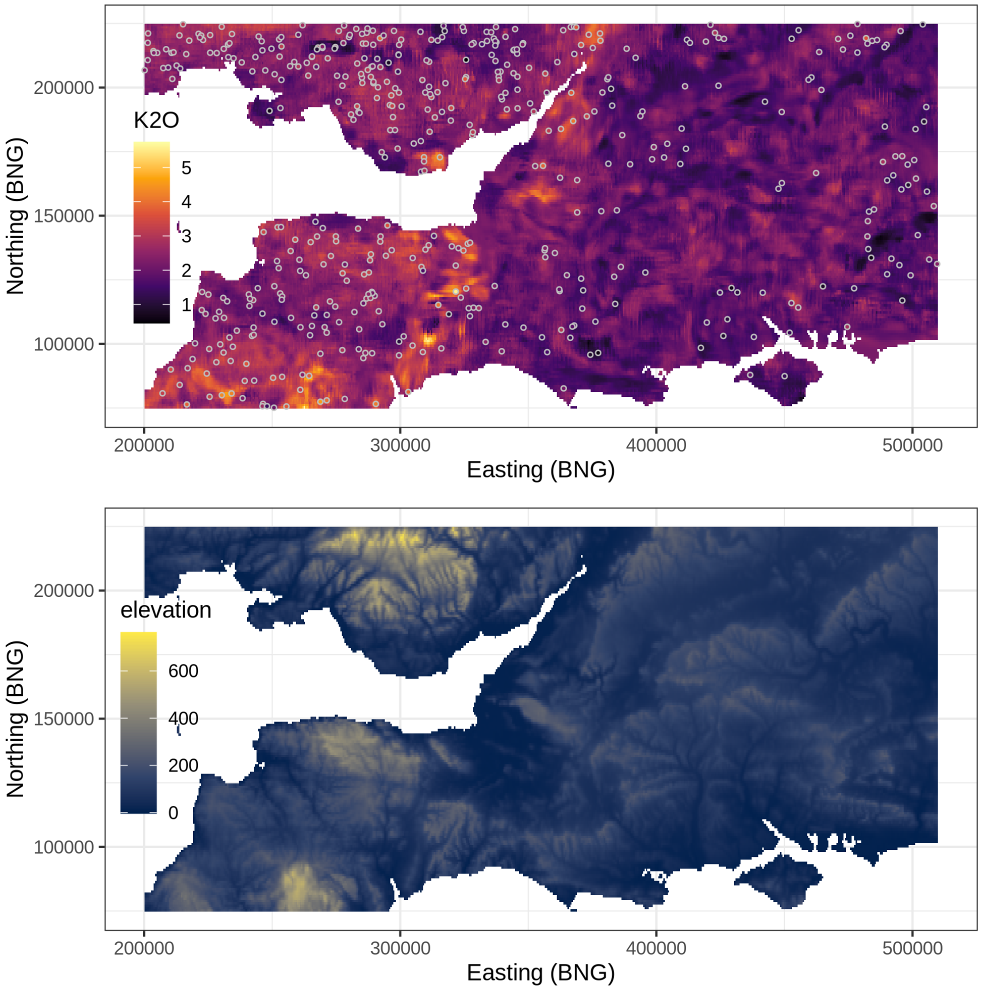


Figure 4.6: **Top -** Map of potassium concentrations as predicted exclusively from terrain texture using our deep neural network. 500 random geochemical sample sites are overlain. These ground-truth point values share the same colour scale as the raster map. The lack of deviation between the ground-truth and the prediction (also seen in Figure 4.5) supports the conclusion that the detail in the map is 'real' and not a product of over-fitting. However, at these scales some checker-board aliasing artefacts are apparent, which we would hope to remove with subsequent refinement of our neural network architecture. **Bottom -** The corresponding SRTM terrain from which the above map is derived via deep learning.

geostatistical applications.

4.4 Discussion

Our results have shown that deep neural networks are capable of extracting a great deal of geochemically-explanatory information from terrain texture alone, and it seems likely that similar success could be had by applying our methodology to other target variables and input grids. Adding additional channels to our terrain

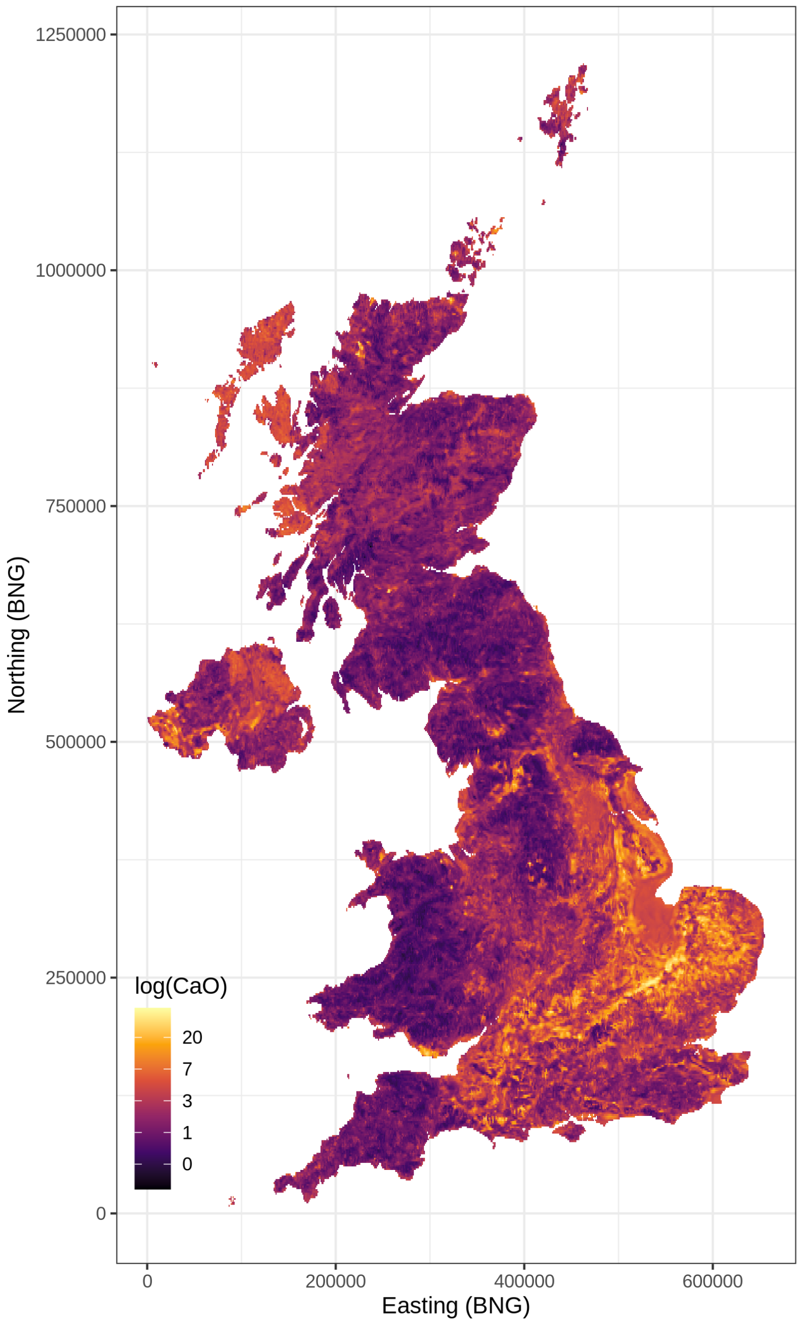


Figure 4.7: Our optimal terrain texture covariate for the prediction of stream sediment calcium. The map was produced by running the UK's SRTM elevation grid through our deep-learned terrain texture filter, optimised for explanatory power with respect to calcium concentrations. This map accounts for 58% of the variance in calcium concentrations through terrain texture alone. Subsequent geostatistical modelling can be used to improve prediction further, by taking account of spatial information (easting, northing, elevation) and perhaps other non-terrain based covariates too.

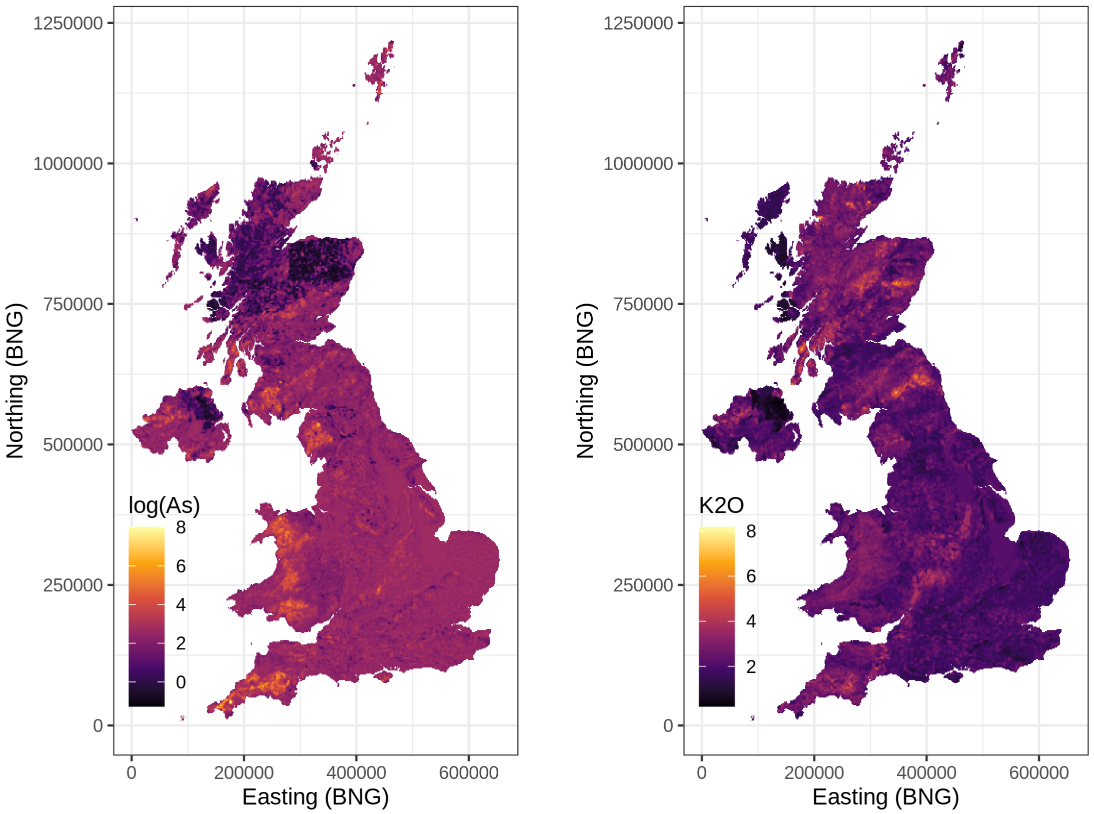


Figure 4.8: **Left -** Our optimal terrain texture covariate for the prediction of stream sediment arsenic concentrations. **Right -** Our optimal terrain texture covariate for the prediction of stream sediment potassium concentrations. In both cases, geological features are clearly apparent, and it is fascinating to see these being revealed through geochemically-optimal filtering of terrain texture alone. We recommend the British Geological Survey's iGeology mobile app (`https://www.bgs.ac.uk/igeology/`) to readers who wish to learn more about the features that these maps reveal.

input images where available, such as for gravity, magnetics, and radiometrics data would likely further improve the neural network's ability to explain geochemistry and perhaps other target variables too. It will be very interesting to explore how widely applicable this deep covariate-learning approach is in future research. Could deep learning revolutionise geostatistics the same way it revolutionised computer vision in 2012? Some encouraging evidence comes from the fact that previous investigations of machine learning for geochemical mapping (but using 'off-the-shelf' covariates; Kirkwood, 2016; Rodriguez-Galiano et al., 2015; Kirkwood et al., 2016a; Zuo, 2017; Kirkwood et al., 2017) have generally found terrain data to be among the least informative when compared to data from geophysical surveys (Kirkwood et al., 2016a). It will be interesting to see what deep covariate-learning can achieve when applied to these innately more geochemically-informative datasets.

In a sense we do injustice to deep learning in this study by not treating our application (modelling element concentrations) as an end-to-end problem: we have only tasked the neural network with the restricted function of learning to derive optimal terrain texture covariates for use in geostatistical models (i.e. either to contribute to $\mathbf{X}$ in Equation 4.1, or to take the place of $\mathbf{X}$ as $\mathbf{D}$ in Equation 4.2), rather than tasking the neural network with replacing the geostatistical model entirely. To do so would require that the neural network also handles the spatial component of the problem. This could be achieved most conveniently by simply providing the neural network with easting, northing, and absolute elevation as additional input variables. The neural network would effectively then replace both $X\boldsymbol{\tau}$ and u in our model formulation. It is actually likely that doing so would result in improved prediction accuracy over the geostatistical model by virtue of the fact that the neural network would be free to learn the interactions between terrain features and spatial location. Conversely, the additive nature of the statistical model formulation (Equation 4.1, Equation 4.2) prevents interaction between the spatial component, u, and the fixed effects 'regression-on-covariates' component, $\mathbf{X}\boldsymbol{\tau}$. This is perhaps a detrimental over-simplification, particularly for large and heterogeneous study areas, although it is well established practice nevertheless.

The reasons we have not gone all the way to providing an end-to-end 'complete solution' neural network in this study are two-fold. Firstly, at this stage we find more scientific interest in investigating the ability of deep-learning to derive optimal covariates for geostatistical modelling, given that the use of covariates in geostatistical models (i.e. Equation 4.1) is such well-established practice. As it is, our 'deep covariate-learning' geostatistical model formulation (Equation 4.2) seems like a reasonable middle ground from which to investigate the opportunities of deep learning within geostatistics without having to leave the established geostatistical modelling framework behind. This brings us to the second reason for not presenting an end-to-end solution here: While an end-to-end approach would play into the defining strength of deep learning — its unique ability to learn features from unstructured data in order to optimise an objective — it would also reveal what is currently deep learning's main weakness: uncertainty quantification. Uncertainty quantification in deep learning is a rapidly developing sub-field and promising breakthroughs have been made (e.g. Gal and Ghahramani, 2016; Kendall and Gal, 2017; Farquhar, Osborne and Gal, 2019), but at the time of writing, it is likely that potential end users of our approach would prefer to use deep learning to derive optimal covariates for use within well-established geostatistical model formulations (e.g Equation 4.1), hence the title and angle of this chapter (although, as you will see in Chapter 5 and beyond, we do further develop this approach into a full probabilistic end-to-end deep learning model, with perhaps surprisingly good result).

Despite the restricted capacity within which we apply deep learning in this study (i.e. to learn optimal covariates for geostatistical modelling, rather than using deep learning as an end-to-end solution in itself), the implications of our results are very significant. Let's take mineral exploration for example, although similar situations are likely to occur in other applications: The original geostatistical approaches (still often used), which rely purely on the spatial auto-correlation of the target variable are almost destined to perform poorly in the search for new mineral deposits. This is because they can only interpolate between observations in the geographic space. In such cases, if we have not been fortunate enough to 'hit' a mineral deposit with one of our samples, then the deposit can easily remain unseen between sampling locations.

Adding a 'regression-on-covariates' fixed effects component to the geostatistical model (e.g. Equation 4.1) alleviates this pathology to an extent, but only in as much as the available covariates can provide a good explanation of the target variable. Using deep learning to derive optimal covariates is therefore a step-change in the geostatistical modelling approach, as it allows us to objectively optimise the explanatory power we can obtain from gridded auxiliary datasets for any geostatistical modelling task. In doing so, we are able to explain the distribution of our target variable in terms of the deeper relationships between the target variable and terrain properties. Although not done as part of this study (or within this thesis), it would be interesting to provide the neural network with a randomly generated spurious covariate in addition to the genuine covariates (in this case just terrain elevation). By assessing to what extent the spurious covariate may influence the neural networks predictions (and latent representations at every layer of the neural network) could provide a means of checking how susceptible the neural network is to learning nonsense features. The hope would be that the neural network would not learn nonsense features, as this would indicate over fitting to spurious relationships, at the expense of ability to generalise in the real world).

The results we have obtained in this study demonstrate that the relationships learned by the deep neural network do generalise spatially. This is evidenced by the fact that our neural network achieves the explanatory power that it does without ever being provided with easting, northing, and absolute elevation by which to infer its spatial position. The patterns that it learns to recognise between terrain texture and geochemistry therefore have to be applicable throughout the study area. This feature of deep covariate-learning therefore makes it an exciting new tool for identifying undiscovered mineral deposits, assuming that some examples of known mineral deposits are included within the training data. Based on our results, we would not be surprised to see deep learning become a key technology for discovering the mineral deposits of the future, each always harder to find than the last.

4.4.1 A look to the future

Interestingly, Gaussian process regression — essentially the same method as kriging under a different name — is today considered a leading machine learning technique for applications where uncertainty quantification is important, and is often applied to higher dimensional problems than the mostly-spatial ones encountered in the field of geostatistics. Gaussian process regression is favoured over other methods for uncertainty quantification due to its well understood mathematical properties and its compatibility with the Bayesian framework (Gibbs, 1998) which has also been adopted within geostatistics (e.g. Diggle, Tawn and Moyeed, 1998; Sahu, 2022). However, in 1995 Radford Neal showed that as the number of hidden nodes in a single layer fully-connected neural network approaches infinity, the network will become mathematically equivalent to a Gaussian process (Neal, 1995). More recently, similar equivalence has been explored between deep fully-connected neural networks and Gaussian processes (Lee et al., 2017), and we now have Deep Gaussian processes (Damianou and Lawrence, 2013), including with convolutional layers (Blomqvist, Kaski and Heinonen, 2019). So what's the catch? Computational complexity. In terms of time, neural network training scales linearly with the number of observations, however Gaussian process inference scales with the cube of the number of observations (Liu et al., 2020). This is perhaps the main reason why deep neural networks have risen into mainstream applications ahead of Gaussian process regression - they are allowing us to solve otherwise unsolvable big data problems, and in many applications deterministic prediction is adequate. However, as the neural network community strives to improve their ability to quantify uncertainty, and the Gaussian process community strives to reduce their computational footprint, the two camps may well converge on methods that provide very similar functionality to practitioners.

Where will this leave geostatistics? It seems important to frame the functionality of well established geostatistical models (Equation 4.1) in the context of the functionality that deep neural networks (and deep Gaussian processes) can bring to the table. As demonstrated in this chapter, the capability to optimally extract information from unstructured data (like terrain grids) is extremely powerful, and could be a game changer in terms of eliminating the need to manually design (sub-optimal) covariates for use in geostatistical analyses. Essentially, we can push our variable selection processes right back to whatever raw unstructured data we have available, and trust (with empirical evaluation) deep neural networks to extract the relevant information. It is worth here a comment to highlight (and caution against) the possibility of data leakage, whereby a model may appear to perform well, but in fact has learned to ‘cheat’ its way to a correct answer using information that has been provided to it, but on which the answer does not causally relate. For example, in the original version of Rajpurkar et al. (2017)’s study on using deep learning to detect pneumonia in chest X-ray images, images of the same patients were used in the training dataset as in the test dataset, which meant that the deep neural network was able to ‘cheat’ and simply learn to identify the patients themselves, rather than any disease they may have been exhibiting. The analogy in the context of deep learning for spatial interpolation may be that if a neural network with a sufficiently wide convolutional perceptual field is used, then it may start to predict the right value simply by memorising the values at distinctive locations in the surroundings. The topic of data leakage in deep learning for environmental modelling remains to be explored in more detail.

There is an argument to say that learning features automatically reduces the interpretability of our model (deep learning as a ‘black box’), which we may have wished to preserve. It is true that there is no way to fully comprehend in ‘explainable’ terms the series of transformations that our neural network applies to terrain texture in order to produce a representation that correlates maximally with the target variable. On the other hand, if deep learning allows us to explain a higher proportion of variance without deferring to spatial auto-correlation, which is itself fairly opaque, then that could be seen as beneficial. With models increasingly being used to support important decision making, it could be argued that models should be judged by the quality of information they provide, rather than by how easily interpretable they are, in order that we can progress towards optimal decision making. In the end, the best approach to choose will be the one that best satisfies the objectives at hand, and this will always be case dependent.

4.5 Conclusions

In this chapter we have demonstrated a new approach for utilising deep learning to derive optimal terrain texture covariates for geostatistical modelling applications. We have shown that our deep learning approach is entirely compatible with the typical geostatistical model formulation (Equation 4.1) and in fact can be used as the exclusive source of covariate information in a 'deep covariate-learning' geostatistical model formulation (Equation 4.2). The results our deep neural network achieves on held-out test data are extremely encouraging. Terrain data has historically not tended to be regarded as particularly informative for most geochemical applications, at least within quantitative modelling, and yet our deep neural network has been able to extract sufficient information to explain 61%, 58% and 64% of the variance of our target variables: log(arsenic), calcium. and potassium concentrations in stream sediments. This is all from using *only* terrain texture, without accounting for spatial variability explicitly (the network was not provided with easting, northing, and absolute elevation as inputs, and had only 16x16km square images of terrain texture to work with). Within the geostatistical modelling framework, this spatial variability is accounted for instead by the spatial random variable component of the model (u in Equation 4.1).

Our results suggest that deep learning has a very significant role to play in the future of geostatistical modelling, and offers a step-change in how we can make use of gridded auxiliary datasets in the modelling process by allowing us to optimise the extraction of information from them. The covariates that our deep learning approach learns to derive are spatially generalisable within the study area, and it is quite possible that they can shed new predictive light on otherwise under-sampled geographic regions, for example for mineral exploration purposes. The strong predictive performance achieved using only 16x16km windows of terrain texture warrants further investigation of our deep covariate-learning approach using different window sizes, as well as including channels for additional auxiliary variables. The apparent ability of deep learning to capture complex structural relationships (for example, appearing to realise that stream sediments do ‘flow’ from upstream catchments) mean it will be interesting to see how much further this approach can be developed. At the same time, it is worth considering that improvements to prediction do not necessarily constitute improvements to understanding, and that improved prediction in the absence of improved understanding may not be as valuable scientifically as prediction and understanding improved in tandem. While deep learning may excel at prediction, attempts to improve understanding as a result have had mixed success (e.g., Kindermans et al., 2019; Zhang et al., 2021; Saxe, Nelli and Summerfield, 2021; Lei et al., 2020; Guo et al., 2016) although this is an area at the forefront of ongoing research (more generally under the guise of ‘XAI’ or eXplainable AI; Gunning et al., 2019; Das and Rad, 2020; Arrieta et al., 2020; Adadi and Berrada, 2018).

If enough explanation can be obtained from gridded datasets alone, then perhaps we will no longer have need for the spatial random variable component of our geostatistical models (u in Equation 4.1), and the fairly opaque spatial autocorrelation based explanation that it provides. Alternatively, we may find that the best overall predictive performance is achieved by using deep learning end-to-end for geostatistical modelling tasks, in which it would have the benefit over the typical geostatistical model (Equation 4.1) of being able to learn interactions between covariates features (which are themselves learned) and spatial location. Reliable estimates of uncertainty are perhaps the main justification for refraining from an end-to-end deep learning approach at the moment, hence in this chapter we demonstrate deep learning within the typical geostatistical modelling framework, but research into improving uncertainty quantification in deep learning is developing at a rapid pace. If the future of transport will be dominated by autonomous vehicles, the future of geostatistical modelling will surely also be driven by deep learning.

4.6 Code and data

We also thank the British Geological Survey for making the G-BASE geochemical data available for this study. For academic research purposes, readers may request access to the G-BASE dataset from the British Geological Survey at `https://www.bgs.ac.uk/enquiries/home.html` or by email to enquiries@bgs.ac.uk.

Chapter 5

Bayesian Deep Learning for Spatial Interpolation in the Presence of Auxiliary Information

Earth scientists increasingly deal with 'big data'. For spatial interpolation tasks, variants of kriging have long been regarded as the established geostatistical methods. However, kriging and its variants (such as regression kriging, in which auxiliary variables or derivatives of these are included as covariates) are relatively restrictive models and lack capabilities provided by deep neural networks. Principal among these is feature learning: the ability to learn filters to recognise task-relevant patterns in gridded data such as images. Here we demonstrate the power of feature learning in a geostatistical context, by showing how deep neural networks can automatically learn the complex high-order patterns by which point-sampled target variables relate to gridded auxiliary variables (such as those provided by remote sensing) and in doing so produce detailed maps. In order to cater for the needs of decision makers who require well-calibrated probabilities, we also demonstrate how both aleatoric and epistemic uncertainty can be quantified in our deep learning approach via a Bayesian approximation known as Monte Carlo dropout. In our example, we produce a national-scale probabilistic geochemical map from point-sampled observations, with auxiliary data provided by a terrain elevation grid. By combining location information with automatically-learned terrain derivatives, our deep learning approach achieves an excellent coefficient of determination (R^2 = 0.74) and near-perfect probabilistic calibration on held out test data. Our results indicate the suitability of Bayesian deep learning and its feature learning capabilities for large-scale geostatistical applications where uncertainty matters.

5.1 Introduction

Maps are important for our understanding of Earth and its processes, but it is generally the case that we are unable to directly observe the variables we are interested in at every point in space. For this reasons we must use models to fill in the gaps. In order to support decision making under uncertainty, statistical models are desirable (Berger, 1985). Kriging — the original geostatistical model — provides smooth interpolations between point observations based on the spatial autocorrelation of a target variable (Cressie, 1990; Stein, 1999). However, additional sources of information are often available thanks in part to the rise of remote sensing (Mulder et al., 2011; Colomina and Molina, 2014) which provides grids of what we consider here to be auxiliary variables (e.g., terrain elevation, spectral imagery, subsurface geophysics). These are complete maps of variables that we are not directly interested in but which are likely to contain information relating to our variables of interest.

How best to extract information from auxiliary variable grids for geostatistical modelling tasks has remained an open question, but has often involved trial-and-error experimentation using manually designed filters to extract features with as much explanatory power as possible (e.g., Ruiz-Arias et al., 2011; Poggio, Gimona and Brewer, 2013; Parmentier et al., 2014; Shamsipour et al., 2014; Kirkwood et al., 2016a; Kirkwood, 2016; Young et al., 2018; Lamichhane, Kumar and Wilson, 2019). For example, Youssef et al., 2016 use slope angle derived from a digital terrain model as a feature to explain landslide susceptibility, but many more complex features may be useful, and these are not necessarily known in advance. To enable the utilisation of complex and unknown features, here we present an end-to-end geostatistical modelling framework using Bayesian deep learning, which frames the information extraction problem as an optimisation problem (Shwartz-Ziv and Tishby, 2017), and in doing so eliminates the need for manual feature engineering and feature selection steps. Our approach therefore has the potential to supersede traditional geostatistical approaches by bringing automatic feature learning to probabilistic geospatial modelling tasks.

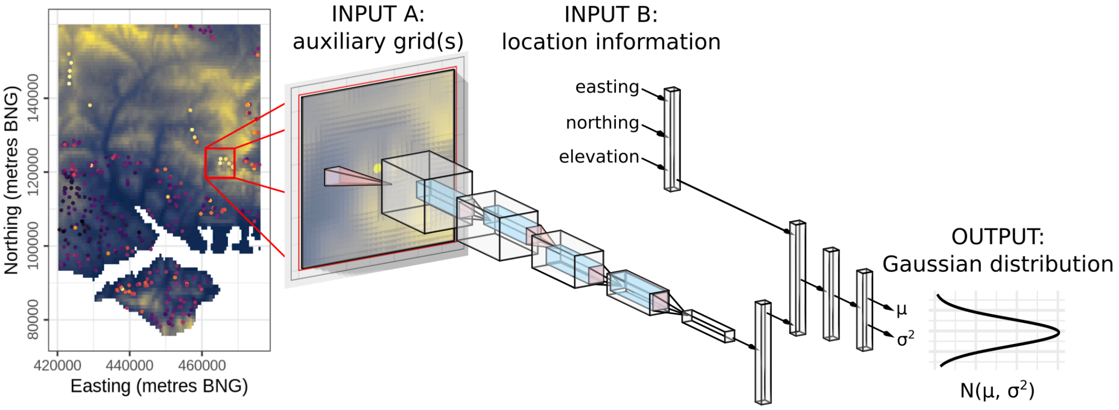


Figure 5.1: Overview of our deep neural network architecture visualised with the help of NN-SVG software (LeNail, 2019). For each observation, input A feeds an image of surrounding terrain into a stack of convolutional layers (shown as horizontal blocks). Simultaneously, input B feeds the observation's location variables into a fully connected layer. These two branches of the network are then concatenated and fed through a further two fully connected layers (shown as vertical blocks) from which the two parameters of a Gaussian distribution are output

Here we present a two-branch deep neural network architecture — convolutional layers for feature learning combined with fully-connected layers for smooth interpolation — that brings the benefits of deep learning to geostatistical applications, and we do so without sacrificing uncertainty estimation: Our approach estimates both aleatoric and epistemic uncertainties (via Monta Carlo dropout; Gal and Ghahramani, 2016) in order to provide a theoretically grounded predictive distribution as output, which is composed of spatially coherent realisations (see Appendix A: Simulation). Our work brings together ideas from the fields of machine learning (Krizhevsky, Sutskever and Hinton, 2012; Srivastava et al., 2014), remote sensing (Zhang, Zhang and Du, 2016; Zhu et al., 2017) and Bayesian geostatistics (Handcock and Stein, 1993; Pilz and Spöck, 2008), and unites them in a general framework for solving 'big data' geostatistical modelling tasks in which gridded auxiliary variables are available to support the interpolation of point-sampled target variables. We demonstrate our approach on a national-scale geochemical mapping task with encouraging results, both in terms of deterministic and probabilistic performance on held out test data. As far as we are aware, our framework is the first to provide both well-calibrated probabilistic output and automated feature learning in the context of spatial interpolation tasks. By using neural networks, we also ensure that our framework is scalable to the largest of problems.

While the framework we present here is new, it can also be seen as a unification and generalisation of a range of prior works. Deep learning (LeCun, Bengio and Hinton, 2015) — machine learning using 'deep' neural networks consisting of multiple stacked layers capable of learning hierarchical composite functions — has seen increasing uptake within scientific communities in the last decade. Deep neural networks typically consist of two components: a (deep) sequence of convolutional layers designed to extract a hierarchically more efficient encoding of the signal, followed by fully connected layers at the end to estimate the desired function from the encoded representation. Both parts are trained jointly using stochastic gradient descent, ensuring that the learnt features are optimised for the task. The popularity of deep learning has followed from breakthrough work by Krizhevsky, Sutskever and Hinton (2012) who achieved a new state of the art in image classification by using deep neural networks to automatically learn informative features from images (rather than manually engineering them). Deep learning has since been widely adopted within the remote sensing community (e.g., Zhang, Zhang and Du, 2016; Zhu et al., 2017; Li et al., 2017; Zuo et al., 2019) and has been applied to a variety of problems in geoscience, for example detecting and locating earthquakes (Perol, Gharbi and Denolle, 2018), detecting faults in 3D seismic data (Wu et al., 2019), and classifying lithologies from drill core images (Alzubaidi et al., 2021). However, difficulty in obtaining reliable uncertainty estimates from deep neural networks (Kendall and Gal, 2017) has meant that deep learning has not been widely adopted for applications where uncertainty matters (or, as in the aforementioned works, the proposed approaches skirt around the ever-present issue of uncertainty and simply use fixed-weight deterministic neural networks instead).

A few authors have made use of deep learning to automate feature learning in a geostatistical context (Padarian, Minasny and McBratney, 2019; Wadoux, Padarian and Minasny, 2019; Wadoux, 2019; Kirkwood, 2020), mostly for digital soil mapping, but only one — Wadoux (2019) — has been able to provide uncertainty estimates, though these were achieved via a bootstrapping approach and found to be underdispersive. Here we make use of a theoretically grounded and practically effective approach to uncertainty estimation in deep neural networks: Monte Carlo dropout as a Bayesian approximation, as conceived by Gal and Ghahramani (2016). The authors are aware of one prior instance of its use in a geospatial setting: for a semantic segmentation task by Kampffmeyer, Salberg and Jenssen (2016). While our work shares similarities with the work of Kampffmeyer, Salberg and Jenssen (2016), our motivation is from the angle of geostatistical tasks in which the challenge is to utilise auxiliary information to interpolate between sparsely-sampled point observations, whereas Kampffmeyer, Salberg and Jenssen (2016) tackle the remote sensing challenge of semantic segmentation: classifying each pixel of airborne images of the urban environment with their corresponding object class (e.g., car, building, tree) by training on fully manually labelled images without gaps. Both approaches require learning features from gridded data, but only ours combines this with general spatial interpolation abilities in order to provide a viable solution to the task of spatial interpolation in the presence of auxiliary information. Overall, while various separate concepts behind our work may be familiar to some readers already, here we bring them together and present Bayesian deep learning as a general solution for big data geostatistics.

Outside of the relatively recent influences of deep learning, there have also been longstanding works in the geoscience community to utilise Bayesian inference within general geological modelling practices. This initially stemmed from developments in geophysics (Tarantola, Valette et al., 1982; Mosegaard and Tarantola, 1995; Sambridge and Mosegaard, 2002) in which Monte Carlo methods were presented as a means to deal with the uncertainty that is inherent to under-determined inverse problems (where many different solutions are capable of generating the observed data, i.e. the solution is non-unique — a common occurence within the geosciences). More recently, Varga and Wellmann (2016) focused on how the Bayesian framework can be used to combine both geological knowledge and quantitative data into geological models, a theme that is further developed by Wellmann et al., 2018; Grose et al., 2019; Schaaf et al., 2020 and Olierook et al., 2021.

Our Bayesian deep learning approach for spatial interpolation in the presence of auxiliary information borrows more heavily from the machine learning and geostatistics literature than from these geological modelling works, but we acknowledge the background on both sides because we share the same motivations: the desire to incorporate all sources of information into our models, and also to characterise both aleatoric and epistemic uncertainties, such that our models will be maximally informative while remaining honest about uncertainty. Our Bayesian deep learning approach could be seen as the 'data rich, prior knowledge poor' end-member on a spectrum of Bayesian modelling methods, with the above mentioned geological modelling approaches falling closer to the 'knowledge rich, data poor' end of the spectrum. The data rich setting brings its own set of challenges in terms of scalability, and the need to deal with large volumes of data is a strong justification for adopting a neural network based approach such as the one we present here.

5.2 Method

5.2.1 Feature Learning for Geostatistics

The core domain of geostatistics has been in the spatial interpolation of point observations in order to produce continuous maps in two or three dimensions. Kriging, the now ubiquitous geostatistical technique conceived by South African mining engineer Danie Krige (1951), originally accounted for only the location and spatial autocorrelation of observations in order to produce smooth interpolations (or threshold-classified smooth interpolations in the case of indicator kriging) that can be considered optimal if no other information is available (Matheron, 1962; Cressie, 1990), often under assumptions of stationarity and isotropy. When other information is available, as is commonly the case today, the pursuit of optimal spatial interpolation becomes more complex. An extension of ordinary kriging, regression kriging (which is also mathematically equivalent to universal kriging and kriging with external drift;Hengl, Heuvelink and Rossiter 2007 - see also Diggle, Tawn and Moyeed, 1998), allows covariates to be included in the model: the mean of the interpolated output is able to vary as a linear function of the value of covariates at the corresponding location (Gotway and Hartford, 1996). For an illustrative example, the inclusion of elevation as a covariate in an interpolation of surface air temperature data could be expected to result in a map that reflects the underlying elevation map, i.e. whose mean function is a linear function of elevation. However, this quickly brings us to the limits of regression kriging: what if a linear function of elevation does not provide as much explanatory power to surface air temperature as some non-linear function of elevation? What non-linear function of elevation would be the optimal one? At the same time, what if we also have wind direction available to use as a covariate? Wouldn't the best predictor of surface air temperature account for not just elevation, or wind direction, but how they interact, with air flowing down from mountains expected to be cooler? We quickly find ourselves in the realms of feature engineering and feature selection, a world of hypothesising and trial-and-error experimentation which has become a necessary but impractical step in the traditional geostatistical modelling process.

The defining strength of deep neural networks is their ability to learn features for themselves owing to their hierarchical structure in which the output of each layer (with non-linear activation function applied) provides the input to the next. Through back-propagation of error gradients, neural networks can automatically learn non-linear transformations of input variables and their interactions as necessary in order to minimise a loss function. It has also been shown that in the limit of infinite width (infinite number of nodes) a neural network layer becomes mathematically equivalent to a Gaussian process (Neal, 1996), which is itself the same smooth interpolator conceived by Danie Krige (i.e. kriging) under a different name. The deep neural network approach we present here combines these spatial abilities with a unique ability to learn its own features from auxiliary variable grids. We achieve this efficiently through the use of convolutional layers: trainable filters which pass over gridded data to derive new features, in a similar manner to how edge detection filters derive edges from photographs (Chen et al., 2017). By stacking convolutional layers, the complexity and scale of features that can be derived increases, along with the size of the receptive field of the neural network (Luo et al., 2016), which allows longer-range dependence structures to be learned.

5.2.2 Neural Network Architecture

As is shown in Figure 5.1, our neural network, which we constructed and trained using Tensorflow (Abadi et al., 2016) and Tensorflow Probability (Dillon et al., 2017), has two separate input branches: a five-layer convolutional branch that takes auxiliary variable images as input (the auxiliary information branch); and a single-layer fully connected branch that takes location variables as input (the geographic location branch). The outputs of these two branches are flattened and concatenated into a single 2048 dimensional vector (128 from the convolutional branch, 1920 from the fully connected branch) that feeds into the final layers of our neural network, which consist of a further two fully connected layers (1024 nodes, and 256 nodes) before outputting the two parameters — mean, μ, and variance, σ^2 — of a Gaussian distribution that represents the target variable. Throughout the network, we use the Rectified Linear Unit ('ReLU') activation function. In total our neural network architecture has 2.8 million trainable parameters.

The depth of the network, and the various widths of the layers, are to some extent arbitrary choices (at least we cannot show that this architecture is 'optimal') but through quite extensive manual tuning and experimentation this arrangement was found to work well while remaining relatively cheap to train (i.e. with test loss plateauing within about 30 mins of training on a single GPU, due to the relatively low number of trainable parameters, as far as deep neural networks are concerned). The 'optimal' architecture will vary from task to task, but given sufficient computational resources (and time) one could attempt to explore the space of possible architectures automatically using, for example, Bayesian optimisation (Kandasamy et al., 2018; White, Neiswanger and Savani, 2021) or evolutionary algorithms (Miller, Todd and Hegde, 1989; Leung et al., 2003; Idrissi et al., 2016).

In our auxiliary information branch, the first four layers are convolutional layers, each with 128 channels, using 3x3 kernels with a stride of 1 (apart from the first, which uses a stride of 3 in order to rapidly reduce the spatial dimensions of the feature learning branch from the 32x32 input image to 10x10 and therefore reduce the number of parameters in subsequent convolutional layers). In convolutional layers, filters, or kernels, of specified dimension (e.g. 3x3 pixels) are scanned across the input image, with a step size specified by the stride parameter (e.g. 1 pixel). For the first convolutional layer, the input image is the raw input image provided to the neural network - i.e. the observation centred image of terrain elevation, however for subsequent convolutional layers, their respective ‘input image’ is output from the previous convolutional layer. At each layer, the number of filters is specified as the number of channels, in our case 128. At each location across a layer’s input image, the output of a convolutional filter is the dot product of the filter’s values and the corresponding pixel values of the image, so that for each position of the kernel

$$\text{kernel} \cdot \text{img} = \sum_{i=1}^{n} (\text{kernel}_i \text{pixel}_i) = \text{kernel}_1 \text{pixel}_1 + \text{kernel}_2 \text{pixel}_2 + ... + \text{kernel}_n \text{pixel}_n. \tag{5.1}$$

This results in an output image with one pixel per kernel position, whose pixel values depend on the values of the kernel, which are trainable - thus giving the neural network the capacity to learn new features from images.

In our architecture, the fifth layer of the convolutional branch is a global average pooling layer, which reduces the final convolutional layer’s 4x4 (x128 channel) output image into a 1x1 (x128 channel) output by simply taking the mean of each channel’s image. Pooling introduces translation invariance into convolutional neural networks, because the exact position at which kernels are activated is lost in the average. While it may seem counter-intuitive to instil any level of translation invariance into a spatial mapping problem, we found that it seems to help reduce overfitting in the network, perhaps by encouraging it to learn features that are more descriptive of general setting (e.g. rock types) rather than of the exact location (e.g. where one specific stream meets another), and which are therefore more useful for generalising to unseen locations. We also found that average pooling outperformed max pooling for this use-case (resulting in lower validation loss) perhaps because it produces smooth transitions between the contextual features present at adjacent locations (rather than having features pop in and out depending on whether they are present anywhere within the extent of the auxiliary information image). While the convolutional architecture we propose here is effective, we also suspect that there is room for improvement, and would encourage further research in this area.

The geographical location branch (from input B in Figure 5.1) of our neural network consists of a single fully connected layer and is thus simpler than the convolutional branch. If the architecture included only the geographical location branch, then our model would simply be a deep fully-connected neural network operating on spatial location inputs (easting, northing, elevation), and could therefore be regarded as approximately performing 'deep kriging', or spatial interpolation using deep Gaussian process regression, which is what our neural network would become in the limit of infinite layer width (Neal, 1996). The idea of 'deep kriging' has been explored by Li, Sun and Reich (2020) in their paper of the same name, who propose a neural network architecture that uses an embedding layer to transform the input space. The inclusion of the auxiliary information branch in our neural network however turns it into something closer to 'deep regression kriging', a term that seems not to have previously been coined. Unlike traditional regression kriging, not only does our deep neural network architecture learn its own regression features automatically from gridded auxiliary information, but also learns interactions between these features and spatial location owing to the fact that we have subsequent hidden layers after the auxiliary and spatial information branches are concatenated. We can therefore think of our neural network as performing interpolation in a self-learned hybrid space - a representation which blends global location information with local contextual information.

So, our neural network provides a forward model from the location information and auxiliary information inputs, whose output is a Gaussian distribution representing the target variable (stream sediment calcium concentrations in the example we present here) via two parameters: the mean, μ, and variance, σ^2. These two output parameters are generated by separate linear functions of the same 256 learned features that constitute the neural network's final hidden layer (which in this case has 256 nodes). While μ is free to vary on the real line, we constrain σ^2 to always be positive (and therefore valid) by using a softplus link function: $softplus(x) = log(exp(x)+1)$. Because these output parameters have access to the same features, there may be some amount of cross-talk between them. This could be forcibly avoided by using separate neural networks to learn each of these output parameters. However, given the nature of the problem it is likely that both μ and σ^2 will vary according to the same underlying processes, such as changes in lithological or hydrological setting. It therefore seems a reasonable approach to allow both μ and σ^2 the capacity to share the same features (and to have them jointly inform the learning of these features). The quality of our results empirically supports this reasoning.

Our probability model $p(Y_s|x_s,w)$ for our target variable Y_s, at any location s, given the inputs x_s and weights of the neural network w is defined as

$$\begin{aligned} Y_s|x_s,w &\sim \mathcal{N}(\mu(x_s),\, \sigma(x_s)^2) \\ \mu(x_s) &= g(layer_{final}) \\ \sigma(x_s)^2 &= log(exp(h(layer_{final}))+1), \end{aligned} \tag{5.2}$$

where both g and h are linear functions of the 256 features of the final hidden layer of the neural network ($layer_{final}$) so that both take the form $g(layer_{final}) = \beta_0 + \beta_1 feature_1 + \beta_2 feature_2 + \beta_3 feature_3 + \cdots + \beta_{256} feature_{256}$. Each of these 256 features is itself a learned transformation of the neural network's inputs, x_s, namely location information and auxiliary information (Figure 5.1). The specifics of these functions are dictated by the trainable parameters, or weights, in the neural network. In the next section we explain how these parameters, and the functions they induce the neural network to represent, are learned.

5.2.3 Quantifying Uncertainties via MC Dropout

Traditionally, neural networks are deterministic models in that they provide a fixed output for a given input, subject to the values of their parameters (or weights, w). Neural networks are commonly trained through back-propagation to converge on a set of weights that minimise a loss function (often mean squared error in the case of regression problems). However, in these traditional deterministic neural networks the weights are fixed, having no distribution, which means that there is no way to estimate the uncertainty in these weights or therefore the uncertainty about the function or model that the neural network has learned. Natural processes inevitably involve uncertainties, and it is right that we should want to estimate these in order to provide well-calibrated probabilistic predictions suitable for use in decision support (Yoe, 2011; Fox and Ülkümen, 2011). We do so here using the Monte Carlo dropout approach of Gal and Ghahramani (2016) for approximate Bayesian inference, which allows us to capture both aleatoric and epistemic uncertainty as described by Kendall and Gal (2017). Aleatoric uncertainty can be thought of as the innate randomness in a data generating process — irreducible noise inherent in the observations of the target variable — and can be represented by using a parametric distribution as the output of the neural network so that, rather than making single point predictions, our model predicts a distribution whose variance acknowledges the inherent randomness in the observations. In our case, our deep neural network outputs the mean, $\mu(x_s)$, and variance, $\sigma(x_s)^2$, of a Gaussian distribution (Equation 5.2) which provides our likelihood function for the neural network ($\mathscr{L}_{NN}$). If we define vector $\boldsymbol{y} = (y_1, \ldots, y_n)$ to be the observed data on Y_s (our output variable). Then the likelihood is defined as the joint probability of the data $\boldsymbol{y}$ given specific values of $\mu(x_s)$, $\sigma(x_s)^2$ and w. This is given by

$$\begin{aligned}
\mathscr{L}_{NN} &= p(\boldsymbol{y}|\mu(x_s),\sigma(x_s)^2,w) \\
&= \prod_{s=1}^{n} p(y_s|\mu(x_s),\sigma(x_s)^2,w) \\
&= \prod_{s=1}^{n} \frac{1}{\sqrt{2\pi\sigma(x_s)^2}} \exp\left(-\frac{1}{2\sigma(x_s)^2}(y_s-\mu(x_s))^2\right) \\
&= \frac{1}{(2\pi\sigma(x_s)^2)^{n/2}} \exp\left(-\frac{1}{2\sigma(x_s)^2_s}\sum_{s=1}^{n}(y_s-\mu(x_s))^2\right).
\end{aligned} \tag{5.3}$$

If we were not also interested in epistemic uncertainty — uncertainty within the model itself — we could simply optimise the weights, w, to arrive at a fixed set which maximise the likelihood (maximise the probability of the data given the model). Assuming a Gaussian error distribution as we do here (with spatial covariance being handled implicitly within the neural network's parameter uncertainty, such that our neural network approximates the role of both the mean function and the Gaussian process in a typical ordinary kriging or regression kriging setup, with the Gaussian error distribution providing only non-correlated 'nugget variance'), maximising the likelihood would lead our neural network to fit the predictive mean, $\mu(x_s)$, equivalently to if we were minimising the mean squared error (MSE),

$$MSE = \sum_{s=1}^{n}(y_s-\mu(x_s))^2, \tag{5.4}$$

which is perhaps the most commonly used loss function for deterministic regression problems. However, by having our neural network learn both the mean, $\mu(x_s)$, and the variance, $\sigma(x_s)^2$, of our target variable as functions of the inputs, we can learn a spatially-precise heteroscedastic (Kendall and Gal, 2017) representation of the uncertainty within the data — the aleatoric uncertainty.

In addition to modelling the aleatoric uncertainty, we also wish to model the epistemic uncertainty — the uncertainty within the model itself. We do so in acknowledgement of the fact that, given that we do not have complete and perfect information (our set of observations is finite), there is uncertainty about the form of the true data generating process. This uncertainty can be reduced by collecting more data, but without infinite observations there will always be room for multiple possible model fits, or explanations, for the data we observe. If we simply task our neural network with learning a single ‘best fit’ function to represent the data generating process (e.g. by maximising the likelihood) then we would be ignoring this epistemic uncertainty about the range of possible fits, resulting in overconfident predictions and poor generalisation. Instead, to model the epistemic uncertainty we model a distribution over the range of possible model fits. To do so requires operating within the Bayesian framework to learn a distribution over the neural network weights, rather than simply learning a fixed set of weights as in non-Bayesian neural networks. However, learning a distribution over each weight, or parameter, in the model is a challenging proposition for large neural networks due to the extreme dimensionality of the model (2.8 million trainable parameters in our case).

Here we use the Monte Carlo dropout approach for approximate Bayesian inference in deep neural networks (Gal and Ghahramani, 2016). This approach places a Bernoulli prior row-wise over the weight matrices of the neural network, which means that for every iteration of training and prediction, the nodes of the neural network each have a probability of being switched off, or ‘dropped out’ (with weights set to zero). Each layer of the neural network can be represented by a matrix of weights which dictate the values of the layer’s nodes as weighted sums of the layer’s inputs, multiplied by an activation function (the ReLU function in our case, where

$$\text{ReLU}(x) = \max(0,x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{5.5}$$

The inputs to the first layer will be the raw inputs provided to the neural network, while subsequent layers will take the node values of the previous layer as inputs.

For example, if we imagine a neural network with four input variables, two hidden layers of eigth and six nodes, and a single linear output (e.g., Figure 5.2) then such a network will possess three weight matrices, W^1, W^2, and W^3, which parameterise the transformation from each layer to the next.

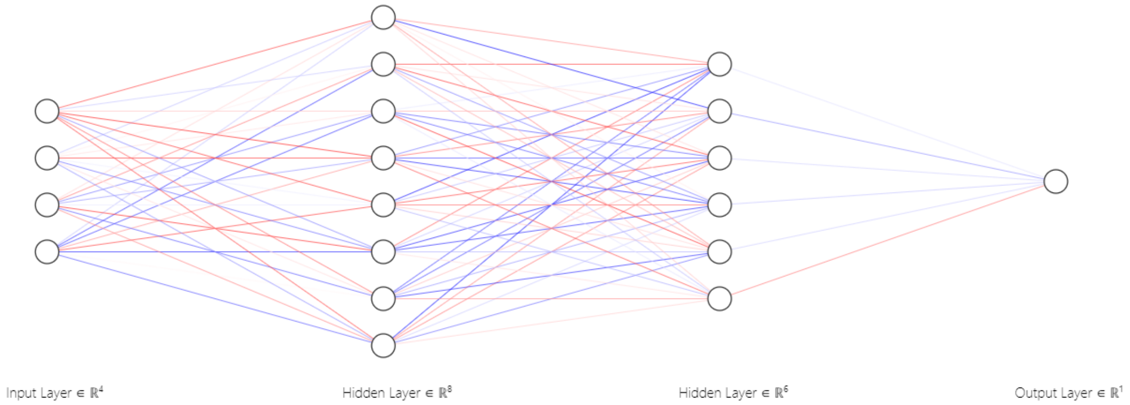


Figure 5.2: An illustration of a neural network with two hidden layers, using NN-SVG software (LeNail, 2019). Each connection represents a weight in the neural network, and is coloured from red to blue according to how positive or negative it is. The configuration of weights between each layer is represented mathematically by a weight matrix. The weights here have just been initialised randomly for illustrative purposes.

If we populate W^2, the matrix that parameterises the transformation from the first to the second hidden layer, with some random weights ($\in \mathbb{R}$)

$$W^2 = \begin{bmatrix} -1.08 & 0.71 & -0.19 & 0.14 & -0.43 & 1.2 \\ 1.22 & 0.85 & 0.82 & -0.04 & -1.5 & -0.31 \\ 0.68 & -0.77 & 0.57 & 1.33 & 0.11 & -2.17 \\ -0.29 & -0.08 & 1.46 & 0.89 & 1.33 & 1.19 \\ 1.12 & -0.1 & 0 & 1.25 & -0.01 & -1.36 \\ -0.15 & -1.16 & -0.45 & 0.27 & 0.28 & 0.03 \\ 0.49 & -1.5 & 1.67 & 1.53 & -0.24 & 0.01 \\ 0.47 & -0.66 & 0.38 & -2.21 & 0.98 & 2.07 \end{bmatrix} \tag{5.6}$$

where the value of each node in the second hidden layer (the columns) is a weighted sum of the values of the nodes of the first hidden layer (the rows), i.e. so that

$$L_1^3 = max(0, \sum_{n=1}^{n} (L_1^2 \times -1.08 + L_2^2 \times 1.22 + ... + L_8^2 \times 0.47) \tag{5.7}$$

where L_1^3 means the third layer's first node. Applying dropout row-wise, i.e multiplying each row by a bernoulli random variable $z \sim$ Bern(π) where $\pi = \Pr(z = 1)$, equates to giving each node of the second layer a probability $1 - \pi$ of being dropped out, in other words having its value set to zero and therefore not have the information it provides propogate through the subsequent layer. The inclusion of this stochastic dropout transforms the neural network from representing a single deterministic function to instead representing a distribution over functions, whose variance relates indirectly to the dropout rate π, as well as to the weights themselves.

The probability or rate at which nodes will drop out is a tuneable hyperparameter. While a Bernoulli prior may seem ‘unrealistic’ — why should a parameter only exist with a fixed probability? — the overall effect of Monte Carlo dropout on the network as a whole is to turn our single neural network into a near infinite self-contained ensemble. Each different configuration of dropped nodes realises a different function (or model) from the ensemble, so that rather than learning a single ‘best’ fit, our neural network learns a distribution over possible fits.

The dropout rate relates to the variance we expect to see between different functions drawn from the ensemble — it acts as our prior distribution over functions. In general, a higher dropout rate will induce higher variance within the ensemble, as samples (or ‘ensemble members’) become less correlated. However, the dropout rate also affects the capacity of the neural network to represent complex functions, for example a high dropout rate of 0.9 would on average leave only 10% of the nodes of the network active for any given sample, effectively causing the ensemble to be composed of much smaller neural networks which would likely be weaker learners (Srivastava et al., 2014). In the same way that theory does not dictate the optimal neural network architecture for a given task, so too the optimal dropout rate is task (and architecture) dependent. By manually tuning the dropout rate in order to minimise loss on the evaluation set, a well-calibrated posterior predictive distribution can be achieved. For our geochemical mapping application, we found a dropout rate of 0.2 for the fully connected layers, and a spatial dropout rate of 0.5 for the convolutional layers (in which filters, rather than nodes, are dropped) to be effective. The manual tuning of dropout rates adds some time to the model development process, but it is just one of many hyper-parameters to consider in the design of the neural network (along with depth, layer width, activation function, convolutional kernel size, dilation, stride, pooling etc). It is worth noting that other approaches to Bayesian inference in deep neural networks have been proposed, in what is a rapidly developing area of research (we recommend the concise review by Wilson, 2020). We therefore remain open minded about what may emerge as the 'best' approach over the coming years, but have found Monte Carlo dropout to perform well in our task.

We now provide a deeper look at the principles behind Monte Carlo dropout as a Bayesian approximation, though for the complete details we refer readers to the original work of Gal and Ghahramani (2016). First, consider the simpler scenario of using our neural network without dropout. In this scenario, the gradient descent training procedure aims to find a fixed set of weights, $w*$, which maximise the likelihood: the probability of the data given the weights, $p(\boldsymbol{y}|w*)$. Given its enormous number of parameters, our neural network could potentially achieve this by fitting the mean of its Gaussian output, μ, directly through our training observations, and setting its variance, σ^2, close to zero everywhere - although this would undoubtedly be a case of overfitting the data. Regularisation techniques can be used to prevent this overfitting by penalising complexity, but regardless, the outcome would still be a fixed set of weights, $w*$, which provides no estimate of epistemic uncertainty.

In order to quantify epistemic uncertainty, we want to learn the posterior distribution of the weights, $p(w|\boldsymbol{y})$, given the data $\boldsymbol{y}$. To do so in the traditional way requires combining the likelihood, $p(\boldsymbol{y}|w)$, with a prior distribution for the weights, $p(w)$, through Bayes rule. In our case, our prior is constructed by assuming randomly initialised fixed weights β and, for each node of the network (each row of the neural network's weight matrices) a Bernoulli random variable $z \sim \text{Bern}(\pi)$ where $\pi = \Pr(z = 1)$. Then the distribution over weights is defined as $p(w) = \beta z$, which defines whether the weight β is 'active' ($z = 1$) with probability π or 'dropped out' ($z = 0$) with probability $1 - \pi$, where $1 - \pi$ is the dropout rate to be tuned manually as a hyper-parameter. Using Bayes rule the posterior is defined as

$$p(w|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|w)p(w)}{p(\boldsymbol{y})}, \tag{5.8}$$

but the Monte Carlo dropout approach provides an approximation whereby we obtain the posterior by training the fixed weights, β, while dropout is active at the rate we specify. Once training is complete, we take the Monte Carlo dropout distribution, $\beta_{TRAINED}z$, to be our posterior, so that

$$p(w|\boldsymbol{y}) = \beta_{TRAINED}z. \tag{5.9}$$

This approximation is efficient in that, once the fixed weights have been trained (for which we can use the standard efficient optimisers for deep learning), the Monte Carlo dropout samples immediately provide independent samples from the posterior, $p(w|\boldsymbol{y})$, with no burn in or thinning required, unlike samples obtained by MCMC methods (e.g. see Raftery and Lewis, 1996). However, it does mean that we are entirely dependent on learning an optimal set of fixed weights, $\beta_{TRAINED}$ (subject to the dropout rates we specify), in order that our approximate posterior results in a well-calibrated model. From the perspective of big data efficiency, this is in fact a selling-point of the Monte Carlo dropout approach, as it means we can use the established tools for training deep neural networks very efficiently, namely stochastic gradient descent in frameworks such as Tensorflow, and treat the dropout rate as a hyper-parameter to be tuned in the neural network design process.

To arrive at our trained weights, $\beta_{TRAINED}$, in Tensorflow we task stochastic gradient descent with optimising β to minimise the negative log-likelihood loss, $-log[p(\boldsymbol{y}|w)]$, which equates to maximising the likelihood. However, with dropout active, a different random sample of weights are updated at each iteration of training (i.e., for each mini-batch that the optimiser operates on, a different Monte Carlo dropout sample of weights are used). This provides Monte Carlo integration over $p(w|\boldsymbol{y})$ during training time, such that the quantity maximised is actually the (approximate) posterior predictive

$$p(Y|x,\boldsymbol{y}) = \int_w p(Y|x,w)p(w|\boldsymbol{y})dw, \tag{5.10}$$

in which the noise provided by Monte Carlo dropout, to induce a distribution over functions, prevents the posterior $p(w|\boldsymbol{y})$ (or alternatively $\beta_{TRAINED}z$) from collapsing to a maximum likelihood point estimate and forces the optimiser to optimise the entire predictive distribution. This enables epistemic uncertainty to be represented by the spread of possible functions in the posterior distribution (over functions) (Kendall and Gal, 2017) and improves generalisation relative to traditional non-Bayesian model fitting (Srivastava et al., 2014).

The uncertainty estimates obtained from Bayesian methods will always have some sensitivity to the choice of prior. In our case, the dispersion of our Monte Carlo dropout posterior is sensitive to our choice of the dropout rate. Fortunately, in a big data setting such as ours, we can use a large evaluation data set to help tune the dropout rate and any other hyper-parameters (more details of our specific setup follow in the next subsection). Our aim with tuning the dropout rate is that the stochastic gradient descent training process should arrive at a solution $\beta_{TRAINED}$ which corresponds to a model that fits the training data as closely as possible without overfitting, which would present itself as a degradation of predictive performance on the evaluation set. We can see this as it happens by monitoring the training and evaluation loss during the stochastic gradient descent process in Tensorflow. With dropout at a suitable rate, training loss will continue to decrease as stochastic gradient descent continues, while evaluation loss will reach a low plateau and stay there. This indicates that the resultant posterior corresponds to a model that well represents both the training data and the evaluation data, without becoming overconfident (i.e. overfitting). The better the uncertainty quantification, the better the predictive performance will be outside of the training data.

The posterior distribution is the key to capturing epistemic uncertainty — it represents the uncertainty in the function, or model, that the neural network has learned. Given the posterior distribution over weights, $p(w|\mathbf{y})$, the posterior predictive distribution of any output Y_s given the associated value of the input x_s, is given by

$$p(Y_s|x_s,\mathbf{y}) = \int_w p(Y_s|x_s,w)p(w|\mathbf{y})dw, \tag{5.11}$$

where $p(Y_s|x_s,w)$ is our Gaussian model (Equation 5.2). Note that Y_s in Equation 5.11 can be any value of the output variable, observed and unobserved alike, e.g. Y_s can be a location s not covered by the observed data $\mathbf{y}$. In practice we calculate this integral using Monte Carlo integration: simulating from the posterior predictive distribution for a given input, x_s, one sample at a time, where each sample is drawn from the Gaussian distribution using a different arrangement of weights w_i, sampled (by Monte Carlo dropout) from the posterior distribution $p(w|\mathbf{y})$. For more on simulation, and the spatial properties of resultant realisations, see appendix A.

5.2.4 Application to Geochemical Mapping

We applied our Bayesian deep neural network to the task of mapping stream sediment calcium concentrations, as log(calcium oxide), across the UK. This geochemical dataset, provided by the British Geological Survey, contains 109201 point-sampled calcium observations (as well as many other elements) measured by chemical assay of sediment collected from the beds of streams across the UK, approximately at random (Johnson et al., 2005). For our auxiliary grid, we use NASA's Shuttle Radar Topography Mission (SRTM) elevation data (Van Zyl, 2001), which we access via the Raster package in R (Hijmans, 2017; R Core Team, 2020) at a resolution of 30 arc-seconds, which translates to a horizontal resolution of 528m and a vertical resolution of 927m once projected into the

British National Grid coordinate system. In total for the UK this provides 611404 grid cell elevation values. We chose calcium concentration partly for its ability to differentiate rock types: calcium carbonate is the main constituent of chalk and a major constituent of limestones, but can be almost completely absent from deeper marine sediments deposited below the calcite compensation depth. We also chose calcium for how easy-weathering and mobile it tends to be in the surface environment, which means that it exhibits a complex relationship with terrain topography. Not only can we expect terrain features to be indicative of underlying bedrock composition (due to different rock compositions weathering differently, producing different surface expressions), but mobile elements will also be transported according to hydrological processes at the surface. In order to make good predictions of calcium concentrations our neural network therefore has to learn and combine knowledge of both bedrock and surface processes.

Constructing our study dataset required linking together the two input types (location information, and auxiliary information in the form of an observation-centred terrain image) to each observation of our target variable. Location information consists of the easting and northing values recorded for each observation in the G-BASE dataset, along with an elevation value extracted from the SRTM elevation grid at that location. The observation-centred terrain images each consist of 1024 elevation values extracted on a regular 32x32 grid centred at the location of the observation site. Bilinear interpolation of the elevation grid was used in all elevation extractions, in order to avoid aliasing issues in the terrain images. We extract the terrain images at a grid cell size of 250m, which means that the neural network has an 8x8km square window centred on each observation from which to learn its terrain features. Constructing the auxiliary information images through bilinear interpolation also means that we are not tied to the resolution of the underlying auxiliary grids. It is worth noting that our framework is capable of ingesting multiple auxiliary variable grids at once (multi-channel images as input), and there is no obligation to use only terrain. For instance, other sources of auxiliary information such as satellite or geophysical imagery may also be available.

In order to facilitate the learning of terrain features, we normalise each 32x32 cell image so that the centre point is at zero elevation. Features are then learned in terms of contextual relation to the sample site, rather than to absolute elevation. However absolute elevation, along with easting and northing, are provided explicitly as the second input to the neural network (after the convolutional layers — see Figure 5.1) in order to provide the network with awareness of overall location in the geographic space as well as awareness of local topography (i.e. from the auxiliary information). All location inputs are scaled to have mean of zero and standard deviation of one, with the elevation images also collectively scaled to have standard deviation one, but without further centring beyond setting the centre of each image to zero elevation.

In order to conduct our study, we split our assembled dataset at random into ten folds, of which one was set aside as a final test dataset (from which we report our prediction accuracy and calibration results in section 5.3), one was used as an evaluation set during the neural network training (to monitor loss on out-of-sample data to guide hyper-parameter tuning, such as tuning of dropout rates), and the remaining eight folds were used as the training set (which amounted to 87361 training observations). Different proportions could have been chosen for this hold-out validation scheme, but we have chosen tenths on the basis that this is common practice and that our held out test set is sufficiently large (10920 observations) for us to be confident in our results.

As mentioned in the introduction of the thesis, there is some controversy over the validity of randomly splitting data into training and testing folds in situations where the resultant folds may be non-independent (i.e., due to spatial autocorrelation). For spatial data like that used here, training-testing splits produced by random sampling are bound to be correlated due to the close proximity of their respective observations, and therefore easier to predict than splits which enforce geographical separation (i.e., ‘block cross-validation’ Roberts et al. 2017).

However, through experimentation Wadoux et al. (2021) found ‘that spatial cross-validation strategies resulted in a grossly pessimistic map accuracy assessment, and gave no improvement over standard cross-validation’. Therefore, despite the potential issue of non-independence within randomly sampled folds of spatial data, evaluation using randomly sampled test observations seems a viable approach given that we generally wish to evaluate the performance of a map-producing model within the spatial extent of data on which it was trained (i.e, to assess its interpolation performance) — this is the real-world use case in which the maps we produce here would be used, rather than to make extrapolations outside of the area in which observations have been collected (although extrapolative abilities are also of interest, and could potentially be where incorporating computer vision techniques brings the most value due to the potential to learn ‘generalisable truths’ from the landscape itself).

To train the neural network we used the Adam optimiser (Kingma and Ba, 2014) with learning rate of 0.001, weight decay of 1e-6, and a batch size of 4096. With the dropout rate tuned to a suitable value (we settled on 0.2 for the fully connected layers, and 0.5 for the convolutional layers), we found that our neural network was resistant to overfitting even when trained for a large number of epochs. This can be seen during training by monitoring predictive performance on the evaluation set, which plateaued after many epochs but did not degrade. This is a good sign, as it suggests that the posterior predictive distribution had become a good approximation of the true data distribution. We trained our neural network for 1000 epochs, which took about 25 minutes on a single GPU workstation (Nvidia Pascal Titan X GPU). Note that all of our result metrics are reported from the third dataset — the test dataset — which was not used during training at all. We would therefore expect the results we present in section 5.3 to well represent the general performance of our method in the context of predicting values of log(CaO) at unobserved locations within the UK.

We chose the SRTM and G-BASE datasets for their ease of access and use as well as for the complexity of the spatial relationships they contain, which we believe provide a good demonstration of the capability of our Bayesian deep learning approach for geostatistical modelling tasks. The methodology we present in this chapter is intended as a general framework for data-rich geostatistical applications where gridded auxiliary variables are available in addition to point-sampled observations of the target variable, and we would encourage readers to use the code we share alongside this chapter (available at `https://github.com/charliekirkwood/deepgeostat`) to test the approach on other geostatistical applications.

5.3 Results and Discussion

The national scale geochemical map that our Bayesian deep neural network has produced (Figure 5.3) is extremely detailed and appears to have successfully captured the complex relationships between our target variable: stream sediment calcium concentrations as log(CaO), and our auxiliary variable grid: terrain elevation. In addition to subjectively achieving good detail in the mapping task, our objective results on held out test data (unseen during the model training and hyper-parameter tuning procedure) are very encouraging. In a deterministic sense, the mean prediction from our Bayesian deep neural network explains 74% of the variance in our target variable (Figure 5.4a). The performance of the network in a probabilistic sense is less easily summarised by a single number, but a comparison of the predictive distribution with the true distribution on the held out test set (Figure 5.4b, Figure 5.4c) indicates a well-calibrated fit (Gneiting, Balabdaoui and Raftery, 2007). We have also measured performance using two proper scoring rules (Gneiting and Raftery, 2007): the continuous rank probability score (CRPS) and logarithmic score (Figure 5.4b), though these will be most useful in future comparisons with other models.

It is apparent in the observations data presented in Figure 5.4 that a cluster of observations share the same identical value of -3.69 log(CaO), which is the lowest

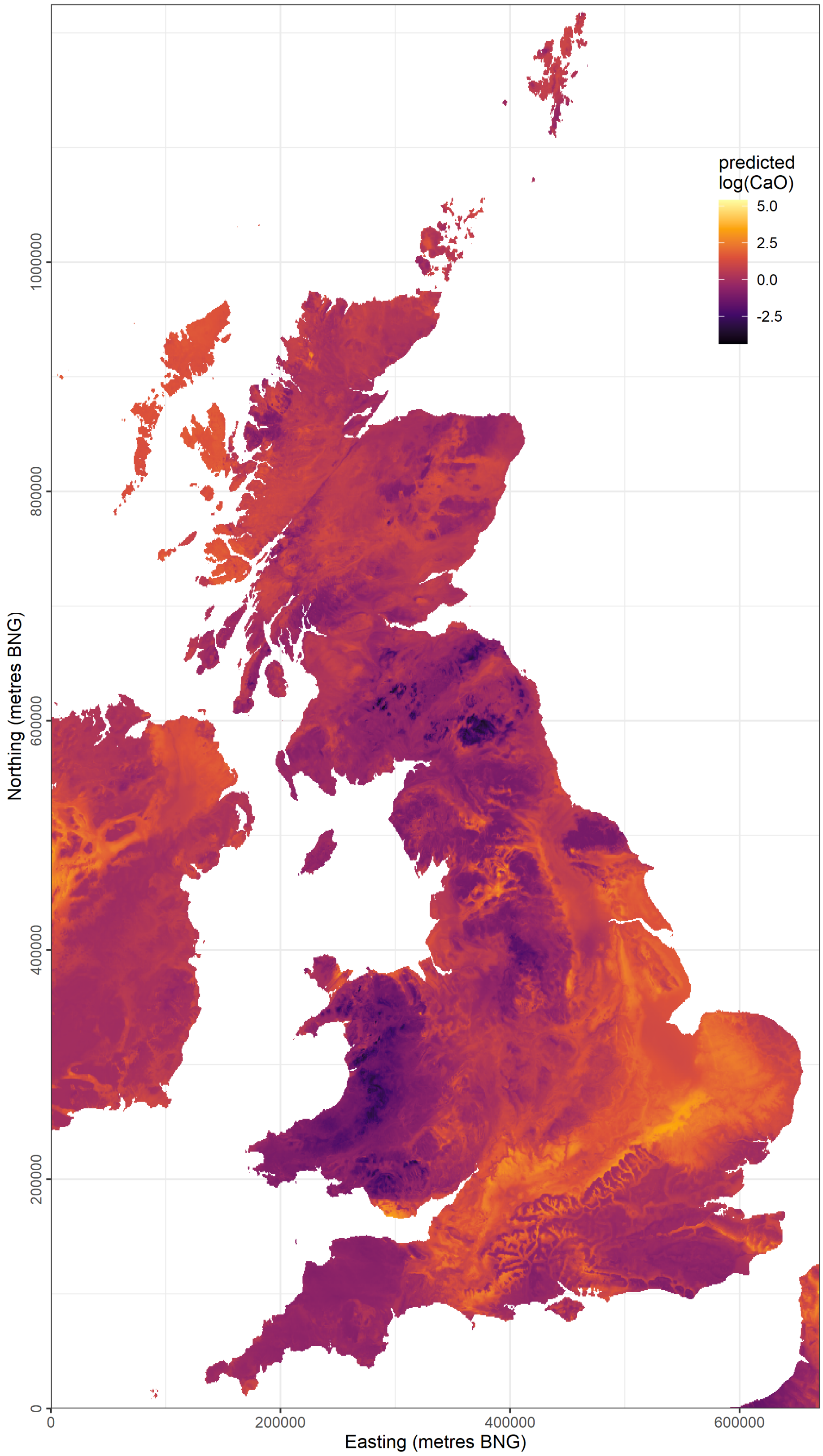


Figure 5.3: Predicted log(CaO) interpolated from stream sediment geochemistry observations across the UK using auxiliary information provided by a digital elevation model. This map shows the mean of our deep neural network's predictive distribution, which has captured complex relationships between terrain features and log(CaO)

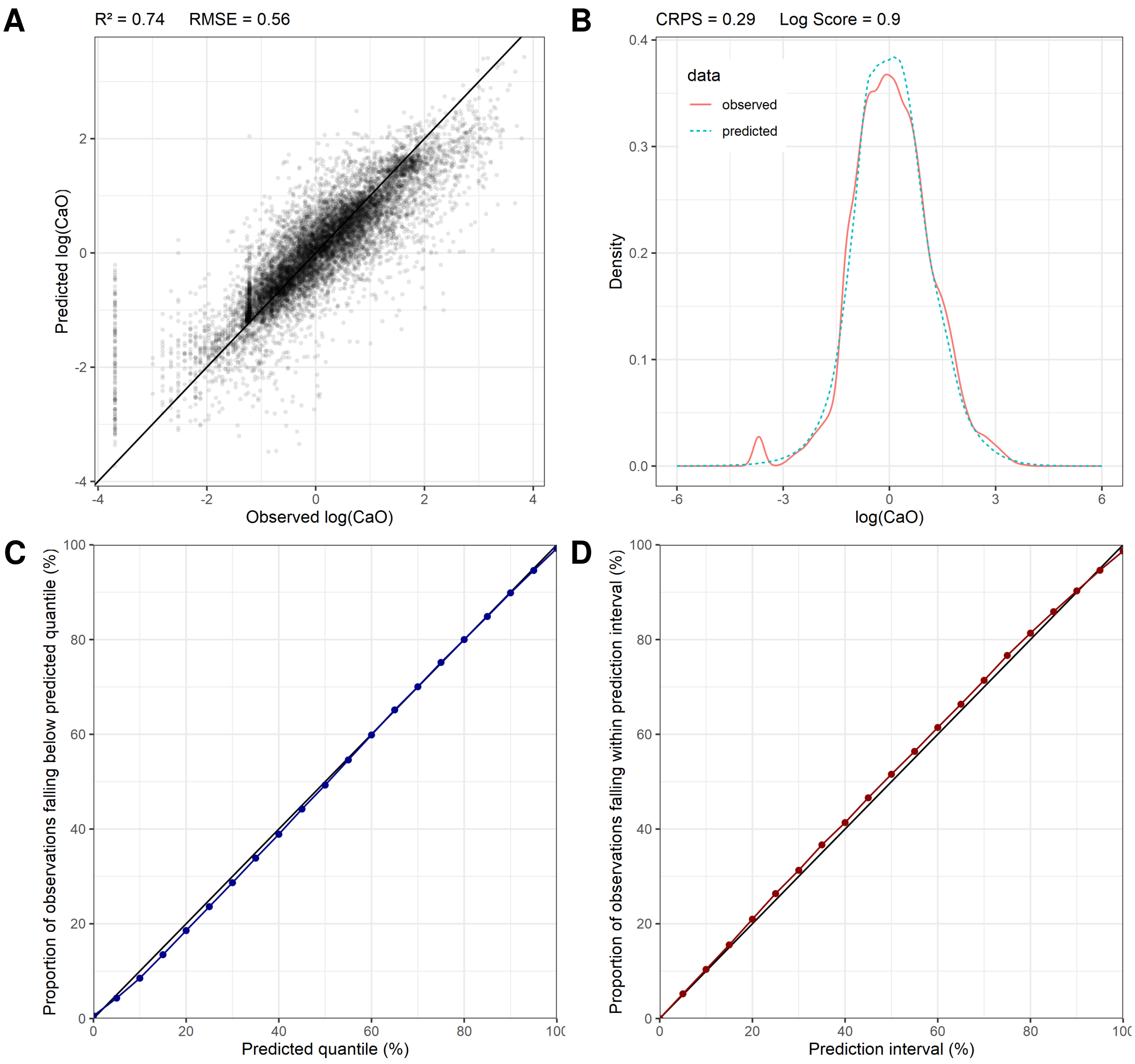


Figure 5.4: Evaluating our model's performance on the held out test data set (n = 10920). **a** Comparison of observed and predicted values, taking the mean of the predictive distribution as a deterministic prediction. **b** Density, **c** Q-Q and **d** prediction interval coverage plot comparisons of observed and predicted distributions

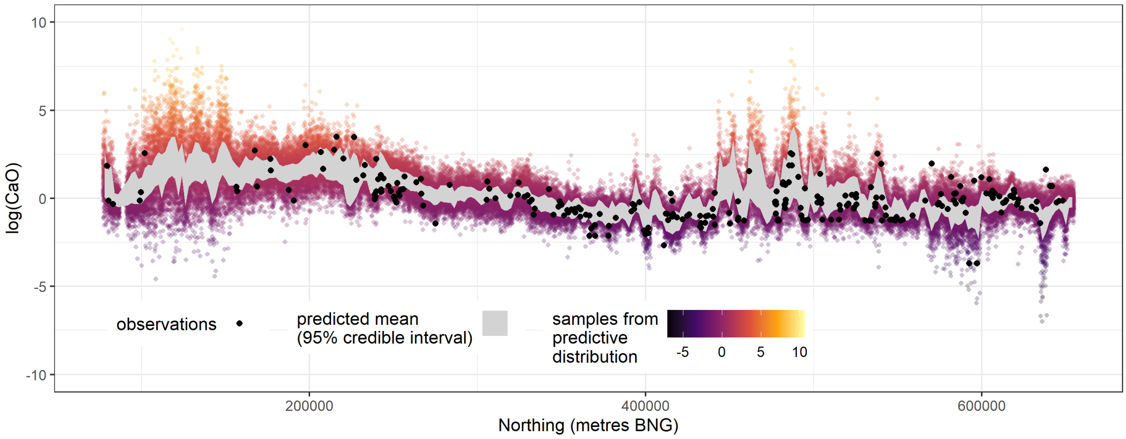


Figure 5.5: South-north cross section of our Bayesian deep neural network's output, running along a line at 400000 metres easting BNG. Also shown are all the observations within 500m either side of this line

value observed and corresponds to 0.025 weight % CaO on the linear scale. We believe that this set of identically valued observations are the result of a bug in error correction of the assay data for low calcium concentration samples and we are therefore not concerned by the discrepancy between our predictions and these observations (the neural network predicts that all should have higher values than recorded). It may even make sense to exclude these spurious observations from our comparisons, but we leave them in as a reminder that data sets in general are not necessarily free of defects, and that probabilistic data models like ours can in fact be a good way to identify statistically implausible defects like these.

Checking the coverage of our prediction intervals on the held out test data (Figure 5.4d), we find that 94.6%, 71.4%, and 51.6% of observations fall within the 95%, 70%, and 50% prediction intervals respectively. We take these numbers (which we may expect to be slightly skewed by the above mentioned spurious measurements in the low tail of the data) on a relatively large held out test set (10920 observations, 10% of the total dataset) as evidence that our Bayesian deep neural network is providing well calibrated probabilities, and therefore that it would be reasonable to use the predictive distribution to support decision making (Gneiting and Katzfuss, 2014).

We visualise the probabilistic capabilities of our deep neural network using a south-north section line through the map along the 400000 metres easting gridline (Figure 5.5). In doing so, we can see that the neural network is able to represent epistemic and aleatoric uncertainty independently as necessary to minimise loss. The credible interval for the mean varies spatially despite the fixed rate of Monte Carlo dropout, showing that the neural network is able to capture spatial variability in epistemic uncertainty. Likewise, the estimated aleatoric uncertainty also varies spatially, and can be high even where epistemic uncertainty is low. For example, we see this behaviour exhibited just south of 600000 metres northing. A quick check of the British Geological Survey's Geology of Britain Viewer (British Geological Survey, 2020) suggests that the geology of this section consists of the Yoredale group of interbedded limestone, argillaceous rocks and subordinate sandstone. The interplay of these compositionally different rock types on fine spatial scales that are unresolvable to the model (and to the geologists who classified the formation) is likely the reason for the comparatively high aleatoric uncertainty estimates in this area even with low epistemic uncertainty: the model has recognised that calcium concentrations here have higher variability at short spatial scales, and has increased its 'nugget' variance to account for this. This is one example of how probabilistic machine learning can be used as a guide towards discovery of further knowledge. By outputting a full predictive distribution, the Bayesian deep learning approach can provide probabilistic answers to all sorts of questions (e.g., Cawley et al., 2007; Kirkwood et al., 2020). Probabilities of exceedance at any location, for example, can be calculated simply as the proportion of probability mass in excess of any chosen threshold. We can also obtain individual realisations from the model through simulation. These realisations have spatial autocorrelation properties similar to that of the data - see appendix A for further details.

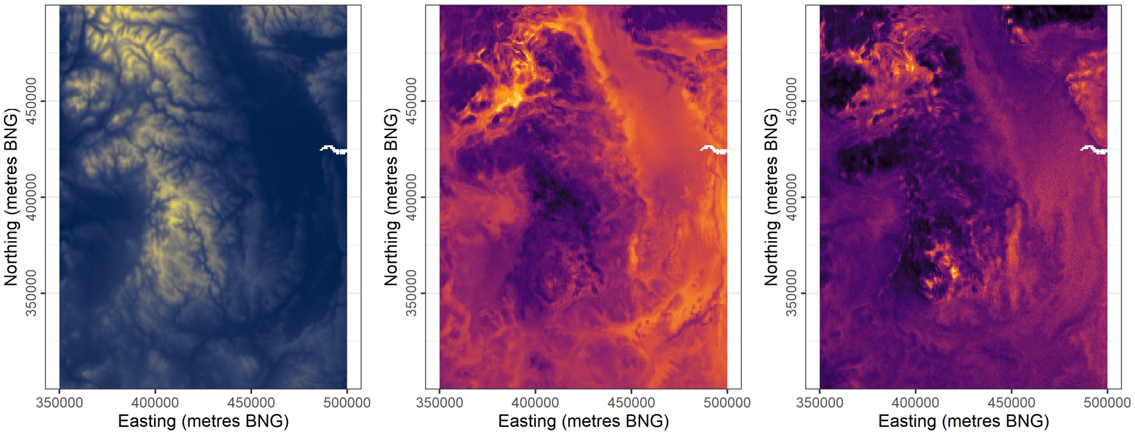


Figure 5.6: Zooming in on a local area. **a**, SRTM elevation data: the source of our auxiliary information. **b**, Predicted log(CaO): deterministic mean. **c**, Uncertainty of predicted log(CaO): standard deviation of posterior predictive distribution. All maps use linear colour scales where brighter = greater. The white inlet is the tip of the Humber estuary

We zoom in on the national scale map, and visualise predictive uncertainty in Figure 5.6. Viewing the deterministic mean map at this finer scale, and comparing it to the elevation map of the same extent reveals in more detail the ability of our deep convolutional neural network to learn the complex ways in which the distribution of our target variable relates to features of terrain. The same level of complexity is reflected in the uncertainty map, and shows that in addition to being very well calibrated (Figure 5.4), our Bayesian neural network is also very specific in its assignment of uncertainty to different spatial locations. In other words, our predictive distribution is both honest and sharp, which is desirable under the paradigm proposed by Gneiting, Balabdaoui and Raftery (2007) that probabilistic predictions should ideally be achieved by maximising the sharpness of the predictive distribution subject to calibration. Our combination of high map detail (in terms of both predictive mean and variance) and near-perfect coverage indicates that our Bayesian deep learning approach is successfully producing a predictive distribution that is both sharp and well-calibrated.

Our deep neural network is able to produce these specific and detailed outputs because it is interpolating not just in geographic space — as in traditional geostatistical models — but also in terrain feature space. This has important implications for mapping tasks. In traditional geostatistical models any predictions made outside the geographic extent of observations would be considered to be extrapolations, and are likely to have high error and uncertainty (Journel and Rossi, 1989). In our case, because our neural network is working in a hybrid space, predictions that would be considered out of sample geographically may still be within sample in terms of terrain features. While regression kriging may also be provided with terrain features as covariates, only a deep learning approach like ours has the capability to automatically learn complex terrain features for itself, and therefore has the potential to discover new ways to predict target variables based on fundamental relationships with the landscape rather than relying on spatial autocorrelation. This may have significant implications for applications like mineral exploration, where obtaining sensible predictions for unexplored regions is a key driver of new discoveries (Sabins, 1999). It also has implications for sample design in the age of ‘deep geostatistics’, which we leave for future work, other than to say that sample design ought to consider both the geographic space and the terrain feature space, and would likely be best guided by the epistemic uncertainty estimates of the deep models themselves.

The effects of fluvial processes on calcium are perhaps the most noticeable terrain-related effects captured in the map, with downslope ‘washing out’ of calcium apparent in valleys. In the zoomed-in region of Figure 5.6 for example, we can see elevated calcium concentrations in the channels that drain away from the calcium-rich area in the north-west of the figure (coordinates approx. 380000, 450000) even where these channels cross through otherwise low-calcium areas. This suggests that the convolutional branch of our neural network may have learned the concept of hydrological catchments and associated sediment transport directly from the data, a capability that Zuo et al. (2019) suggest will be important for improving robust mapping of geochemical anomalies in the future.

Further work will be needed to fully explore the capacity of our approach to learn complex physical process by example, and perhaps also to investigate the physical plausibility of the resultant predictions. However, the authors are aware of no other methods that could match the capabilities of our Bayesian deep learning approach in this geochemical mapping task. Numerical models may be able to represent physical processes more accurately, but they can struggle to accurately quantify uncertainties (e.g., Smith, 2013). Conversely, traditional geostatistical modelling approaches like regression kriging may do well at quantifying uncertainties, but have no capabilities in feature learning, which limit their capacity to fully utilise the information contained within auxiliary datasets. An approach known as topographic kriging (Laaha, Skøien and Blöschl, 2014) has been developed specifically for interpolation on stream networks, but this is unable to generate predictions outside of the manually designated stream network, and so is of limited use for general mapping applications. We therefore postulate that the Bayesian deep learning approach we present here represents an evolution in capabilities over previous geostatistical approaches, for its ability to automatically learn such complex relationships between target variables and the landscape.

5.4 Conclusion

Our Bayesian deep learning approach to spatial interpolation in the presence of auxiliary information achieves excellent predictive performance on held out test data according to both probabilistic and deterministic metrics, and in doing so produces maps with a high level of functional detail whose well-calibrated probabilities would be suitable for use in decision support. Our approach is unique in combining the following capabilities: I) automated information extraction from auxiliary variable grids via convolution, II) pure spatial interpolation abilities not dissimilar to that of ordinary kriging (each fully-connected layer in our neural network architecture would be equivalent to a Gaussian process in the limit of infinite layer width), III) outputting a well-calibrated predictive distribution by using Monte Carlo dropout for approximate Bayesian inference, and IV) the ability to handle very large datasets, including compatibility with GPU acceleration. As such, our approach brings new feature learning abilities and 'big data' efficiencies from deep learning to the established geostatistical domain of probabilistic spatial interpolation.

The major benefit of our end-to-end deep learning approach is the ability to automatically learn and utilise the complex relationships between auxiliary grids and target variables that it would not be possible or practical to manually specify, for example capturing the effects of fluvial processes on calcium distributions in our demonstration. Traditional geostatistical methods have no ability to automatically learn features, hence the significance of this work. By improving our ability to utilise auxiliary information in mapping tasks, we also reduce reliance on spatial autocorrelation for making predictions. This has the potential to improve the generalisation of geostatistical models, including beyond the spatial extents of a study area, owing to the potential of deep learning approaches to learn the 'fundamental truths' that may relate target variables to auxiliary grids. This potential remains to be explored.

5.5 Code and data

The code to reproduce this study is available at `https://github.com/charliekirkwood/deepgeostat` and includes functions to download NASA's SRTM elevation data via the raster package in R. We are unable to provide open access to the stream sediment geochemistry target variable dataset, however for academic research purposes, readers may request access to this dataset from the British Geological Survey at `https://www.bgs.ac.uk/enquiries/home.html` or by email to enquiries@bgs.ac.uk.

5.6 Appendix A: Simulation

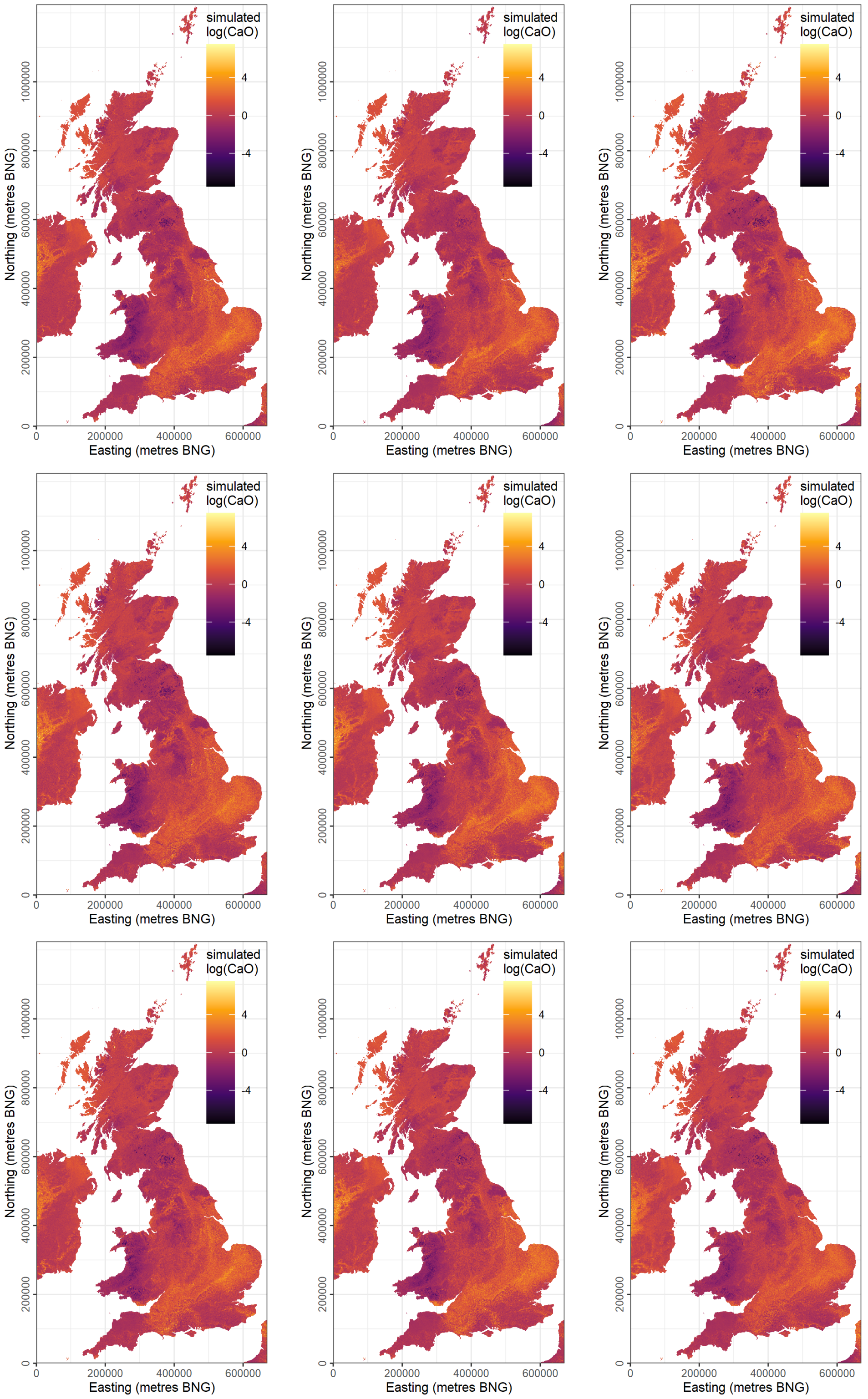


Figure 5.7: Nine simulated maps, or realisations, from our deep neural network. Each map is a sample from the posterior predictive distribution. Crossing your eyes to focus on two maps at once can help to make the differences more apparent

Our Bayesian approach means that we have not learned just a single ‘best fit’ for our neural network, but a distribution over possible fits (see subsection 5.2.3) from which we can simulate different spatially coherent maps, or realisations (Figure 5.7). Each realisation presents predictions from a possible model fit, which collectively construct our posterior predictive distribution, which itself represents our current state of knowledge. Each realisation can therefore be thought of as depicting what we might observe if the data collection procedure was repeated (at all grid cells of the map), subject to our current state of knowledge. We simulate these realisations by sampling from our posterior predictive distribution,

$$p(Y_s|x_s,\mathbf{y}) = \int_w p(Y_s|x_s,w)p(w|\mathbf{y})dw. \tag{5.11}$$

We do so by iterating two steps. First, we sample $w_i \sim p(w|\mathbf{y})$, which is to say we sample one configurations of weights from our posterior distribution. Second, we sample $Y_{s_i} \sim p(Y_s|x_s,w_i)$, in other words, for each location s across the map we sample one data value from our sampling distribution (the Gaussian output of our model) conditional on both the inputs to the neural network at that location and the configuration of weights from the first step. The form of our model — in which we represent aleatoric uncertainty using independent Gaussian noise — means that, conditional on x_s and w_i, the simulated values Y_s are independent for each spatial location s. This independent noise can also be thought of as our model's 'nugget effect' (Clark, 2010).

In practice, due to the fact we are using Monte Carlo dropout, making predictions across an entire map using the same sampled configuration of neural network weights, w_i, requires freezing the dropout mask for multiple calls to Tensorflow's predict function (one call for each grid cell of the map). We have provided code to achieve this, as this functionality is not built in to Tensorflow, which, when Monte Carlo dropout is active, would normally sample a new configuration of weights for each individual prediction, preventing spatially coherent maps of posterior predictive samples (i.e. realisations) from being obtained.

Whether or not we use dropout mask freezing in order to realise spatially coherent realisations, the posterior predictive distribution at any location, $p(Y_s|x_s, \mathbf{y})$, remains the same. In this chapter we have focused on the quality of our deep neural network's posterior predictive distribution, as assessed by its ability to provide good probabilistic predictions at the locations of unseen held out test data (see Figure 5.4 and section 5.3). This is the general use case that we envision our Bayesian deep learning approach being used for. However, for some applications users may be interested in the properties of individual realisations in addition to the properties of the predictive distribution overall. For example, in resource estimation and mine planning, obtaining realisations that fit the main characteristics of the revealed reality (Journel, 1974, of which spatial autocorrelation is seen as most important) has enabled more efficient optimisation of mining activities (Dimitrakopoulos, 1998; Dimitrakopoulos, 2018; Menabde et al., 2018) and analysis of risk (Vann, Bertoli and Jackson, 2002).

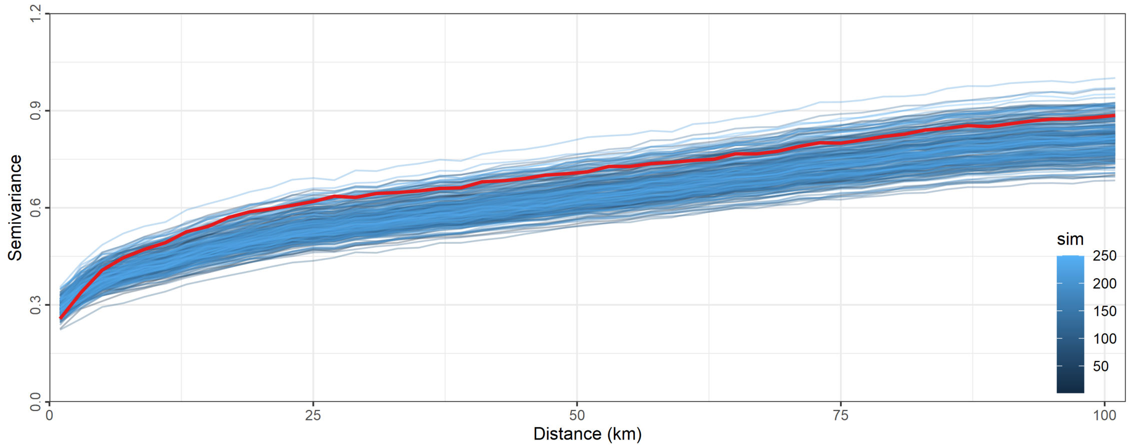


Figure 5.8: In blue: semivariograms for 250 simulations of log(CaO) values predicted at the locations of the held out test observations (n = 10920). In red: semivariogram of the held out test observations. Variograms produced using a bin size of 2km

Unlike traditional geostatistical approaches, the deep learning approach we present here is not parameterised to model spatial autocorrelation specifically. This has benefits, such as freeing us from assumptions of stationarity and isotropy, but the flexibility of our deep neural network could come at a cost in terms of our model's ability to simulate realisations with spatial autocorrelation properties that match those of observations. To investigate this, we have checked the spatial autocorrelation of 250 simulated realisations against that of the held out test data by comparing variograms (Figure 5.8). Each realisation's variogram is calculated by taking its values at the same 10920 locations as the held out test observations so as to eliminate any differences that might arise from considering different locations. As is to be expected, each of our realisations displays slightly different spatial autocorrelation properties, which results in a distribution of semivariances at each lag distance (we are using 2km bins).

We find that for all lag distances (Figure 5.8) the semivariance of the observations is within the range of the semivariances of the simulated realisations, suggesting that overall there is reasonable agreement between the spatial autocorrelation of realisations and the spatial autocorrelation of the data. To more critical eyes, there is some indication that realisations may on average have slightly too much 'nugget' variance (too much variability at zero distance) while not having quite enough variability at longer ranges (Figure 5.8), however we reserve making more absolute judgements of these higher-order properties of our model's output for further work and testing — in this study our priority has been point-wise predictive performance and calibration (Figure 5.4). For use cases where the spatial autocorrelation of realisations is a priority, we would recommend further investigation into these properties of Bayesian deep learning approaches like ours.

Overall, on the basis of this comparison of simulation and observation variograms using held out test data (Figure 5.8) it appears that, in addition to providing a well-calibrated and sharp predictive distribution (Figure 5.4 and section 5.3), our Bayesian deep learning approach also produces realisations with similar spatial autocorrelation properties to the observations. Each simulated realisation represents values we could observe if we were to repeat the data collection procedure, subject to our current state of knowledge (as represented by the posterior predictive distribution; Gelman et al., 2013). An additional consideration is the extent to which this similarity in spatial autocorrelation properties is influenced by the non-independence of the held out test data in this study (due to spatial autocorrelation in the dataset, and the use of random sampling rather than geographic block sampling to derive the test dataset). Experimentation with different schemes for producing test datasets with different levels of dependence (e.g. random sampling, block sampling) would provide insight into how well the deep model's apparent abilities to capture the spatial autocorrelation structure of the data hold up when making predictions far out-of-sample.

It is an additional benefit of our Bayesian approach that obtaining realisations comes at no additional computational cost over what is already required to make general predictions with our model. This is because Monte Carlo sampling must be used to obtain our otherwise intractable posterior predictive distribution, and each of these samples is a realisation. So simulation is an innate part of our Bayesian approach. It is also the case that Monte Carlo dropout neural networks are a computationally efficient Bayesian method. On a single GPU workstation, it takes under 30 minutes to train our model on 87361 training observations. Simulation then takes about 5 seconds per realisation for the entire 0.6 million grid cell map.

Chapter 6

A deep mixture density network for outlier-corrected interpolation of crowd-sourced weather data

As the costs of sensors and associated IT infrastructure decreases — as exemplified by the Internet of Things — increasing volumes of observational data are becoming available for use by environmental scientists. However, as the number of available observation sites increases, so too does the opportunity for data quality issues to emerge, particularly given that many of these sensors do not have the benefit of official maintenance teams. To realise the value of crowd sourced 'Internet of Things' type observations for environmental modelling, we require approaches that can automate the detection of outliers during the data modelling process so that they do not contaminate the true distribution of the phenomena of interest. To this end, here we present a Bayesian deep learning approach for spatio-temporal modelling of environmental variables with automatic outlier detection. Our approach implements a Gaussian-uniform mixture density network whose dual purposes — modelling the phenomenon of interest, and learning to classify and ignore outliers — are achieved simultaneously, each by specifically designed branches of our neural network. For our example application, we use the Met Office's Weather Observation Website data, an archive of observations from around 1900 privately run and unofficial weather stations across the British Isles. Using data on surface air temperature, we demonstrate how our deep mixture model approach enables the modelling of a highly skilled spatio-temporal temperature distribution without contamination from spurious observations. We hope that adoption of our approach will help unlock the potential of incorporating a wider range of observation sources, including from crowd sourcing, into future environmental models.

6.1 Introduction

As environmental scientists, the volumes of observational data that we have at our disposal are ever increasing. Movements such as the Internet-of-Things (IoT), exemplified in the context of weather data by the Met Office's Weather Observation Website (Kirk, Clark and Creed, 2021), have enabled near real-time collection and sharing of environmental data by low cost sensing equipment around the world. The implications of this are numerous, but where once the challenge was to collect sufficient data for specific modelling problems (often hindered by expense), now often the challenge is to maximise the utility of the high volumes of data that we already have. The rise of 'data science' can be viewed, to some extent, as a response to this shift in challenges.

In the case of weather modelling and forecasting, it is likely that harnessing the ever-growing network of IoT type environmental sensor data, in addition to the observations provided by traditional official weather stations, will facilitate the development of higher-precision, finer-scale models which can serve more specific predictions to stakeholders (Bell, Cornford and Bastin, 2015; Chapman, Bell and Bell, 2017). Linked to this, a benefit of IoT type sensor data is that these observations have the potential to be more representative of the weather experienced by the device owners themselves (e.g. due to private weather stations being located at homes), rather than representative of remote rural locations (as tends to be the case for official weather stations). The adoption of data from these unofficial private weather stations and IoT type environmental sensors could therefore enable models to provide more specific, personalised weather information at hyperlocal scales.

However, while crowd-sourcing weather data can greatly increase the number of observations being made, and the number of unique locations which are observed, it also opens the door to data quality issues owing to the low cost, low maintenance nature of unofficial weather stations compared to official weather stations. The traditional way of addressing data quality issues is to have some form of manually-guided rules based quality control procedure to subjectively approve or deny the inclusion of each sensor's observations into downstream models. While a manually-guided procedure may seem the best approach in terms of having complete hands-on control at an individual observation level, such an approach will tend to suffer from scalability issues as the number of sensors increases, and is difficult to achieve consistently through space and time.

As the number of sensors enters or exceeds the thousands, it becomes necessary to automate aspects of the quality control procedure in order to keep up with the scale of the task. Common approaches include statistical time-series analysis or rule-based outlier detection algorithms to help identify sensors that are producing data of questionable quality, which can then be excluded from input into subsequent models. Here we propose a unified approach whereby detection of outliers is achieved as part of a downstream statistical data model itself: in this case a Bayesian deep neural network based spatio-temporal interpolator of crowd-sourced temperature observations collected by the Met Office's Weather Observation Website, with a mixture model or mixture density network architecture to enable automatic identification and correction of outliers as part of the modelling process.

In this chapter we proceed by briefly providing some background on IoT sensor data and its potential benefits for environmental modelling applications, as well as an overview of existing methods for outlier detection. We then introduce our deep mixture model approach for spatio-temporal interpolation with simultaneous probabilistic outlier detection, using an example dataset composed of surface temperature observations collected by the Met Office's Weather Observation Website. By adopting a mixture model approach, we incorporate our knowledge about data issues into the data model through our choice of probability distributions, which provide our likelihood function. This, in combination with our Bayesian approach allows us to quantify both aleatoric and epistemic uncertainties — uncertainty about the data and uncertainty about the fit of the model — in order to provide a well-calibrated posterior predictive distribution. Bayesian deep learning frameworks (here we use Tensorflow Probability) allow us to combine the above benefits of Bayesian statistical modelling with the flexibility and scalability of deep neural networks.

We assess the performance of our model on held-out test data, finding our approach to be successful in filtering outliers in order to provide 'clean' spatio-temporal interpolation that is free from outlier-induced anomalies. In addition, as our probabilistic approach (including epistemic uncertainty via Monte Carlo dropout as a Bayesian approximation) provides a well-calibrated predictive distribution rather than single point predictions, it therefore provides useful information for downstream applications and decision making.

6.2 Background

6.2.1 IoT sensor data in environmental modelling

Within the field of environmental modelling, the concept of 'models of everywhere' (Beven, 2007) has been proposed. This is a concept which stems specifically from hydrology but is applicable across environmental sciences. The concept aims to "change the nature of the modelling process, from one in which general model structures are used in particular catchment applications to one in which modelling becomes a learning process about places" (Beven and Alcock, 2012). This idea is driven by the need to constrain uncertainty in the modelling process in order to support policy setting and decision making (Blair et al., 2019a). The concept is a reaction to the shortfalls of the use of 'generic models', in which spatially-discretised (gridded) predictions are likely to fail to provide well-calibrated probabilistic predictions for the specific locations or areas (not grid cells) which are of interest to stakeholder decision making (Beven et al., 2015). However, the issue is not simply one of scale, and increasing the resolution of imperfect models does not solve the problem of what Beven et al. (2015) term 'hyperresolution ignorance' in that uncertainty about parameters will still exist, and a model outputting at finer scales will not necessarily be providing more information. This is a topical issue for weather forecasting too, as numerical weather prediction models continue to increase in resolution.

Technological challenges such as limited computational power have slowed the adoption of the ‘models of everywhere’ concept, but Blair et al. (2019b) propose that data science, including cloud computing infrastructure, may provide the means to make the ‘models of everywhere’ concept a reality by using data mining techniques to combine information from remote sensing and in-situ earth monitoring systems in data-driven models. This includes live-updating IoT sensors (Atzori, Iera and Morabito, 2010; Nundloll et al., 2019) such as the unofficial weather stations which provide data to the Met Office’s Weather Observation Website (Kirk, Clark and Creed, 2021). These provide a greater number of observations from more numerous unique sites than traditional monitoring systems (e.g. traditional weather stations), and the combination of increasingly dense observations (by IoT sensors) and machine learning may allow data-driven models to supersede alternative modelling approaches, such as ‘generic’ physics based models. However, IoT sensors have greater potential for data quality issues over traditional monitoring systems owing to their cheaper costs, less stringent maintenance, and being more numerous. Therefore, in order to maximise the benefits of IoT sensor data for environmental modelling, issues of data quality have to be addressed as part of the solution.

We propose that the approach we present here, which uses Bayesian deep learning to combine information from remote sensing and in-situ earth monitoring in order to provide specific and well-calibrated predictions for any point within the extent of observed space and time, does satisfy the ideals behind the ‘models of everywhere’ concept. As such, it can be viewed as an example of the kind of large scale data-driven environmental modelling that is likely to become more feasible as computing power continues to increase - putting ‘models of everywhere’ at our fingertips.

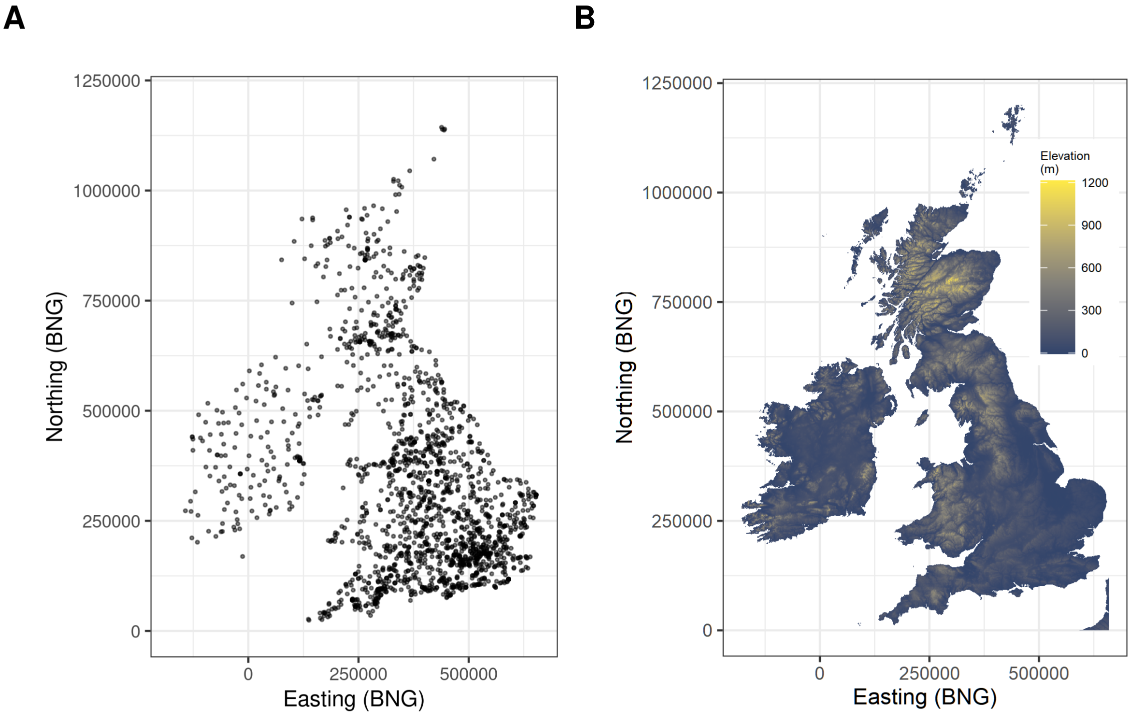


Figure 6.1: **A**, The locations of the 1893 private weather stations across the British Isles that provide crowd-sourced data to the the Weather Observation Website. **B**, SRTM elevation data for the British Isles which our model uses as auxiliary information.

6.2.2 Outlier detection

There are many possible approaches to outlier detection, ranging from fully-manual data checking, to manually designed rule-based filters, to statistical and machine learning based systems, which may include both supervised and unsupervised learning (with supervised learning having the downside that it requires the creation of manually labelled training datasets in advance; e.g. Nesa, Ghosh and Banerjee, 2018). For a full review of outlier detection techniques we refer the reader to Wang, Bah and Hammad (2019), who provide a general review of developments in outlier detection since the year 2000. In addition, Ayadi et al. (2017) provide a review of techniques specifically for wireless sensor networks, including a comparison of the respective pros and cons of statistical, nearest-neighbour, artificial intelligence, clustering and classification based approaches (although these categories have some overlap). Napoly et al. (2018) propose a combination of rule-based and

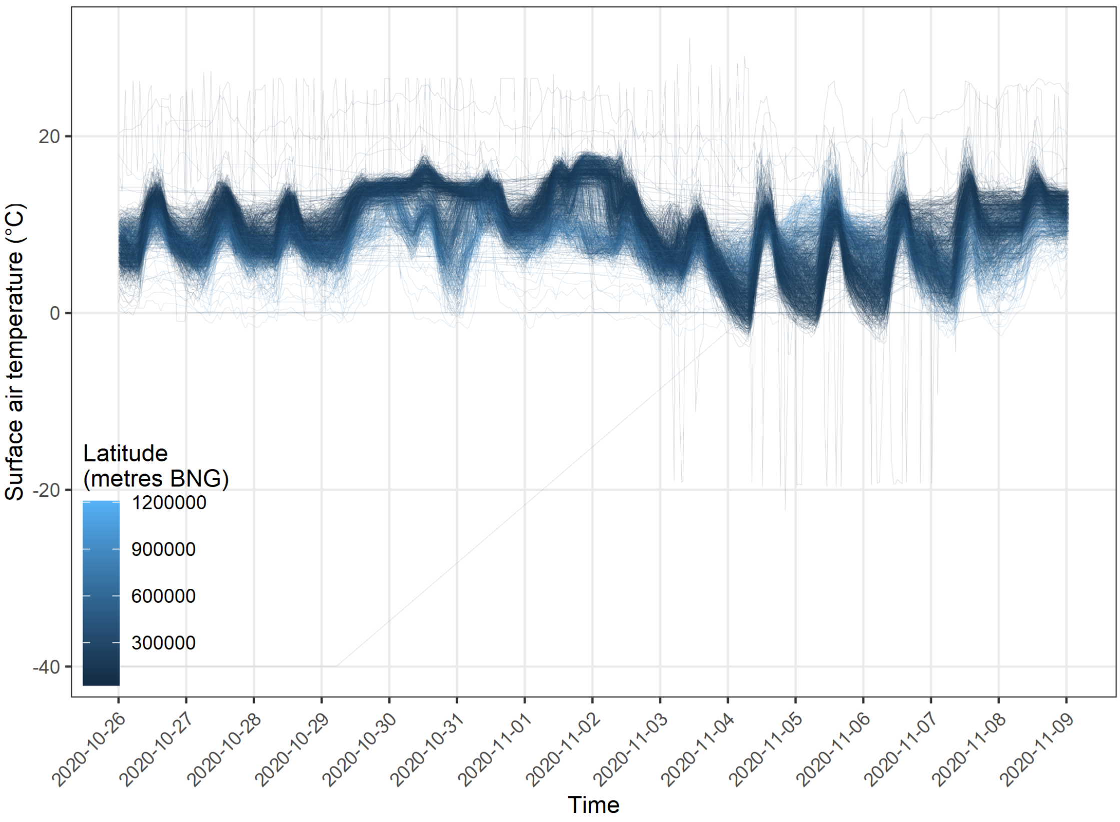


Figure 6.2: Time-series visualisation of our Met Office Weather Observation Website temperature data for a seven day period in November 2020. Each line represents one weather station, which are coloured such that higher latitudes are a lighter shade. Note how the data is not clean, but contains spurious observations on either side of the central distribution

z-score thresholds for outlier detection in crowdsourced air temperature data. This approach has been adopted by other authors (e.g. Venter et al., 2020; Zumwald et al., 2021) but this is not the approach we take.

The approach we adopt for this study is a regression approach using a deep neural network mixture model — or mixture density network (Bishop, 1994) — through which we represent the conditional distribution of reported temperature values as a mixture of a Gaussian and a Uniform distribution, with parameters learned by our deep neural network. We explain the full details of the approach in subsequent sections, but in brief terms, our approach incorporates outlier detection into the spatio-temporal modelling process itself, by having the neural network learn the probability that an observation is an outlier (whose values are best explained as having been generated by the Uniform distribution) as an unsupervised subtask to the overall supervised spatio-temporal modelling task. The benefit of this holistic approach is that it allows the user to incorporate knowledge about data issues into the data model itself through the use of suitable probability distributions, and makes for more seamless model checking when compared to a two-stage procedure of separate outlier-detection followed by data modelling.

6.3 Method

6.3.1 Dataset

We demonstrate our approach using surface air temperature data from the Met Office's Weather Observation Website archives (Kirk, Clark and Creed, 2021). These data contain observations from 1893 unique IoT type weather stations (Figure 6.1), from which we have taken a continuous 14 day window from 2020/01/26 to 2020/11/09 to use as our dataset in this study. The data provide our target variable, surface air temperature in degrees Celsius, as well as spatio-temporal location information in the form of British National Grid (BNG) Easting and Northing, and a timestamp. Collectively these Weather Observation Website weather stations record 8000 observations per hour on average, which equates to about four observations per site per hour, although this varies by site. Each sensor records observations at different intervals, rather than synchronously at set times, so that collectively the observations provide good coverage across continuous time (Figure 6.2).

In addition to using the Weather Observation Website data, we also make use of gridded UK elevation data as covariate or auxiliary information in order to help inform the spatio-temporal interpolation. The data used comes from NASA's Surface Radar Topography Mission (SRTM; Farr et al., 2007) and is accessed via the Raster package in the R programming language. The elevation data is rasterised with a grid size of 528 by 927 metres (longer latitudinally than longitudinally), resulting in 0.66 million grid cell elevation dataset covering the UK and Ireland.

For input into our model, we extract terrain elevation images centred on each observation (in the case of training) or location to be predicted. The images extracted have a resolution of 32x32 grid cells with a grid cell size of 500m (we use bilinear interpolation so that the image resolution is not locked to the overall digital elevation model resolution). These images provide auxiliary information, from which the convolutional layers of our deep neural network learn to extract useful contextual covariates (e.g. as explained in Kirkwood, 2020; Kirkwood et al., 2022) for the task of spatio-temporal interpolation of surface air temperature data. Illustrative examples could include slopes facing the sun that warm faster, or valleys that channel cool air from cold mountainous areas. There are likely to be many such complex interactions between the landscape and surface air temperatures, and by providing elevation data as images to our deep neural network we allow them to be learned from data. Further details of the preparation of our dataset for model training, evaluation, and testing are provided in the section ‘Practical setup’.

6.3.2 Mixture model concept

We design our model to address three considerations: 1) The capacity to represent our target phenomenon (a spatio-temporally varying temperature distribution in this case), under the assumption that outliers can be objectively identified and excluded. 2) The capacity to successfully identify outliers. 3) A means by which to achieve both 1 and 2 simultaneously within a single probabilistic data model.

At the heart of our model is a two-part mixture probability distribution whose individual component distributions - p_{signal} and $p_{outlier}$ - represent the two classes of observation that we judge to exist within our dataset, as evidenced by exploratory visualisation of the data (Figure 6.2). These are 1) The ‘true’ signal distribution of our target phenomenon, which we assume here is a Gaussian distribution as is common for temperature measurements, and 2) the outlier distribution, in this case we choose a Uniform distribution ‘catch-all’ that can account for the generation of spurious observations by biased or faulty weather stations. It is worth noting that the selection of these distributions is a modelling choice, and that different target variables are likely to warrant the use of different distributions in the model output, from which the likelihood is derived (the probability of the data given the model).

We then introduce parameter θ – the probability that an individual data point comes from the "true" Gaussian distribution of temperature. Equivalently, $1-\theta$ is the probability that a data point is spurious and therefore comes from the uniform distribution. More formally, let $y_{s,t}$ denote the temperature at location s and time point t. The probability distribution of $y_{s,t}$ is defined as:

$$\begin{aligned} p(y_{s,t}) &= \theta_{s,t} p_{signal}(y_{s,t}) + (1-\theta_{s,t}) p_{outlier}(y_{s,t}), & (6.1) \\ p_{signal}(y_{s,t}) &= \mathrm{N}(\mu_{s,t}, \sigma^2_{s,t}) \text{ with density } \frac{1}{\sqrt{2\pi}\sigma_{s,t}} \exp\left\{ \frac{1}{2\sigma^2_{s,t}} (y_{s,t} - \mu_{s,t})^2 \right\} & (6.2) \\ p_{outlier}(y_{s,t}) &= \text{Uniform}(\mu_{s,t} - 50, \mu_{s,t} + 50) \text{ with density } \frac{1}{100}. & (6.3) \end{aligned}$$

The "true" temperature distribution is therefore assumed Normal with mean $\mu_{s,t}$ and variance $\sigma^2_{s,t}$, while the spurious observations are centered at the "true" mean $\mu_{s,t}$ but are allowed to vary uniformly around this mean. This range of 100°C was chosen from exploratory data analysis and was deemed sufficient to capture the outliers in the data.

A perhaps more intuitive way of interpreting this model, is to introduce a latent binary variable, $Z_{s,t}$ where $\Pr(Z_{s,t} = 1) = \theta_{s,t}$ and $\Pr(Z_{s,t} = 0) = 1 - \theta_{s,t}$. The probability model for temperature $y_{s,t}$ conditional on $Z_{s,t}$ is then:

$$\begin{aligned} y_{s,t} | Z_{s,t} = 1 &\sim \text{Normal}(\mu_{s,t}, \sigma^2_{s,t}) & (6.4) \\ y_{s,t} | Z_{s,t} = 0 &\sim \text{Uniform}(\mu_{s,t} - 50, \mu_{s,t} + 50) & (6.5) \\ Z_{s,t} &\sim \text{Bernoulli}(\theta_{s,t}). & (6.6) \end{aligned}$$

We can think of $Z_{s,t}$ as the result of a 'coin toss' where at any given location s and time point t, we can get a spurious observation with probability $1-\theta_{s,t}$. Note that $\theta_{s,t}$ varies with space and time, in order to flexibly capture the flawed data points (as opposed to assuming a constant θ).

Note further that the fact that $\mu_{s,t}$ appears in the model for the spurious data points, i.e. the Uniform distribution, allows some information from such data points to be utilised. This is based on a belief that on average, the flawed data are centred on the true mean, such that negative and positive biases cancel each other out (though in practice this may well be optimistic; Bell, Cornford and Bastin, 2015). Note that any flawed data point which is much further from the mean $\mu_{s,t}$ than the Normal$(\mu_{s,t}, \sigma^2_{s,t})$ distribution implies, will be "absorbed" by the Uniform distribution so that the Normal part of the model can be interpreted as the model for the true temperature process. As such, predictions from the Normal$(\mu_{s,t}, \sigma^2_{s,t})$ part after the model is implemented, can be seen as "corrected".

6.3.3 Network architecture

The parameters of our mixture distribution are $\mu_{s,t}$, $\sigma^2_{s,t}$ and $\theta_{s,t}$. We therefore require that our model has the capacity to learn to optimise these parameters in relation to space and time so that predictions from (6.2) are a reasonable representation of the real data generating processes at location s and time t (as we assess through model checking against held-out test data).

To achieve this, our neural network architecture consists of two halves, which we term the signal network and the outlier network. The signal network is tasked with learning the parameters, $\mu_{s,t}$ and $\sigma^2_{s,t}$, of our 'true' Gaussian distribution, which are conditioned on the space and time variables that we provide as inputs to the model (the details of which we explain in subsequent sections). The outlier network meanwhile is simply tasked with learning $\theta_{s,t}$ or in other words the probability that an observation is an outlier, which is conditioned on site ID (which we provide one-hot encoded) and time. One-hot encoding means representing our n site IDs as n separate predictor variables, to which we assign the value 1 only if an observation corresponds to that site, otherwise a value of 0 is assigned. The one-hot encoding approach allows us to input categorical variables into the neural network in a sensible way. We provide site IDs (rather than more general spatial variables such as easting and northing) to the outlier network because it has no need to learn generalisable patterns, its sole purpose is to identify outliers probabilistically during the training phase, and this ability is improved by making the task as simple as possible. Overfitting is not a concern since the outlier network serves no purpose in the spatio-temporal interpolation beyond the training stage.

From the perspective of deep neural networks as “black boxes”, we can view our signal and outlier networks simply as function approximators that learn to provide optimal values of their respective output parameters, such that

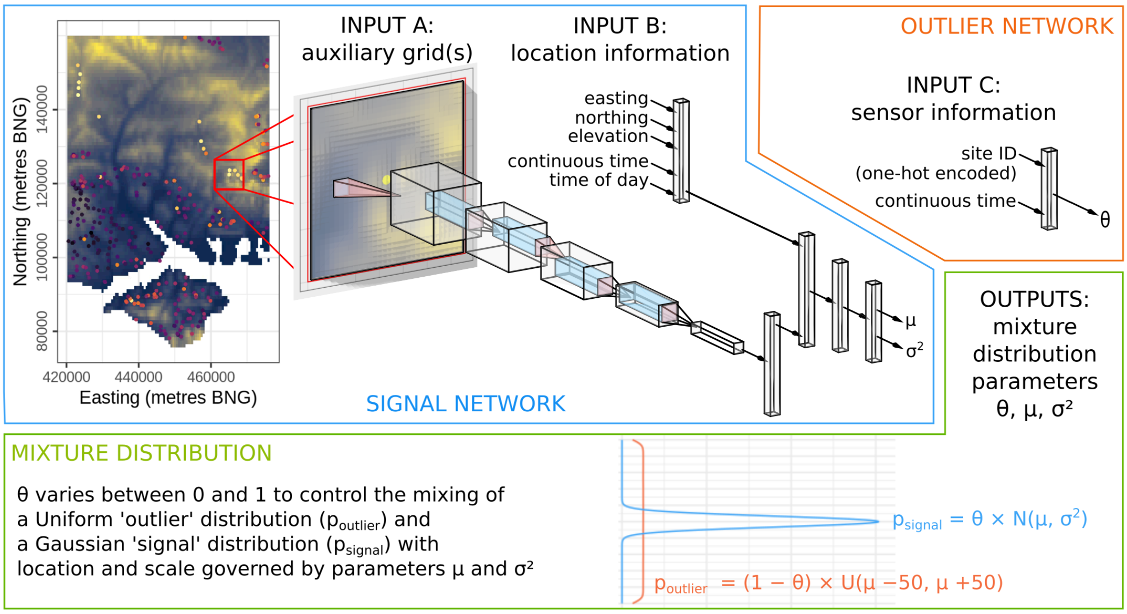
$$\begin{aligned}(\mu_{s,t}) &= g(f_{signalnetwork}(x_{space,time})), && (6.7)\\ (\sigma^2_{s,t}) &= h(f_{signalnetwork}(x_{space,time})), && (6.8)\end{aligned}$$

and

$$\text{logit}(\theta_{s,t}) = f_{outliernetwork}(x_{site,time}), \quad (6.9)$$

however we have designed the architecture of the two branches — signal network and outlier network — in line with their specific goals. The architectures of each branch, and the specific space and time variables that they take as inputs, are explained in the following paragraphs, accompanied by Figure 6.3 as a visual aid.

Our signal network architecture (Figure 6.3) is designed for terrain-aware interpolation, which it achieves through the combination of a convolutional branch to derive relevant terrain features from gridded auxiliary information (e.g. terrain elevation, satellite imagery), and a fully-connected branch for interpolation in space and time. The combined effect is to achieve spatio-temporal interpolation in a hybrid space that includes local terrain context so that, for example, the differences between valleys and hill-tops (and anything relevant about their orientations) can



θ varies between 0 and 1 to control the mixing of a Uniform 'outlier' distribution ($p_{outlier}$) and a Gaussian 'signal' distribution (p_{signal}) with location and scale governed by parameters μ and σ²

Figure 6.3: The architecture of our deep neural network, which combines a signal modelling network and an outlier detection network. The signal network learns the parameters of our Gaussian output distribution as a function of its inputs for each observation. Meanwhile the outlier network learns the probability of an observation being an outlier as a function of site ID and time (based on the likelihood of the observation having been generated by the Uniform distribution rather than the Gaussian). Architecture extends from Kirkwood et al. (2022)

be recognised. Unlike more traditional geostatistical approaches, which might offer the model pre-defined derivatives from terrain analysis as input features, our deep learning approach allows these derivatives to be learned optimally for the task at hand via trainable convolutional filtering of raw terrain elevation grids (Behrens et al., 2018; Padarian, Minasny and McBratney, 2019; Wadoux, 2019; Kirkwood, 2020; Kirkwood et al., 2022).

For its location input (input B in Figure 6.3) our signal network receives easting, northing, and elevation as spatial location information (all in metres), and continuous time and time of day as temporal location information (in minutes). To provide a cyclic representation of time of day to the network (to aid learning of the diurnal cycle), we transform our minute-of-the-day variable into position on a circle defined by the two dimensions $\sin(2\pi t/T)$ and $\cos(2\pi t/T)$ where t is the specific minute of the day and $T = 1440$: the total number of minutes in a day. It is important that our signal network is able to generalise well to unobserved locations, and so over-fit must be avoided. In aid of this, and in line with the Bayesian interpretation

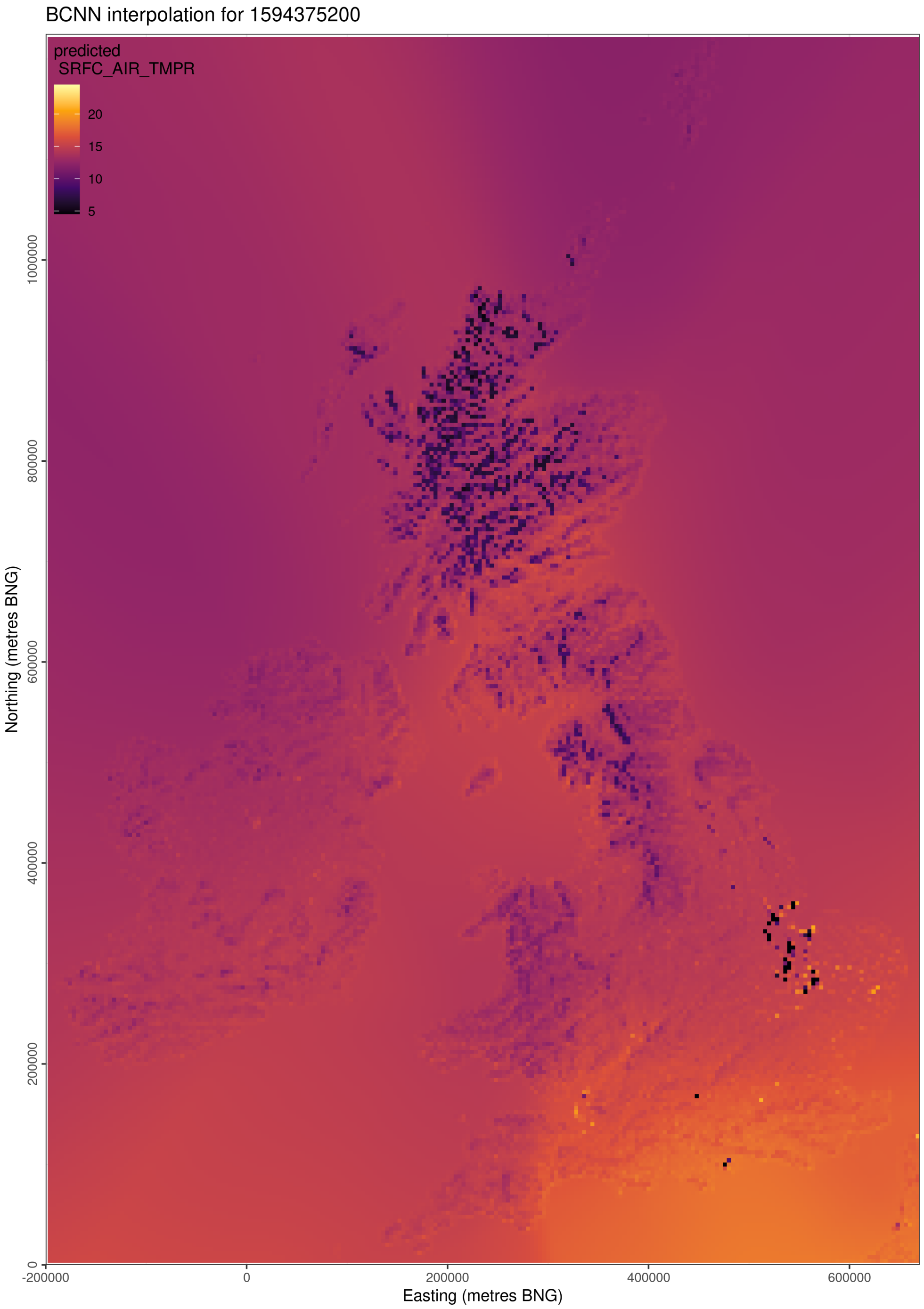


Figure 6.4: Snapshot map of mean surface air temperature produced during preliminary modelling by our signal network only (before the development out of our mixture-model and outlier detection network). Erroneously high and low temperature measurements contaminate the map with dark and bright spots respectively, and are clearly a problem in need of being corrected, which our final architecture (Figure 6.3) achieves. This map is lower resolution than our final model outputs due to being part of the preliminary investigation phase.

of our model — which we discuss in the next section — we run our signal network with a dropout rate of 0.5 on all hidden layers (or spatial dropout in the case of convolutional layers).

In contrast, generalisation is not a concern for our outlier network (Figure 6.3), whose sole task is to model the probability that each training observation is an outlier. To make this task as simple as possible, we provide the outlier network directly with one-hot encoded site IDs, as well as continuous time, such that the outlier network provides outlier probabilities as a linear function of site ID plus a (site-tailored) non-linear function of continuous time (eq. 6.10) facilitated by passing continuous time through a single hidden layer.

$$\text{logit}(\theta_{s,t}) = f_{outliernetwork}(x_{site,time}) \approx \beta_{SITE_s} + f_{SITE_s}(t) \tag{6.10}$$

For full layer-by-layer details of our neural network architecture, we encourage readers to view our code at `https://github.com/charliekirkwood/deepoutliers`.

Note that if we use only the signal network and omit the outlier network and mixture-model architecture (of Figure 6.3) then, as could be expected, the resultant interpolations are contaminated by the outliers. See Figure 6.4 for an example of what this looks like (and why we'd like to avoid it).

6.3.4 Bayesian inference

With the parameters and architecture of our model established, we would like to use the Bayesian framework to learn a posterior distribution for all trainable parameters given the data, D, on which we will train the model. The parameters that control the probability distribution of temperature are $\theta_{s,t}$, $\mu_{s,t}$ and $\sigma^2_{s,t}$ but these are of course themselves functions of the weights within the entire neural network, which we collectively refer to as w. By Bayes' rule we can obtain this posterior distribution over the weights given the data as

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \tag{6.11}$$

So that $p(w|D)$ is proportional to the likelihood of the data given the weights, $p(D|w)$, multiplied by our prior distribution over the weights, $p(w)$. Assuming independence of temperature values given $\theta_{s,t}$, $\mu_{s,t}$ and $\sigma^2_{s,t}$, the likelihood of our mixture model is:

$$p(D|w) = \prod_s \prod_t p(y_{s,t}) \tag{6.12}$$

where $p(y_{s,t})$ is given in equation (6.1).

Here, we adopt a prior distribution for w by utilising Monte Carlo Dropout as suggested by Gal and Ghahramani, 2016. The prior is defined by assuming that a particular “fixed” weight β_j in the network can be randomly “dropped out”, by introducing a set of Bernoulli random variables $B_j \sim \text{Bern}(\pi)$. An individual weight w_j is then defined as

$$w_j = B_j \beta_j \tag{6.13}$$

so that $w_j = \beta$ with probability π and $w_j = 0$ with probability $1 - \pi$. The fixed weights β are learned by stochastic gradient descent during training, whereas the dropout rate π is considered a hyper-parameter of the network and is fixed *a priori*. Equation (6.13) means that the weights w_j are probabilistic in nature so that stochastic forward passes can be used in a Monte Carlo setting to provide an approximate posterior distribution $p(w|D)$ for w.

The particular setup assumes that π is fixed *a priori*, preferably by tuning it. It is however possible that this is automatically estimated using ‘Concrete Dropout’ (Gal, Hron and Kendall, 2017), or by exploring the number of other approaches to Bayesian inference in neural networks that have been proposed (e.g. MacKay, 1995; Graves, 2011; Neal, 2012; Heek and Kalchbrenner, 2019). At present, Bayesian inference in deep neural networks, with their extreme dimensionality (a modest 696 114 trainable parameters in our case), remains a challenge and an ongoing topic of research.

After obtaining the posterior distribution $p(w|D)$ (the practicalities of which we discuss in the next section), we are in a position to compute the posterior predictive distribution for any point in space and time. To obtain robust predictions of the phenomenon of interest, we can set $\theta_{s,t} = 1$ (i.e. exclude the uniform distribution component) and thus generate predictions exclusively from the 'true' Gaussian distribution (6.2). Specifically, we can obtain samples from the posterior predictive distribution of any $y_{s,t}$ (both observed and not):

$$p(y_{s,t}|D) = \int_{w} p_{signal}(y_{s,t}|w)p(w|D)dw, \tag{6.14}$$

6.3.5 Practical setup

We use Weather Observation Website surface air temperature observations from a fourteen day period from 26/10/2020 to 09/11/2020 for this study. This period was selected for containing interesting weather patterns (as evident even in the simple time series of observations; Figure 6.2), including storm Aiden which passed over on the UK on the 31st of October 2020. We randomly subsampled the observations from this period to a single observation per site per hour (where available), which provides 417141 observations in total. We then split this dataset by site ID into 10 folds of approximately equal unique number of unique sites (about 145 unique sites per fold). We split our folds in this site-wise manner in order to assess the fit of the model at sites unseen during training, and therefore to assess the ability of the model to interpolate to new spatial locations throughout the period of observed time.

We assigned data folds one to eight to be used for training, with fold nine providing an evaluation set for hyper-parameter tuning, and fold ten providing a held out test set for assessing the performance of the final trained model at locations unseen by the model. Running on a single GPU workstation (with Nvidia GTX 3070) our neural network trains at one epoch every 3 seconds, so that training for 600 epochs takes about 30 minutes.

6.4 Results and discussion

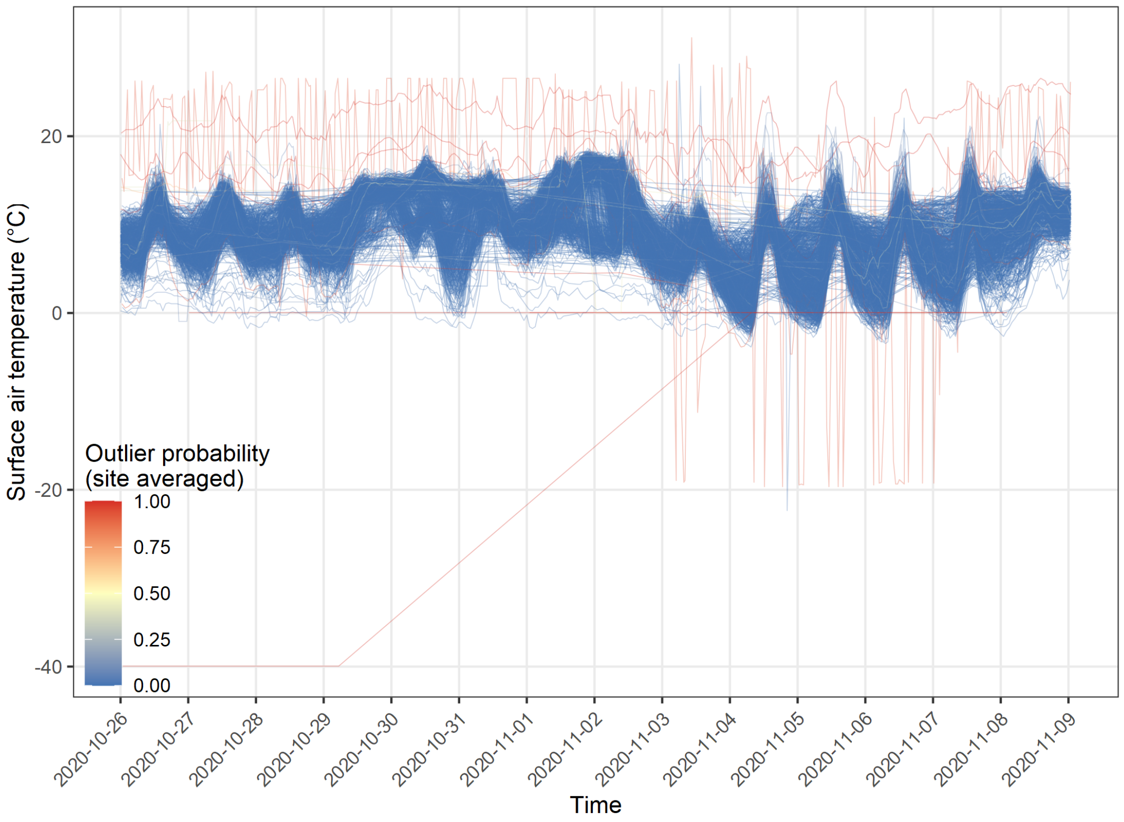


Figure 6.5: Time-series visualisation of our training dataset, coloured by posterior probability that observations would more likely be generated by the uniform outlier distribution rather than the Gaussian signal distribution (for the purposes of the figure, probabilities are averaged by site in order to obtain a fixed line colour per site). We proceed to make predictions using only the Gaussian output of our deep mixture model, such that all predictions represent the true, clean temperature field without spurious measurements.

Our approach has not required the manual labelling of outliers in the training data, but we can see from the output of the model — specifically the parameter θ, which controls the mixing of the Gaussian and Uniform distributions — that observations that visually appear to be outliers have been assigned a high probability of being outliers generated by the Uniform distribution (see for example Figure 6.5, in which sites are coloured by the average predicted outlier probability of their observations). On the basis of this qualitative assessment, we have confidence that predictions generated by our neural network's Gaussian output distribution are a clean (outlier free) representation of the true surface air temperature - we also find this to be evident in the clean look of maps generated by the model (using only the Gaussian distribution for prediction), which do not contain the localised bright or dark spots that could be expected if the model had incorrectly fitted to outlier observations. All subsequent reporting of results, and their discussion, is made on the basis of using only the Gaussian distribution for prediction, so that all predictions are 'outlier-filtered'.

In terms of the quantitative performance of the model as assessed on held out test data (from sites unseen by the model during training), we find that our deep learning approach to spatio-temporal interpolation provides a good degree of predictive skill in both a deterministic and probabilistic sense (Figure 6.7). In a deterministic sense (Figure 6.7A), the mean of the predictive distribution provides an R^2 of 0.90 and a root mean square error (RMSE) of 1.15 degrees Celcius. Probabilistically, our model achieves a continuous rank probability score of 0.6 (Figure 6.7B), and the predictive distribution has good calibration, with held out test observations falling within the 95% prediction interval 92.7% of the time. We can see from the quantile-quantile plot (Figure 6.7C) and prediction-interval coverage plot (Figure 6.7D) that the probabilistic calibration of the predictive distribution performs well across the range of predicted quantiles, although we do see a slight under-dispersion in the tails (i.e. beyond a 90% prediction interval). This may be attributable to limitations of our Monte Carlo dropout approach to approximate Bayesian inference, in that our posterior distribution is fundamentally centred about a single optimum, rather than composed of diverse samples from separate local optima as in full Bayesian inference via MCMC sampling methods (or other proposed approximations such as the 'deep ensembles' approach; Lakshminarayanan, Pritzel and Blundell, 2016) which may reduce the diversity and coverage of the posterior. However, as we assess here on held-out test data, this under-dispersion, if present, appears to be minimal and not overly concerning, especially given that some of our test observations are outliers themselves, which means that perfect calibration (using predictions from our Gaussian distribution alone) cannot be expected.

Overall the performance metrics indicate that our deep mixture density network approach to outlier-filtered spatio-temporal interpolation is doing a good job of providing accurate and trustworthy predictions of historic surface air temperatures for locations in space which have not been observed. It provides a statistical hindcast which is likely to be both computationally cheaper and better calibrated than numerical hindcast alternatives. When run over a long duration, our approach should also provide high quality probabilistic climatology estimates at any unobserved location, which may be useful for planning purposes.

Turning to the maps produced by our model, we can see that our deep learning approach produces detailed predictions which take account of surface topography. For any snapshot in time, we can obtain a map of the predicted mean (average value of $\mu_{s,t}$; Figure 6.8), the average aleatoric uncertainty (average value of $\sigma^2_{s,t}$; Figure 6.9), the epistemic uncertainty of the mean (standard deviation of the posterior distribution of $\mu_{s,t}$; Figure 6.10), and the total uncertainty (standard deviation of the posterior predictive distribution; Figure 6.11). Maps of any desired predictive quantiles, or other statistics of the posterior predictive distribution, can also be produced. In all such maps we can see that our deep learning approach produces predictions and predictive uncertainties that are highly spatially specific. In combination with the high quality of probabilistic calibration achieved (e.g. Figure 6.7 this indicates that our model is producing a predictive distribution that is both sharp and well-calibrated - the ideals for probabilistic predictions and forecasts as proposed by Gneiting, Balabdaoui and Raftery (2007).

Additionally, we can sample from the posterior distribution to generate simulated realisations of surface air temperature fields for any snapshot of time within the observed period. We can generate these both with the Gaussian output distribution active in order to achieve samples of the predictive distribution itself (including aleatoric uncertainty; Figure 6.12), or sample from only the posterior distribution of the mean (without independent noise from the Gaussian) in order to view alternative hypotheses for the mean temperature field at a given time (i.e. the epistemic uncertainty; Figure 6.13). These simulated realisations help to convey the uncertainty in the model, by offering different explanations for plausible data generating processes.

To visualise the output of the model through time, rather than purely in space, we can compare samples from the model (again with and without aleatoric uncertainty included) to observations recorded at a held out test site as timeseries (Figure 6.14). As is indicated by the overall model fit and calibration metrics (Figure 6.7, the predictive performance for held out test sites is good - we can see in the timeseries of samples from the model that samples of the mean track the observations quite closely (but do not track noise in the observations) meanwhile, samples from the posterior predictive distribution, with aleatoric uncertainty included, do a good job of covering the distribution of observations, including noise. Both epistemic and aleatoric uncertainty vary through time (and space, as we saw in Figure 6.9 and Figure 6.10). Animations of the model output (perhaps the best way to view spatio-temporal model output) are available to view at `https://github.com/charliekirkwood/animations`.

The role of the model we present here, as a spatio-temporal interpolator of weather data, is similar to the role that would traditionally be filled by numerical hindcasting (e.g. Palmer et al., 2004). This is where the same physics-based numerical weather prediction models used for forecasting are fitted retrospectively to historic weather observations, to provide a ‘best fit’ of historic weather conditions. In order to provide an indication of uncertainty, ensemble hindcasts can also be run, but it is generally the case that numerical weather prediction ensembles are underdispersive in relation to observations (e.g. Gneiting et al., 2005). By providing well-calibrated spatio-temporal interpolations, our deep learning approach may have the potential to provide a probabilistically-superior (and computationally cheaper) alternative to numerical hindcasting, despite our model having no notion of the physical equations that govern atmospheric dynamics (i.e. Navier-Stokes; Kimura, 2002). The level of detail of spatial structure captured by the model will be limited by a combination of the resolution of auxiliary information (gridded terrain elevation data in this case) and the spatial density of observations, but the model remains free to provide predictions for any point in space. The resolution, or spatial precision, of our proposed approach can naturally improve as the density of observations, and the resolution of auxiliary information, increases.

Our deep spatio-temporal model could also have a role to play in forecast analysis; the problem of determining the best initial conditions for numerical weather prediction (Lorenc, 1986) and of representing the uncertainty in these initial conditions (Bauer, Thorpe and Brunet, 2015b) so that ensemble members may be initialised from them (e.g., Figure 6.6). However, whereas analyses for numerical weather prediction also incorporate information from the numerical model itself (both in terms of the model's structure; with parameters being estimated on a discrete grid and representing known physics, and also using recent forecasts as a source of prior information for the current state in data assimilation e.g. Bengtsson et al. (1982) and Benjamin et al. (2019)) our spatio-temporal deep neural network is non-physical and could instead be viewed as an evolution of traditional kriging approaches for analysis of weather observations (Lorenc, 1986) but which can

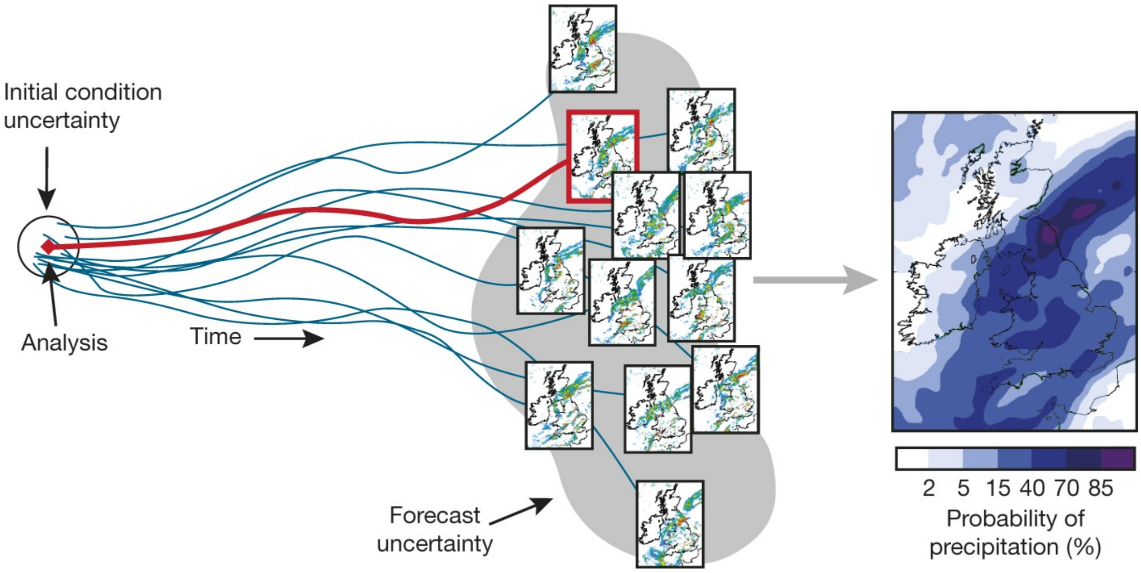

Initial condition uncertainty
Analysis
Time
Forecast uncertainty
2 5 15 40 70 85
Probability of precipitation (%)

Figure 6.6: Schematic diagram of 36-h ensemble forecasts used to estimate the probability of precipitation over the UK, which shows the role of the analysis in estimating initial condition uncertainty from which the numerical ensemble members are initialised. Taken from Bauer, Thorpe and Brunet (2015b) where it was provided by K. Mylne of the UK Met Office.

more flexibly incorporate information from terrain, as well as filter out outliers (in the case of our mixture model). Further work will be necessary to establish whether our approach can bring improvements to existing methods of analysis.

It is interesting to observe the difference between samples of our model's posterior distribution both with and without aleatoric uncertainty — the independent noise provided by the Gaussian output distribution — included (e.g. by comparing the top and bottom of Figure 6.14, or comparing Figure 6.12 with Figure 6.13). As can be seen in Figure 6.14, the independent noise of our Gaussian output distribution is required in order to provide well-calibrated coverage in relation to observations (at least in our setup, in which independent noise is a part of the model). Without this aleatoric uncertainty included, the distribution over our plausible mean functions would be underdispersive in relation to the observations. This has parallels in the setup of numerical weather prediction and hindcasting (e.g. Rawlins et al., 2007; Rougier and Beven, 2013; Bauer, Thorpe and Brunet, 2015b), in which ensembles tend to be underdispersive in part for the same reason: that while these numerical ensemble members do capture epistemic uncertainty in initial conditions Rougier, 2013 and perhaps across model parameters Leutbecher and Palmer, 2008, they tend not to model aleatoric uncertainty. In order to achieve well-calibrated numerical weather forecasts, statistical-post processing must therefore be used, such as Bayesian model averaging in which individual ensemble members are 'dressed' with suitably scaled Gaussian noise (Raftery et al., 2005), thus effectively transforming the underdispersed ensemble at the bottom of Figure 6.14 to the well-calibrated ensemble at the top of Figure 6.14. It is perhaps another strength of our Bayesian deep learning approach that 'ensemble' predictions of both forms (with and without aleatoric uncertainty) can be generated equally easily by sampling from the same model, and that our full posterior predictive distribution (which includes aleatoric uncertainty) is innately well-calibrated and requires no subsequent post-processing.

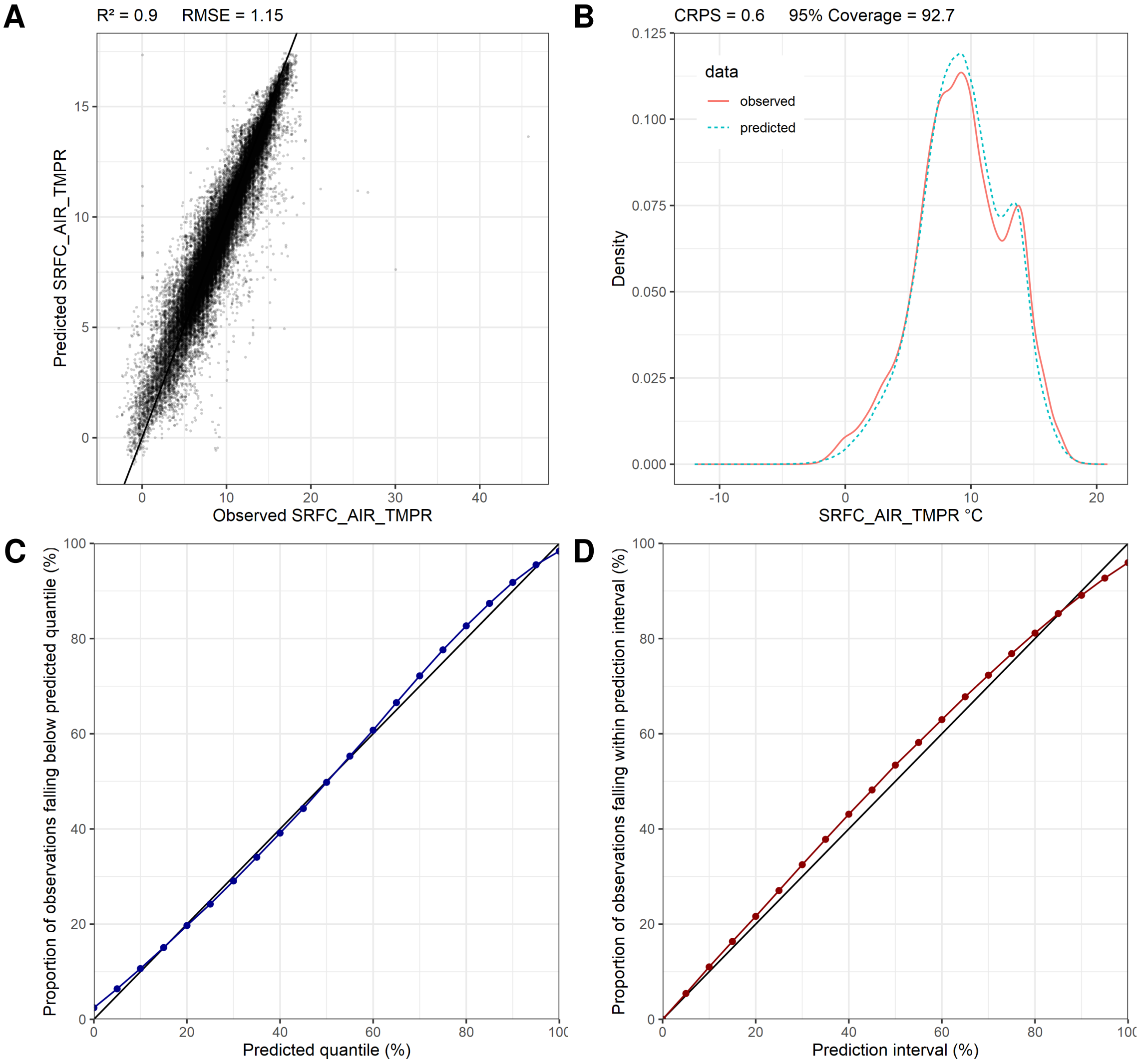


Figure 6.7: **A**, Deterministic comparison of observed and predicted values of surface air temperature, taking the mean of the predictive distribution as the prediction. Note some outlier observations are present in the test set. **B**, Probabilistic comparison of observed and predicted distributions of surface air temperature, taking 50 samples from the predictive distribution for each observation. **C**, Q-Q plot and **D**, prediction interval coverage plot to check the calibration of our model's predictive distribution against test observations. All use data from the held out test set (n = 41836), taken from sites kept unseen until after hyper-parameter tuning and model training.

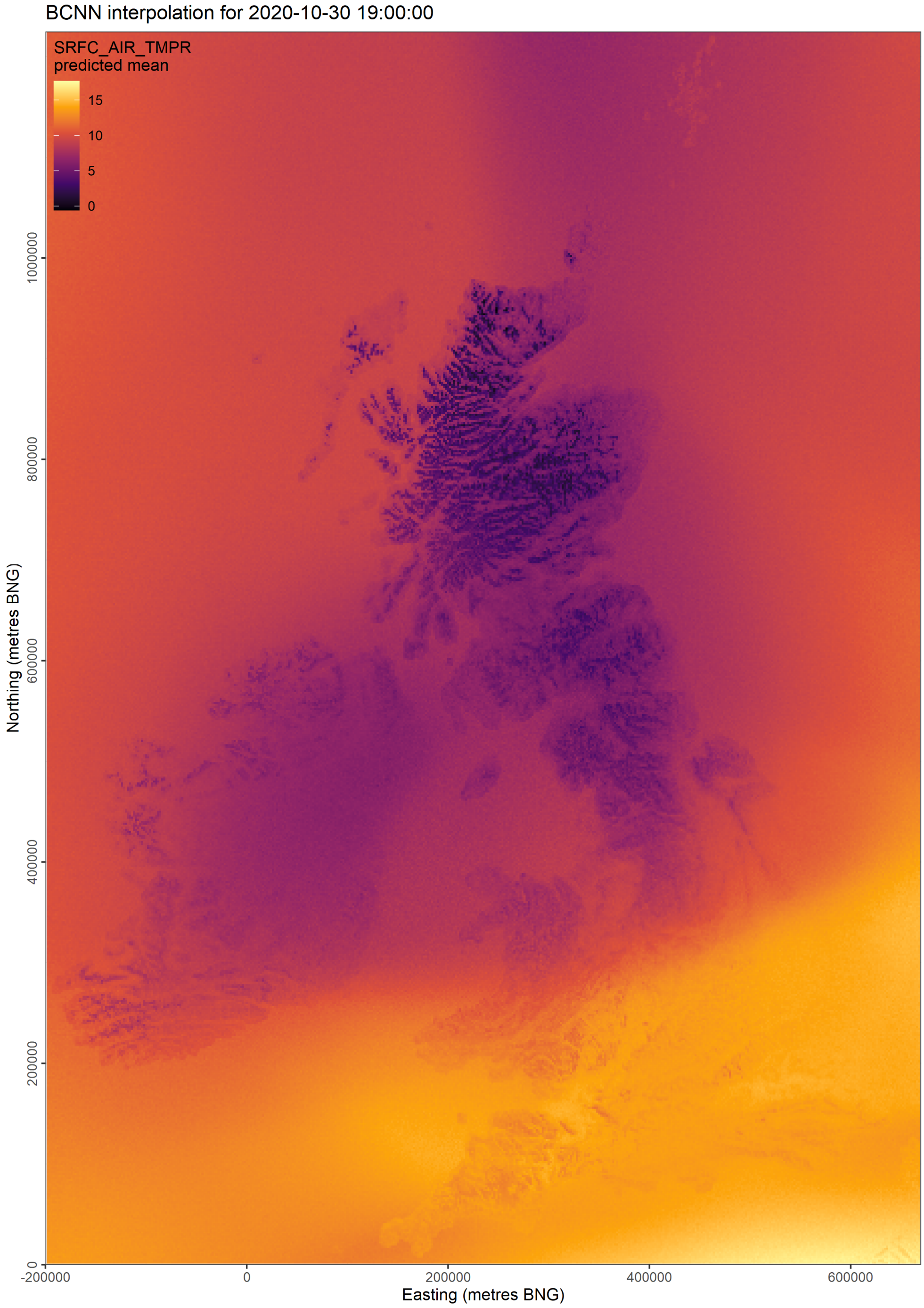


Figure 6.8: Mean surface air temperature map for a single snapshot in time, as predicted by our deep neural network. The use of convolutional layers in our neural network architecture allows our predictions to be informed by patterns relating air temperature to terrain features, and in doing so produce detailed spatio-temporal fields.

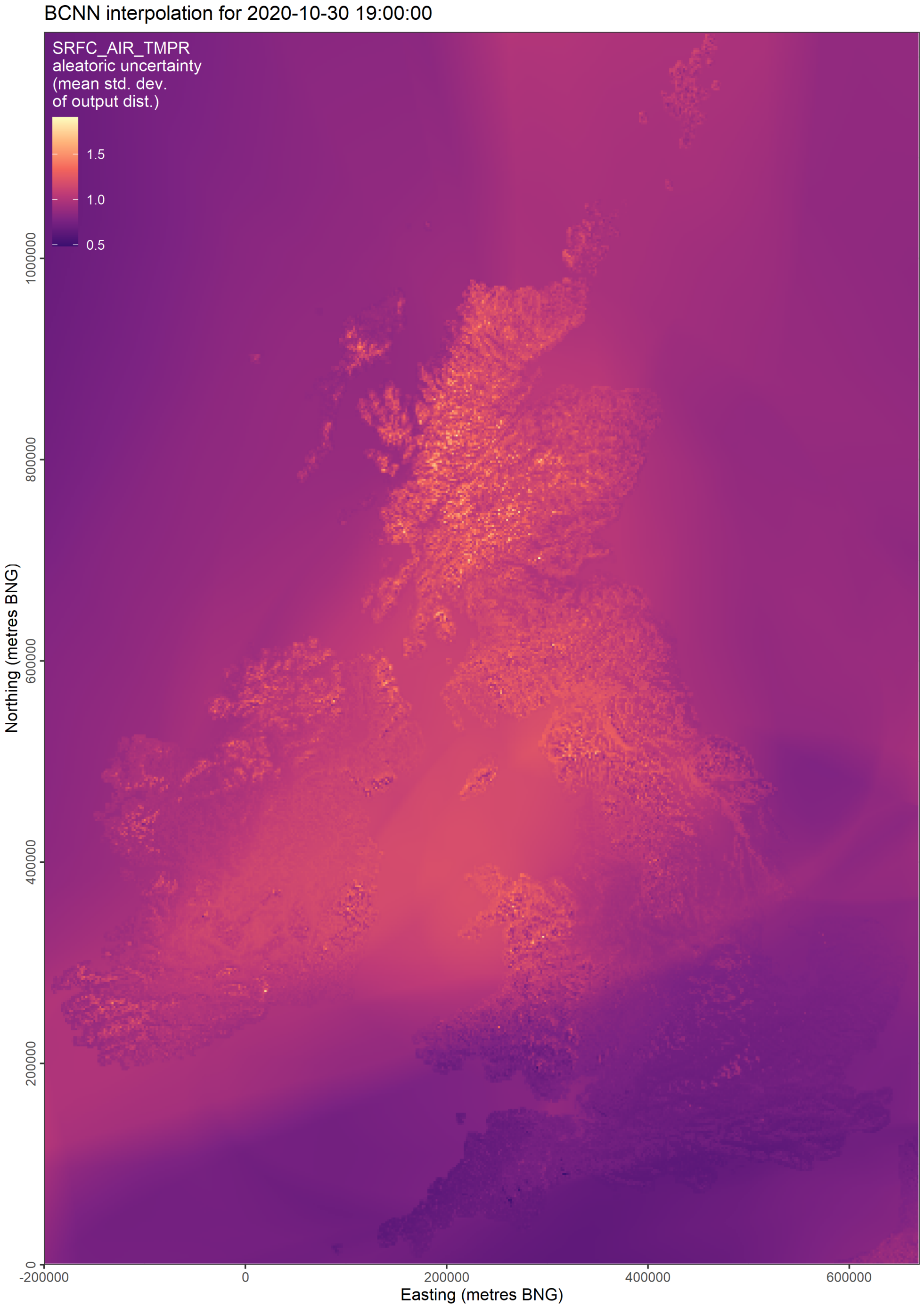


Figure 6.9: Aleatoric uncertainty (mean standard deviation of the Gaussian output distribution, ℃) at the same snapshot in time.

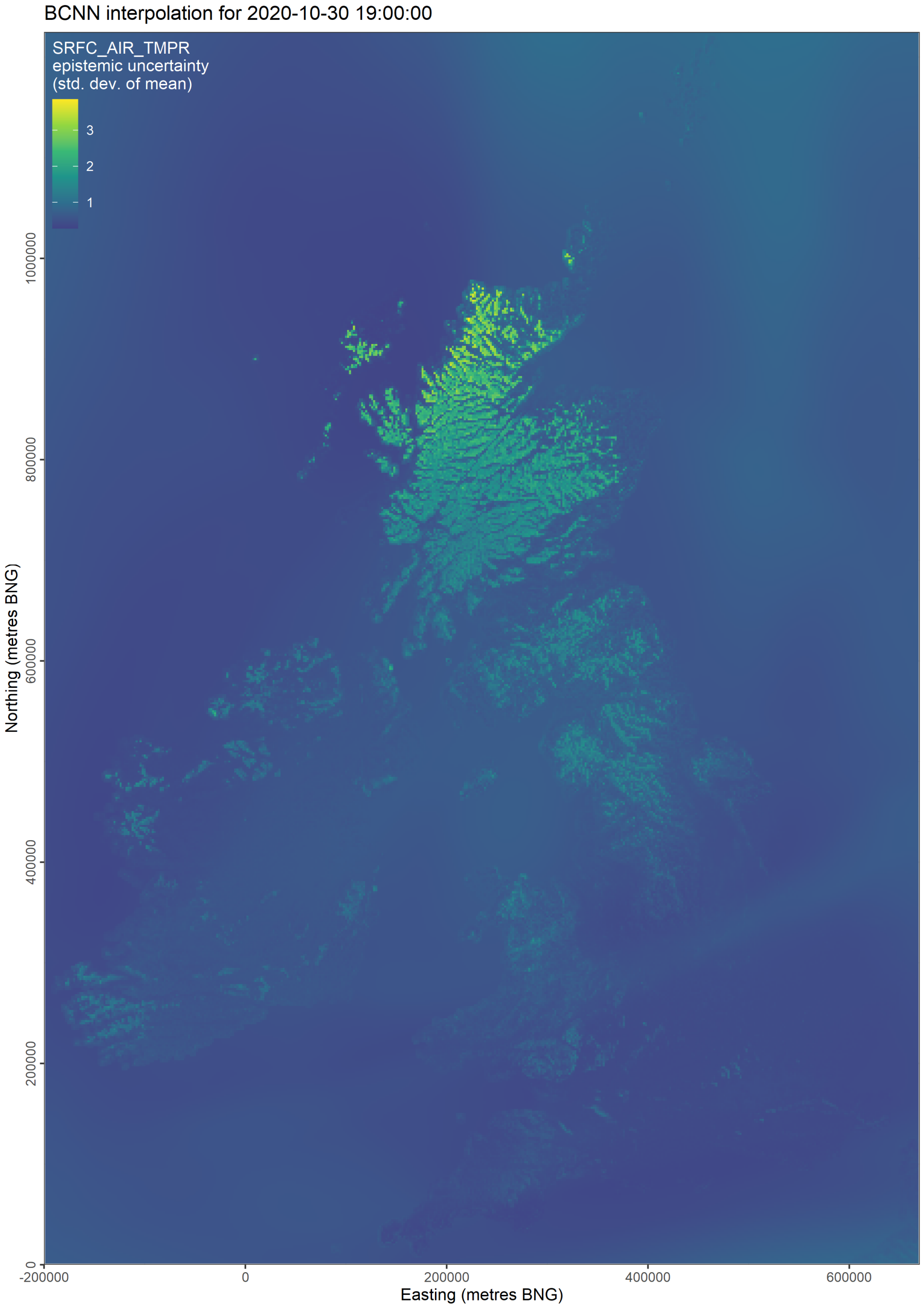


Figure 6.10: Epistemic uncertainty (standard deviation of the mean, ℃) at the same snapshot in time.

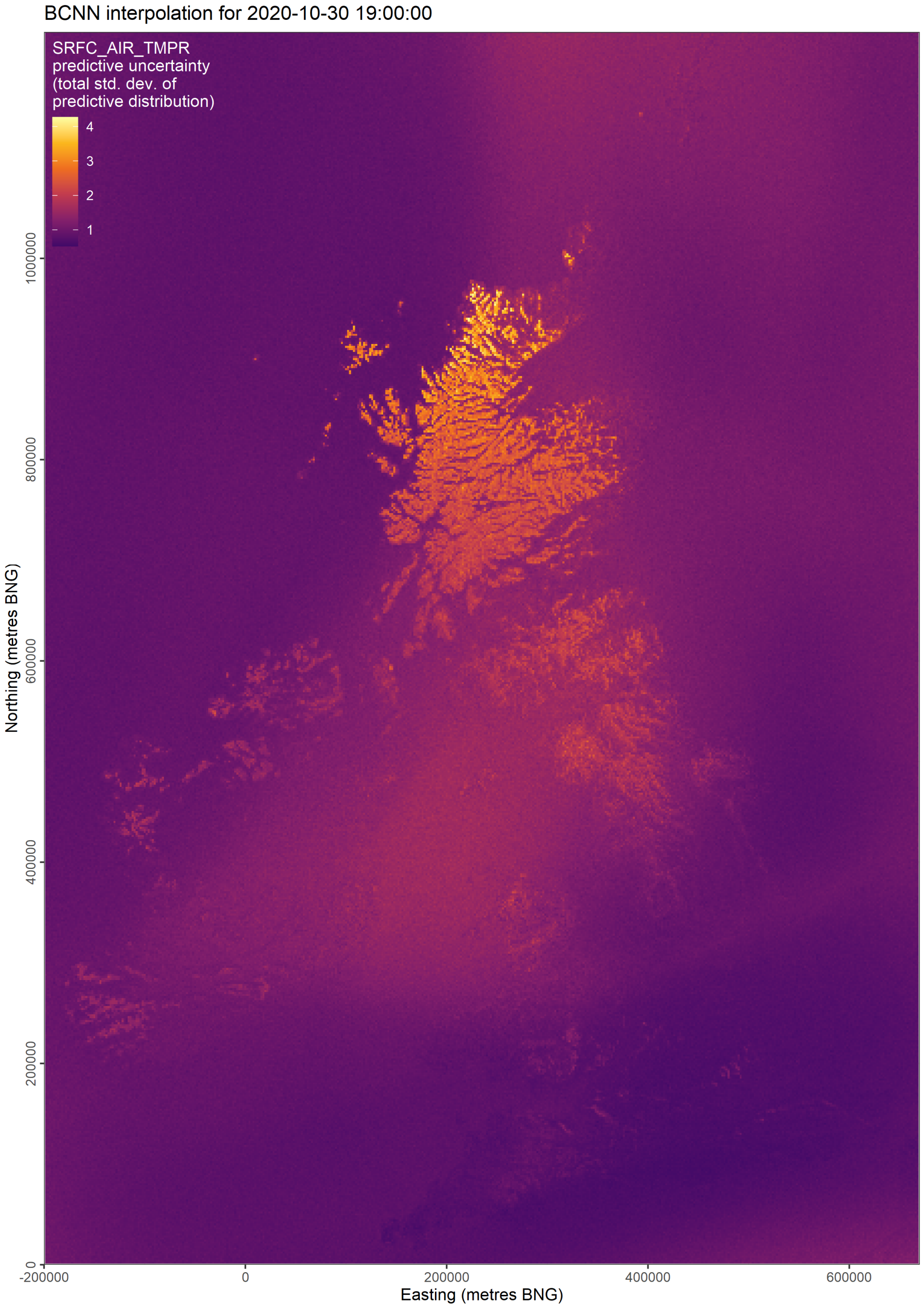


Figure 6.11: Total uncertainty (standard deviation, ℃) of the predictive distribution at the same snapshot in time.

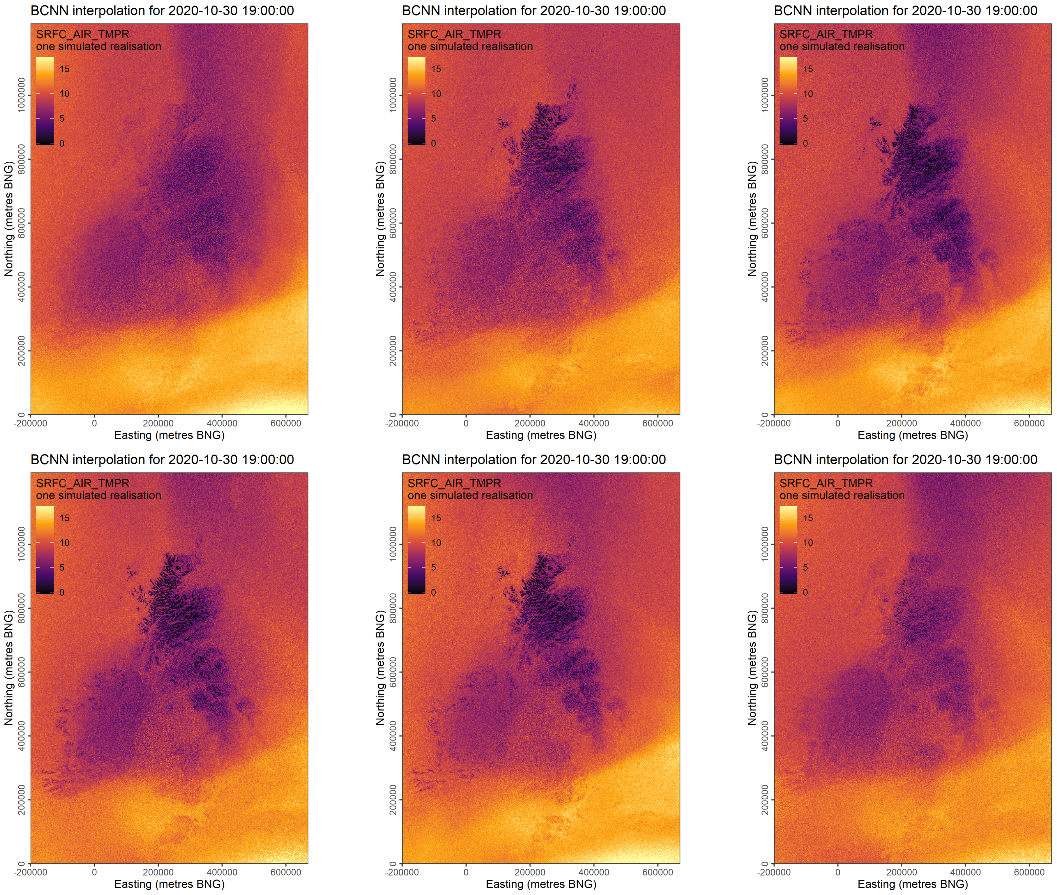

BCNN interpolation for 2020-10-30 19:00:00
SRFC_AIR_TMPR
one simulated realisation
15
10
5
0
Northing (metres BNG)
1000000
800000
600000
400000
200000
0
-200000
0
200000
400000
600000
Easting (metres BNG)
BCNN interpolation for 2020-10-30 19:00:00
SRFC_AIR_TMPR
one simulated realisation
Easting (metres BNG)
Northing (metres BNG)
BCNN interpolation for 2020-10-30 19:00:00
SRFC_AIR_TMPR
one simulated realisation
Easting (metres BNG)
Northing (metres BNG)
BCNN interpolation for 2020-10-30 19:00:00
SRFC_AIR_TMPR
one simulated realisation
Easting (metres BNG)
Northing (metres BNG)
BCNN interpolation for 2020-10-30 19:00:00
SRFC_AIR_TMPR
one simulated realisation
Easting (metres BNG)
Northing (metres BNG)
BCNN interpolation for 2020-10-30 19:00:00
SRFC_AIR_TMPR
one simulated realisation
Easting (metres BNG)
Northing (metres BNG)

Figure 6.12: Six samples, or simulated realisations, from the posterior predictive distribution at 19:00 on 30/10/2020. As a collective (in the limit of infinite samples) such samples represent the total uncertainty in our spatio-temporal interpolation of surface air temperatures.

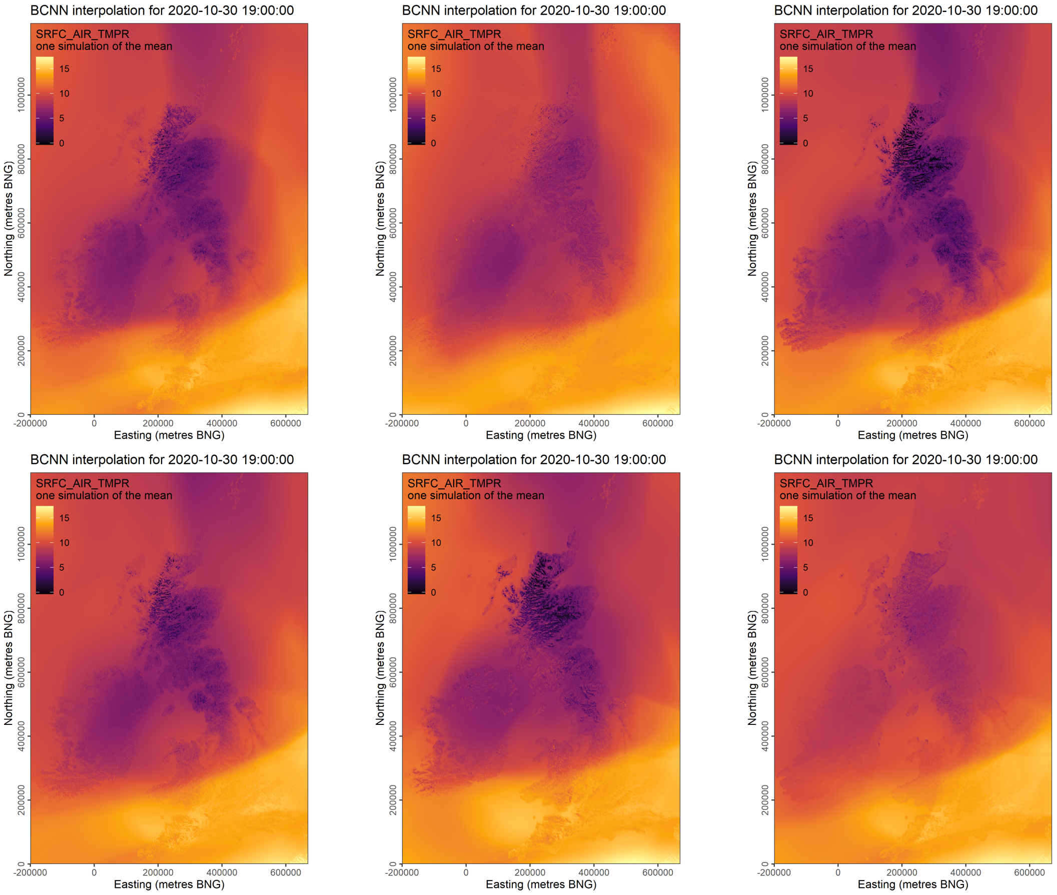


Figure 6.13: Six samples from the posterior distribution of the mean at 19:00 on 30/10/2020. As a collective (in the limit of infinite samples) such samples represent the epistemic uncertainty in our spatio-temporal interpolation of surface air temperatures.

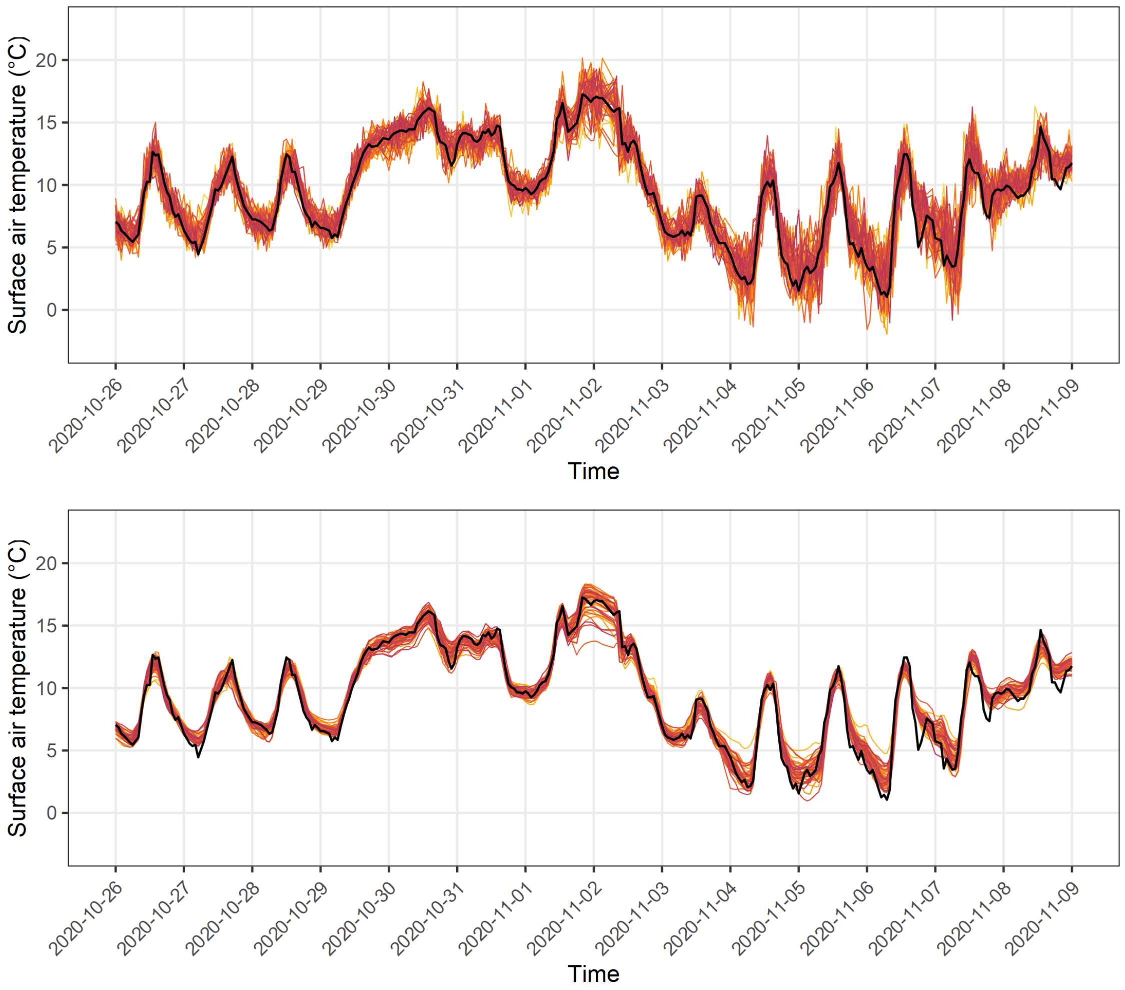


Figure 6.14: Time series of samples generated from the trained model, showing total uncertainty (top, epistemic and aleatoric uncertainty), and uncertainty in just the predicted mean (bottom, epistemic uncertainty only) for predictions on a held-out test site.

6.5 Conclusions

We have presented a deep learning approach that provides well-calibrated outlier-corrected spatio-temporal interpolation of crowd-sourced weather observations. Our deep mixture density network approach to outlier classification unifies outlier detection and correction as part of a same single probabilistic data modelling process, which provides a more streamlined modelling and model-checking workflow compared to alternative two stage techniques (in which outlier detection and filtering is performed separately prior to data modelling).

Our unified approach allows us to, through a single probabilistic data model (our Bayesian deep neural network), generate high fidelity spatio-temporal predictions from historic crowd-sourced weather observations. The ultimate functionality is therefore similar to that of numerical hindcasting or reanalysis, but our approach is likely to be computationally cheaper and our predictions are innately well-calibrated, requiring no post-processing. By providing a full predictive distribution, the uncertainty of predictions is fully quantified, therefore making our output useful to decision makers. The predictive uncertainty can also be viewed and mapped as its two separate components: aleatoric uncertainty, or irreducible uncertainty in the data, and epistemic uncertainty, or reducible uncertainty about our state of knowledge. In addition our predictions can be provided at any point in space and time, therefore catering for hyper-local scales, and so may be viewed as satisfying the requirements of a ‘models of everywhere’ approach to harnessing Internet of Things (IOT) type weather observations. On the basis of all these benefits we therefore consider our approach to have potentially powerful applications for quality control, data assimilation and climatological studies that maximise the utility of IOT data for environmental modelling applications in an increasingly data-rich world.

Compared to ‘standard’ Bayesian hierarchical modelling using either Marcov Chain Monte Carlo (MCMC) or Integrated Nested Laplace Approximation (INLA) methods of inference (Blangiardo et al., 2013; Bakar and Sahu, 2015), our deep neural network approach (using Monte Carlo dropout for inference) brings the potential to benefit from computer vision by learning new features through which the target variable relates to the landscape. The Monte Carlo dropout approach is also very computationally efficient and scalable, and thus suitable for modelling very large datasets. On the other hand, ‘standard’ Bayesian Hierarchical Modelling provides a precise framework in which to specify prior beliefs mathematically (as opposed to the more ‘hand-wavy’ approach of ‘specifiying’ prior beliefs in the form of neural network architectures). As a result, Bayesian hierarchical models are perhaps in general the more sensible option where the importance of prior belief is particularly important; i.e., for modelling problems with a relatively small number of observations. As the number of observations increases however, the exactness with which we specify our prior beliefs, and of our Bayesian inference methods, becomes arguably less important, and methods that are suited to learning potentially new relationships from very large datasets become increasingly sensible. Our deep Bayesian spatio-temporal model occupies a position closer to this ‘data rich, prior knowledge poor’ end of the spectrum. It is also worth noting that even for systems that we do believe we understand apriori - such as atmospheric dynamics - deep learning approaches are beginning to be demonstrated which do outperform our existing numerical weather prediction systems (e.g., Bi et al., 2022) which provides some evidence that the importance of exactly specifying our prior beliefs (which may themselves be incorrect, either directly or due to a computational need to make approximations) is perhaps diminishing as the amount of available data increases.

6.6 Code and data

The code to reproduce this study is available at `https://github.com/charliekirkwood/wowpaper` and includes functions to download NASA’s SRTM elevation data via the raster package in R. Data from the Met Office’s Weather Observation Website can be downloaded from `https://wow.metoffice.gov.uk/`

Animations of the deep mixture density network’s spatio-temporal interpolations are also available as .mp4 files in the repository `https://github.com/charliekirkwood/wowpaper`

Chapter 7

Geological mapping in the age of artificial intelligence

While weather forecasting has been a mathematical practice since the first half of the 20th century, modern geological mapping practice typically continues to rely on mental modelling and hand-drawing. In many ways the two disciplines are ‘cousins’ in that both aim to provide information about the current (and future) state of their respective spheres; the atmosphere and the lithosphere.

The fundamental advantages of probabilistic modelling apply equally to both disciplines: probabilistic predictions and forecasts allow us to make decisions which account for uncertainty and therefore (hopefully) avoid losses due to unforeseen occurrences. On the surface it might seem that weather forecasts have higher uncertainty, but this is dependent on their lead time: while our uncertainty about the future state of the atmosphere increases with lead time, we can actually be quite certain about the current state of the atmosphere due to the comprehensive range of sensing technologies by which we observe it (i.e., satellites, radar, weather stations). On the other hand, our knowledge of even the current state of the lithosphere is quite uncertain, due to the comparative difficulty of making lithological observations, but also due to the comparative simplicity of hand-drawn geological mapping practices relative to atmospheric modelling practices.

If artificial intelligence has the potential to improve probabilistic weather forecasting — as the evidence increasingly demonstrates — then even more-so it has the potential to not just improve, but to revolutionise geological mapping, which has not undergone the same ‘quiet revolution’ in Numerical Weather Prediction that weather forecasting has benefited from over the decades. Instead, when it comes to geological mapping the accepted ‘benchmark’ remains mental modelling and hand-drawn mapping practice.

This chapter is an exploration of the viability of the geostatistical Bayesian deep learning methodology conceived in chapters 4, 5, and 6 to tackle the task of geological mapping. It is intentionally written in accessible language, and has appeared in Geoscientist magazine — the magazine of the Geological Society of London — as a feature article.

7.1 Introduction

Our technology has progressed immensely in two centuries. It was back in 1804 that the world's first steam locomotive, invented by Cornish engineer Richard Trevithick, hauled 10 tonnes of iron ore and 70 passengers on a nine-mile journey from a Merthyr Tydfil ironworks to Abercynon in the valleys of South Wales. This first steam train barely exceeded walking speed, but by demonstrating machinery that could bypass the relevant limitations of horses, Trevithick revolutionised our approach to transport (though of course, the changes did not all happen overnight).

It was during this time of industrial revolution, interspersed with riots by luddites who understandably sought a fairer share of the benefits of new machinery, that William Smith published the first geological map of Britain in 1815. This was followed in 1820 by the map of George Bellas Greenough, who was the president of the Geological Society of London at the time. These two pioneers had disagreements in their ideas for the best mapping approach: Smith used fossils to recognise and map different strata, while Greenough is said to have favoured 'mineralogical views.' Such is the nature of science. Despite their considerable age, the maps of both Smith (Figure 7.1) and Greenough (Figure 7.2) still look entirely familiar to us today – bodies of similar rock are classified as distinct units and mapped as coloured polygons (or as three-dimensional volumes using cross sections). Is this all that geological mapping ever needed to be?

7.1.1 Limitations of the traditional approach

It could be said that the production of a general-purpose geological map, with the aim to provide an overall summary of geological information rather than to support a specific task in particular (such as planning for construction, mineral extraction, or geohazard mitigation) is as much an artistic endeavour as a scientific one, with no right or wrong answer. Smith's map may emphasise biostratigraphy, while Greenough's may highlight mineralogy, but without a particular purpose for the map in mind, can we really say that one is better than the other?

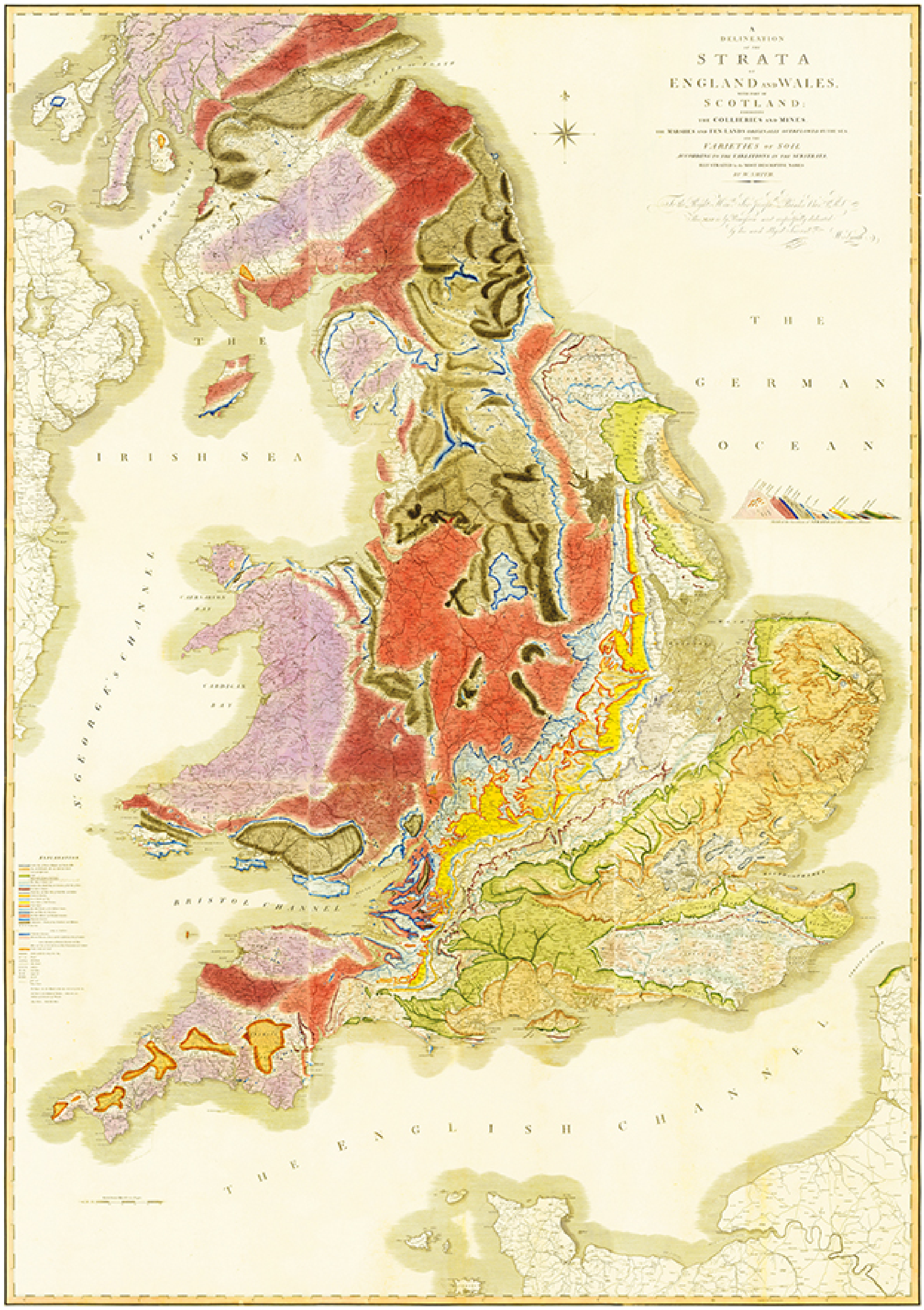


Figure 7.1: William Smith's original geological map of Britain published in 1815.

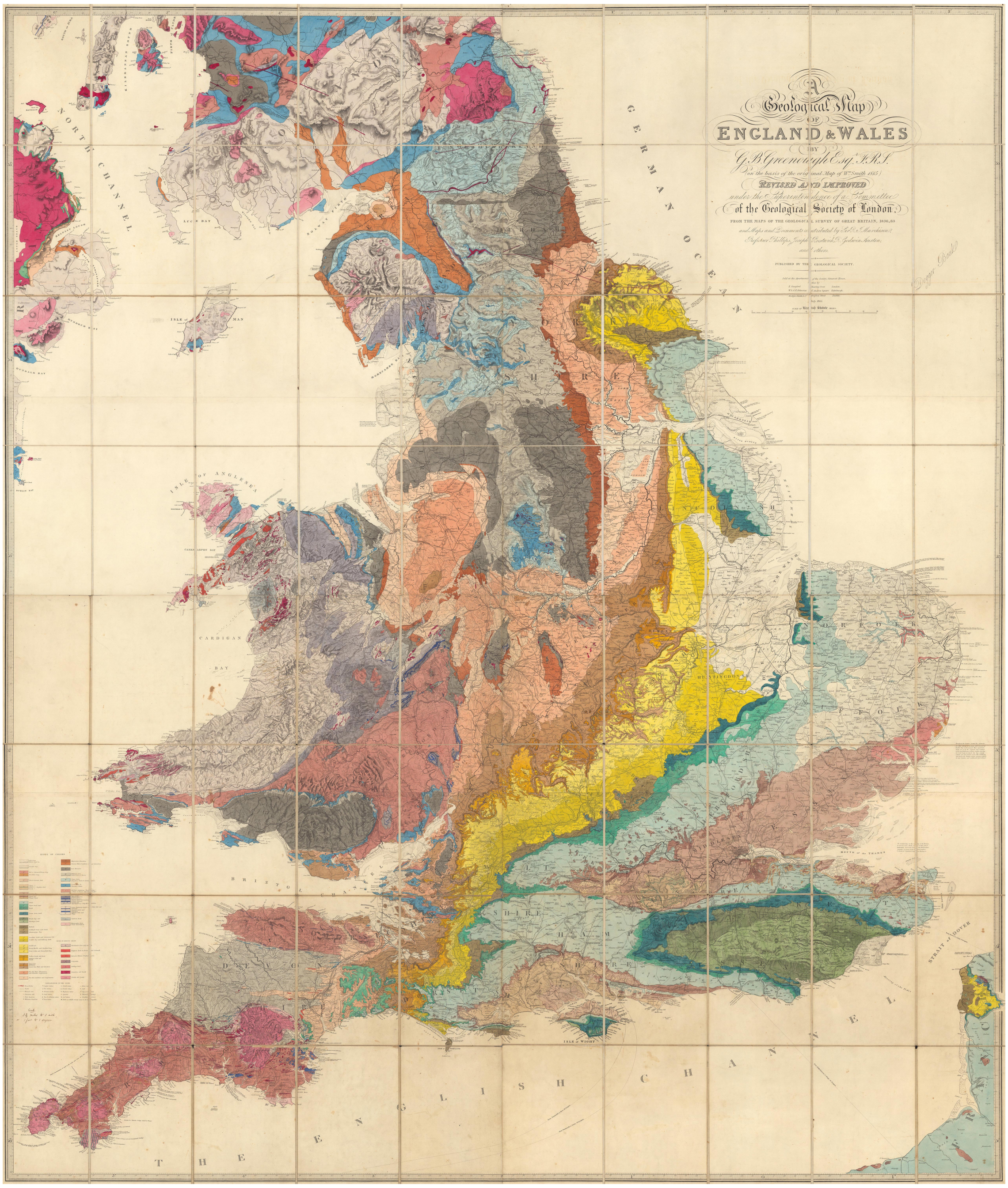

A Physical and Geological Map of England & Wales
by G.B. Greenough Esqr. F.R.S.
(on the basis of the original Map of Wm. Smith 1815)
Revised and Improved
under the Superintendence of a Committee
of the Geological Society of London,
from the maps of the Geological Survey of Great Britain, 1836–63
and Maps and Documents contributed by Sir R. I. Murchison,
Professor Phillips, Joseph Prestwich, R. Godwin Austen,
and others.
Published by the Geological Society.
German Ocean
North Channel
Isle of Man
Isle of Anglesea
Cardigan Bay
Bristol Channel
The English Channel
Isle of Wight
Strait of Dover
Dogger Bank

Figure 7.2: George Bellas Greenough's geological map of Britain published in 1820.

Ideally, we would assess the quality of any map by how closely it matches reality (the same philosophy is now behind the development of ‘digital twins’ in many fields). However, in the case of traditional ‘classification first’ geological maps this assessment is not easy to make. That is because, while in reality geological properties vary continuously through space, classified geological maps instead make the assumption that geological properties will be fixed uniformly within the boundaries of each mapped lithological unit – their classes cannot represent internal spatial variability. This means that when it comes to providing geological information for any point in space, the classified geological map can only offer the overall attributes of the unit in which the point sits as a summary, rather than provide precise location specific information.

The geological classification procedure therefore introduces a high degree of spatial imprecision to our maps (it does not matter where within a unit our point of interest lies), as well as reduces the fidelity with which our maps can represent geological properties by limiting the number of possible values that any geological property can take (to the number of unique classes on the map). For these reasons, the traditional ‘classification first’ geological mapping approach muddies the water when it comes to comparing our maps with reality. This is a problem because comparison with reality is the best guide we have in our efforts to improve our knowledge and understanding of the Earth.

In addition to these difficulties with providing high-fidelity location-specific information, another limitation of traditional classified geological maps is that they do not communicate uncertainty. A geological map provides predictions of the geology we can expect to find at all locations across the map. However, not all locations can be observed during the mapping process, and this means that the practice of geological mapping is an exercise in ‘filling in the gaps’ between observations, which may be few and far between. Without infinite observations we will never be certain about the true form of the geology that we are mapping. In light of this inherent and unavoidable uncertainty, the logical solution is to work within a

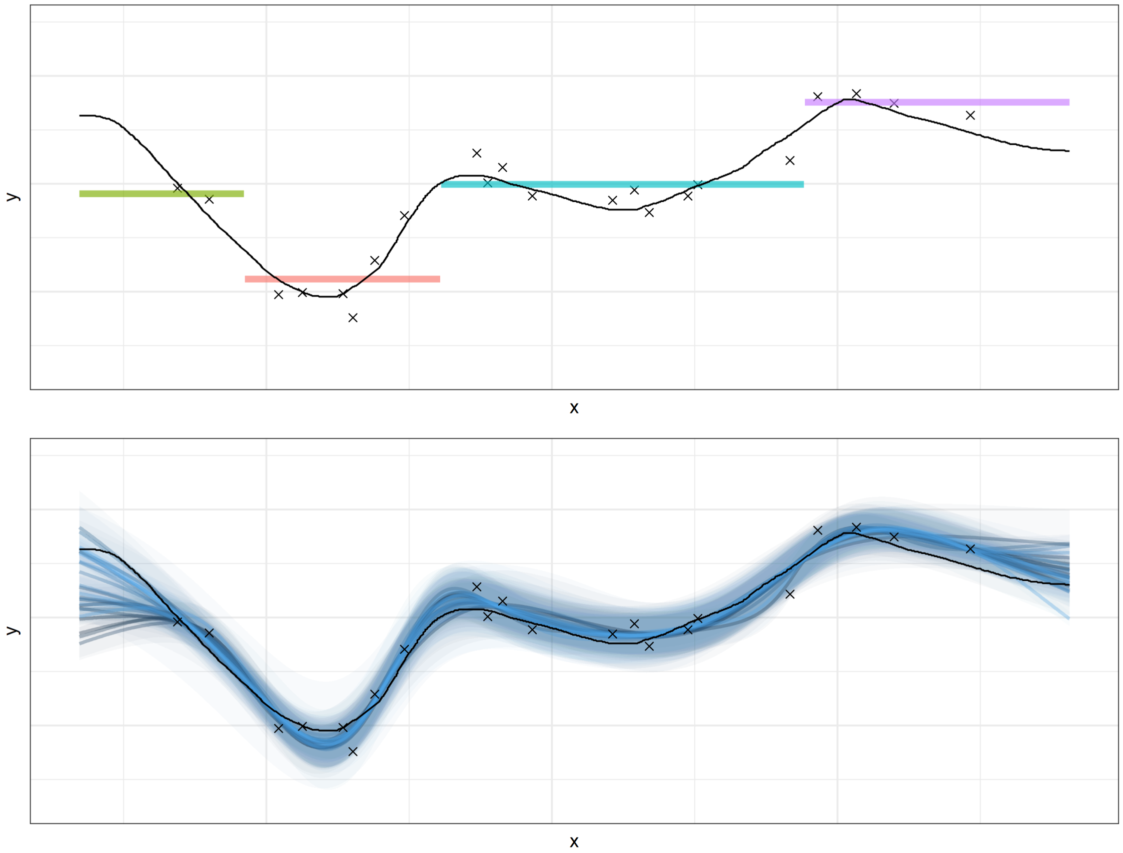


Figure 7.3: Conceptual comparison of 'classification first' mapping (**top**) and 'properties first' mapping using Bayesian AI (**bottom**). The black line shows the true value of some geological property, y, through space, x. Black crosses are the observations of this property, with some error, which the two mapping approaches utilise. Note that regardless of their number, or the exact position of their boundaries, a single set of discrete classes fundamentally fails to represent continuous geological properties, and cannot convey uncertainty. These issues only become more severe in the real-world mapping case of attempting to represent multiple properties at once using the same class boundaries. Conversely, mapping 'properties first' using Bayesian AI methods brings the potential to obtain skilful probabilistic maps of any geological properties of interest.

probabilistic framework and to use Bayesian reasoning to formally acknowledge that “the true map could be this, or it could be that” – and so on ad infinitum – rather than to falsely claim that one single map is definitely ‘the truth’, which is the traditional geological mapping approach. We will discuss more on this later.

These criticisms of the traditional ‘classification first’ approach to geological mapping should not take away from the magnitude of William Smith’s achievement in publishing the first geological map of Britain back in 1815, a full six decades ahead of the invention of the lightbulb (and even ahead of the invention of the first lighter and friction match in 1823 and 1826 respectively). Back then it was reasonable for William Smith to make the simplifying assumption that the variability within different units of rock could largely be ignored during the mapping process, because this made his goal of mapping Britain achievable using the technology of the time (i.e. travelling by horse and drawing on paper). Ignoring within-unit variability meant that ‘all’ William Smith had to do was delineate boundaries between lithological units – although in fact he did also use gradational shading to indicate some within-unit variability, a capability that has unfortunately been lost in the modern digitisation of classified geological maps.

We also cannot blame William Smith for failing to convey the uncertainty associated with his map. He was limited to the medium of paper and had no access to computational power beyond his own brain. By some coincidence however, it was in 1814, just a year before the publication of Smith’s map, that French polymath Pierre Simon Laplace published his Philosophical Essay on Probabilities, which presents what we now consider to be a Bayesian approach of using probability to deal with uncertainty (after work following Thomas Bayes in the 1700s). The Bayesian approach provides the mathematical foundations for quantifying uncertainty – namely Bayes theorem – which remains a cornerstone of modern AI.

Today, with our electricity, trains, cars, planes, computers, satellites, and MRNA vaccines, our technology has progressed enormously. This is fortunate for us, as the issues we face in the 21st century undoubtedly have higher stakes than ever – we now know how interconnected the world is, from lithosphere to atmosphere to cryosphere to biosphere, and how our own imperfect actions are causing a climate crisis. If there was ever a time that humanity would most benefit from having precise high-fidelity knowledge of the Earth's crust, with uncertainties quantified, that time is surely now. And yet, the traditional two-century old 'classification first' approach to geological mapping seems to remain accepted practice. William Smith himself, being the innovator that he was, would likely question why we have not moved on, given all our technological advantages today.

7.1.2 Transition to geological properties mapping

If there's just one change we should be making to our geological mapping procedures, it's to transition from a 'classification first' approach to a 'properties first' approach. From this many improvements can naturally follow. In a 'properties first' approach we would focus on mapping the multitude of properties on which our classifications have traditionally been based (e.g. mineralogy, texture, age, and more), and only then (if necessary) apply a classification scheme afterwards, which can be tailored specifically for the task at hand (e.g., Harff and Davis, 1990).

The key benefit of adopting a 'properties first' approach to mapping is that this allows us to directly assess the quality of our maps (including uncertainty quantification) against observations, without the imprecision of a classification scheme getting in the way. The 'properties first' approach also enables the option of providing task-specific geological information directly as continuous non-classified output, thus avoiding the problem of losing information to discretisation (the process of converting naturally continuous properties into discrete classes) which has always been the case for traditional classified geological maps (Fig 2).

As you may well be thinking, this 'properties first' approach is not necessarily a new idea. For example the mineral exploration industry has been operating ‘properties first’ for decades due to the fact that it is much easier to find metal deposits by observing and mapping the concentrations of those metals directly, rather than trying to infer their distributions on the basis of classified geological maps (although in reality, both approaches are used side by side in order to gain as much task-specific information as possible – but is there a better way?). Work published by South African mining engineer Danie Krige in 1951 laid the foundations for the now ubiquitous geostatistical approach to spatial interpolation known as kriging, the mathematics of which were formalised by French mathematician Georges Matheron in the 1960s. Krige and Matheron could therefore be considered the original masterminds of the properties first mapping approach. Crucially their work allowed geological properties to be mapped probabilistically for the first time, but not without making some perhaps overly-restrictive simplifying assumptions.

By the way, it is worth us acknowledging that mapping geological properties by hand is not practically feasible, if for no other reason than that the geologist would have to be an incredibly good artist to map any continuous geological property using continuous colour (or pencil) shading with any kind of accuracy. This then is another reason for the historic dominance of the ‘classification first’ approach to geological mapping – when working by hand we in general find it much easier to draw lines than to shade with continuous gradation, and so the origins of geological mapping as a hand-drawn practice favoured the delineation of distinct classified units using lines. However, statistical interpolation methods such as kriging and more complex artificial intelligence approaches allow us to bypass these artistic limitations when producing maps of continuous geological properties.

Why is it then, that the ‘properties first’ approach has not already taken over as the standard approach to geological mapping? I believe the reason is that we have been limited by the simplicity of our geostatistical models, or more specifically by the simplifying assumptions that technological limitations have so far been forcing us to make. Again, this is not a criticism of Krige or of Matheron, for they, like Smith, were pioneers of revolutionary ideas which remain foundational to this day

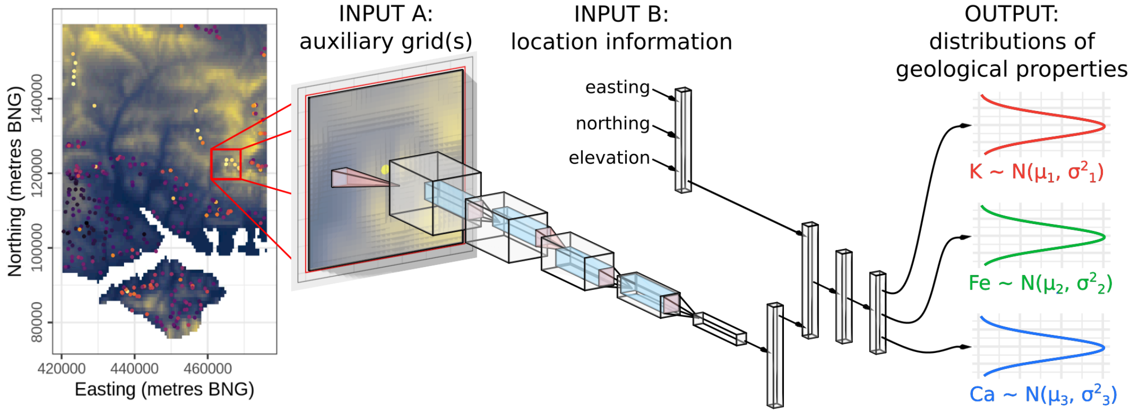


Figure 7.4: Architecture of the deep neural network that generates the RGB (K, Fe, Ca as centred-log ratios) AI map shown in this chapter. The architecture combines parallel chains of information processing in order to learn both local contextual features from gridded terrain elevation data (Input A) and global position features (Input B), as well as the interactions between these two feature types. The 'thought processes' involved are not unlike those of a field geologist. After Kirkwood et al., 2022.

(as an aside, let us try and make it the case that the future pioneers of geoscience can be women just as easily as men).

Ordinary kriging relies on a key assumption that the autocorrelation properties of the observations (how the similarity between observations relates to the distances between them) will not change through space. It usually also assumes that the autocorrelation properties will be the same in all directions. While these assumptions may be acceptable when working on the scale of a single mine, when mapping at regional or national scales any field geologist would agree that geological structure is important, and so maps produced by kriging, which fail to capture this structure, seem unconvincing to us. This is surely a major reason for the fact that maps produced using geostatistical modelling have not already won over the geological mapping community. Instead, 'classification first' maps have retained the limelight because, for all their flaws, their hand drawn linework does allow geological structure to be conveyed. But can we marry the strengths of the two? In order for the 'properties first' approach to geological mapping to become widely adopted, we will require a new generation of geostatistical modelling methods which do away with the need to make assumptions that are geologically unbelievable – in short, this is where artificial intelligence comes in.

So far, in my experience, geologists have been able to criticise geostatisticians when they make unconvincing assumptions about the stationarity of the autocorrelation of geological properties through space (clearly there is deeper structure at play, which ordinary kriging cannot capture). Meanwhile, geostatisticians have been able to criticise geologists when they make unconvincing assumptions about the uniformity of geological properties within each mapped unit (clearly there is spatial variability within lithological units, which the traditional ‘classification first’ mapping approach cannot capture). Real progress will be made when we can collectively agree on – or at least not be diametrically opposed over – a set of foundational assumptions on which our mapping procedures should operate. Doing so will allow us, as a community, to harness the power of artificial intelligence to generate high quality maps that not only closely match reality but also appear convincing to us as geoscientists. Crucially, using computational approaches also allows us to quantify uncertainties in a way that traditional mapping methods simply cannot compete with, thus providing information that is more transparent and trustworthy to users.

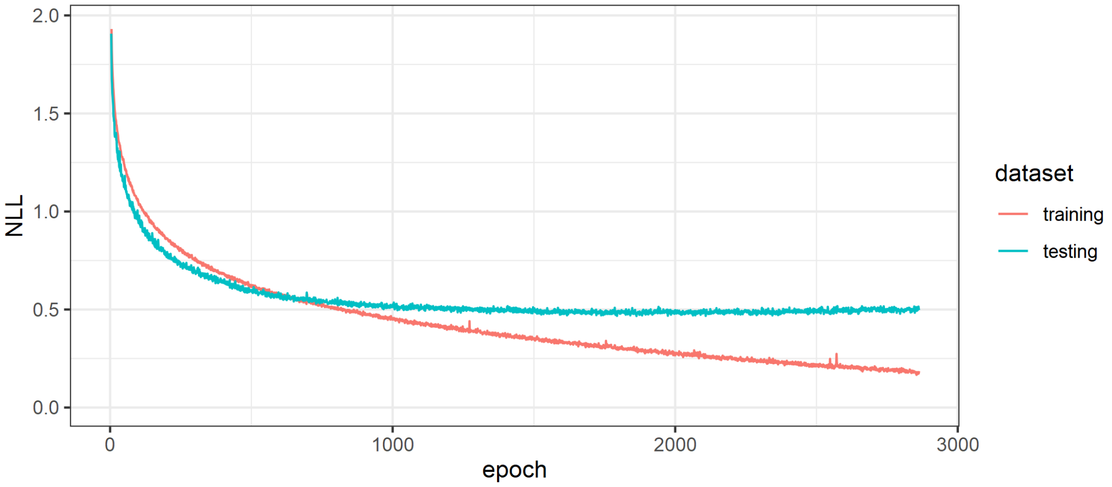


Figure 7.5: Monitoring the negative log-likelihood of the Bayesian neural network as it trains by stochastic gradient descent.

7.2 Artificial intelligence for geological mapping

While the mathematical foundations of how to learn from data can be traced back centuries (to at least Laplace and Bayes in the 1700s) it is only now that our computers are becoming powerful enough to realise the wider potential of what can be achieved – hence the current revolution in data science and artificial intelligence.

Most prominently, the last ten years has seen the breakthrough of ‘deep learning’ – an approach to modelling that uses neural networks with multiple stacked layers to hierarchically learn the salient features of a problem. The classic example is the task of recognising a cat in an image – the combination of features such as whiskers and pointy ears would be highly indicative of the presence of a cat, but previously the recognition of these features relied on manually constructed filters. It is difficult to precisely describe to a computer what exactly a cat’s ears look like, and so these manually constructed filters tended not to be very effective. The breakthrough moment for deep learning came just ten years ago in 2012, when Ukrainian computer scientist Alex Krizhevsky and colleagues used deep neural networks trained on graphical processing units (GPUs) to beat all existing approaches for classifying images. This result has triggered a revolution in artificial intelligence not unlike the revolution in transport triggered by Richard Trevithick’s steam locomotive – these are moments that change the course of history.

The deep learning revolution has been made possible by a combination of advances in computer hardware and efficient algorithms. In simple terms, the thing that is new is the ability to perform optimisation in hugely high dimensional spaces, and in relation to enormous datasets – and these abilities will only continue to improve. Deep learning is perhaps the most effective way to capitalise on these computational advancements, hence its rise to popularity over the last decade. To appreciate the implications, let’s look within the context of geological mapping.

As previously discussed, transitioning from a ‘classification first’ to a ‘properties first’ approach to geological mapping enables direct comparison between our maps and reality, unmuddied by the classification process itself. This allows us to define precise measures of the quality of our maps, and to use these to guide their improvement. Perhaps the best way to assess the quality of a map is to use the principle of likelihood. This is a statistical concept, which refers to the probability that the scenario depicted by a given model (or map in this case) would generate the observations that we see. You could say that if a map is a good match to reality, then the likelihood will be high, because there is a high chance of the geological scenario depicted by the map generating the observations we see in the real world. In classical statistics it is common to take a ‘maximum likelihood’ approach to model fitting – this means using optimisation to pinpoint the values of model parameters that are most likely to have generated the observations we see. I would argue that as mapping geologists we instinctively aim to do the same thing, so that the single geological map (or 3D model) that we produce best explains what we observe in the field. As mapping geologists we may therefore instinctively be likelihood maximisers, even though the ‘parameters’ with which we are working are not components of a digital model, but ideas within a mental model in our own heads – still we are aiming to optimise our geological interpretation in relation to our observations.

The issue of what parameters our models should have (in order that they produce geological maps, in this case) is a difficult one to address. Going back to the cat recognition example, we may expect an effective model to require at least one parameter representing ‘cat ear ness’, but how values of that parameter should relate to the raw pixel values of the input image is almost impossible for us to describe in a precise and accurate way. Likewise for geological mapping, we can expect that our models will require parameters representing the kinds of features we look for in the field as geologists, such as breaks of slope and differences in terrain textures and vegetation to provide clues about the underlying bedrock, as well as contextual associations with surrounding features at a range of scales, but again defining the precise nature of these relationships is a significant challenge.

Deep learning offers a solution to this struggle of trying to define what parameters our models should have, and how they should interact. In its simplest form the deep learning solution is to construct a large multi-layered system of

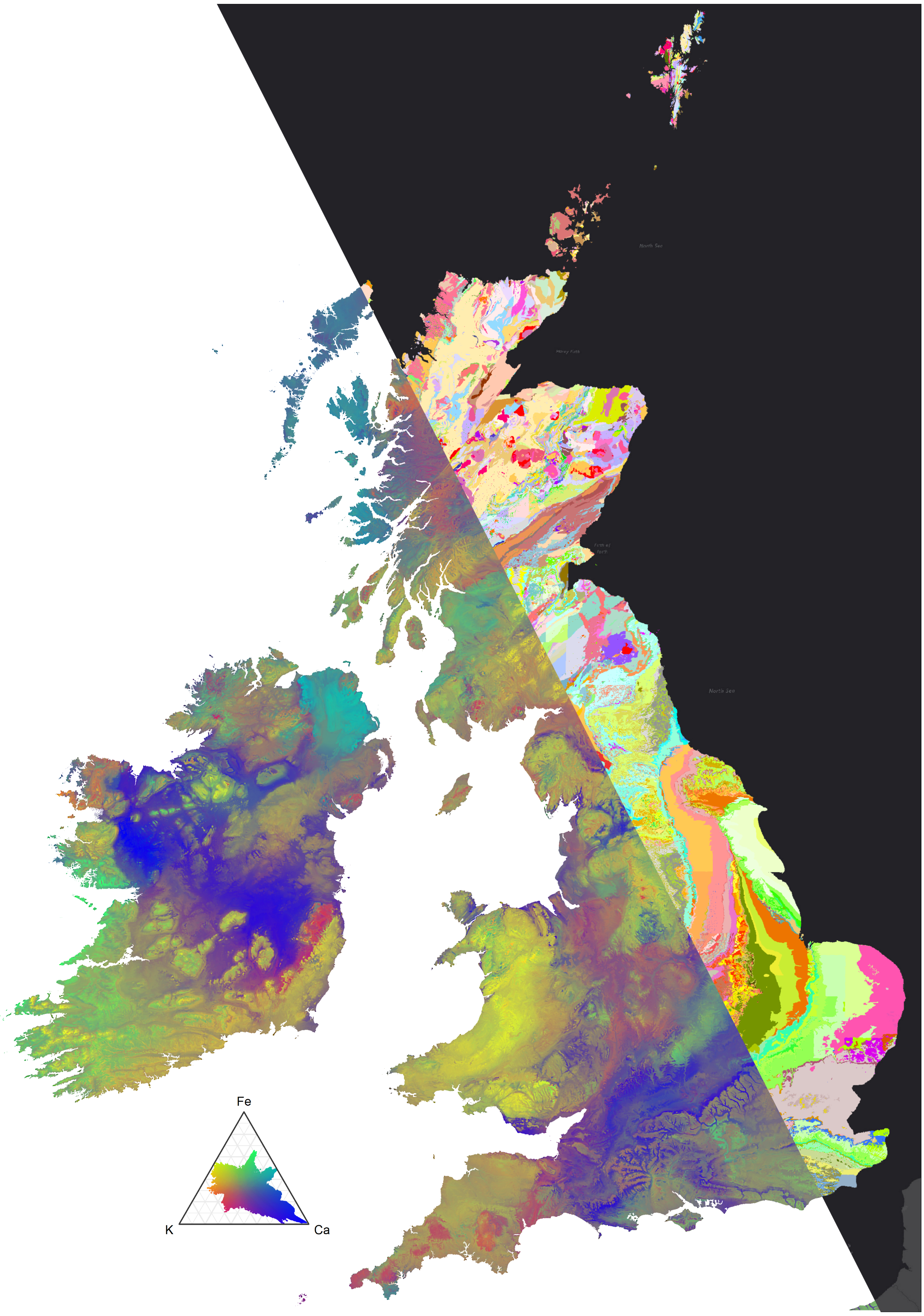


Figure 7.6: (Left of diagonal) AI-generated ‘properties first’ geological map, which provides probabilistic predictions of geological properties through space. The properties mapped here are centred log-ratios of potassium (K; red), iron (Fe; green), and calcium (Ca; blue) concentrations as observed in stream sediments. (Right of diagonal) A traditional ‘classification first’ geological map whereby rock units are manually classified according to convention. Source: The British Geological Survey’s Geology of Britain Viewer (http://mapapps.bgs.ac.uk/geologyofbritain3d/). Contains British Geological Survey materials © UKRI 2022

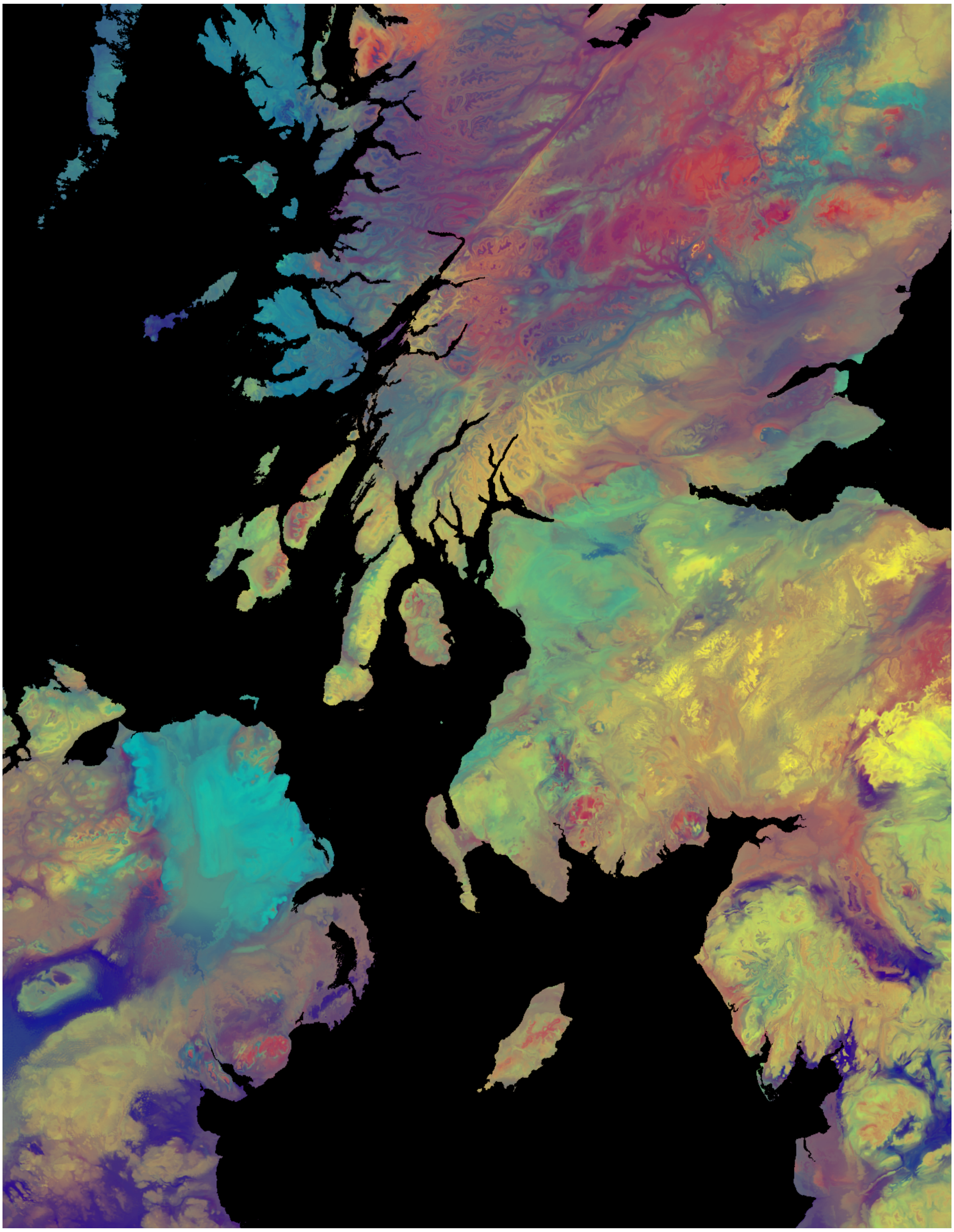


Figure 7.7: A zoomed in section of the AI generated geochemical map, to show finer detail

interconnected parameters – an artificial neural network – and use optimisation to find the values of these parameters which produce the output we require, which in this case may be to produce a geological map with the maximum likelihood of having generated our observations. In concept, this training process (Figure 7.5) is not incomparable to the process of training a human geologist with the necessary skills to produce good geological maps.

Crucially, the more prior knowledge a geologist has, the better job they are likely to do when mapping a new area. The same applies to artificial intelligence – we can incorporate our geological wisdom into the design of AI mapping systems by structuring them such that they are equipped to learn the types of relationships that we believe are important (Figure 7.4, Figure 7.6, Figure 7.7). As such, while deep learning can be conducted 'mindlessly', the best results come from skilful incorporation of prior knowledge into the system. Therefore, a transition to AI mapping does not mean a transition away from the importance of geological expertise. Instead it is an opportunity for us to collate our collective expertise inside the digital domain.

Viewed in the parameter space of our models, the process of maximising the likelihood is quite literally like climbing a mountain of probability – somewhere there exists a summit whose coordinates are the exact values of parameters which correspond to a model (or map in our case) which is most likely to have generated the observations we see. However, as previously mentioned, if we don't have infinite geological observations there will always be room for multiple possible geological interpretations because of the uncertainty about what is going on in the gaps between our observations. Therefore, the act of producing only a single best-fit geological map (the maximum likelihood approach) is inherently overconfident – we are putting all our eggs into one basket despite logic dictating that this basket is just one of many possibilities. It is easy to see how simply using a single 'best' map for important decision making can therefore lead to catastrophic outcomes, because by doing so we are blind to the unavoidable uncertainties involved in the production of the map, which may well be large enough to derail our projects in the real world.

The challenge of dealing with uncertainty is perhaps another reason why the ‘classification first’ geological map has endured so long – without the proper systems in place to model uncertainty, it is easier and ‘safer’ to simply avoid trying to produce detailed maps at all. In the extreme, a single class map that simply labels everything ‘rock’ will never be wrong. However, it will clearly also never be useful – instead we need our predictions to be as specific as possible while remaining honest about uncertainty.

So, how can we deal with uncertainty? We need to shift our goal away from producing a single best geological map (the maximum likelihood approach) and instead aim to produce all of the maps which are possible given the data that we have observed. This is the Bayesian approach, and while it might sound like pie in the sky, our technology is reaching the point that this is now achievable even for extremely complex models like those that geological mapping requires. If maximising the likelihood was like climbing a mountain of probability in parameter space, Bayesian modelling instead requires exploring the entire mountain, or range of mountains, in order to understand the shape of this probability landscape. By modelling all possible maps, we can think of simulating a digital flipbook – infinitely long – where each page shows a realisation of a different possible geological map. As a collective, the possible maps describe a distribution over the possible geological scenarios that could have generated the observations we see. By collecting more observations – perhaps specifically targeting the areas of greatest uncertainty – we can increase our knowledge and reduce the spread between possible geological scenarios, particularly in the immediate vicinity of the observations themselves.

This is all extremely useful, because it means we can plan our activities and policies around probabilities rather than potentially incorrect absolutes, and therefore optimise the outcomes of our geology-related endeavours (as well as optimise the collection of new data). Of course, probabilities need to be well calibrated against reality in order to be useful, but this is all part of the workflow of developing AI systems (Figure 7.8). Interestingly, much of the work in assessing the skill of probabilistic predictions has been developed in the context of weather forecasting, which is in many ways the spatio-temporal cousin of geological mapping, and also had its early roots in hand drawn maps.

Somewhat ironically – because it's only seen as a training exercise – the closest thing we have to a Bayesian approach in the world of traditional (non computational) geological mapping practice is 'the undergraduate mapping trip', on which a cohort of student geologists are each tasked with producing their own map of the same study area. The result is a collection, or ensemble, of possible geological maps. This ensemble conveys uncertainty ("it could be this, or it could be that") better than any single map could, which is the same reason that weather forecasters have been using ensemble models for about the last 30 years. However, in the case of the undergraduate mapping trip we may question the overall skill of the resultant ensemble on the grounds that it has been produced by inexperienced geologists, whose individual interpretations may not always be sensible. But imagine if instead the ensemble of maps was produced by infinitely many world-leading geological mapping experts – in essence this is what artificial intelligence has the potential to provide us with, if we design it well.

7.3 Where the future leads

There is really no way to go about achieving Bayesian geological mapping in a rigorous way without using powerful computers – and so we must port our mapping procedures into the digital domain, and that doesn't mean drawing maps and models by hand on a computer, it means designing machine learning systems that can themselves conduct the task of geological modelling and mapping. Not only will this allow us to produce geological models and maps at new levels of fidelity, and with fully quantified uncertainties, but in the process we would be

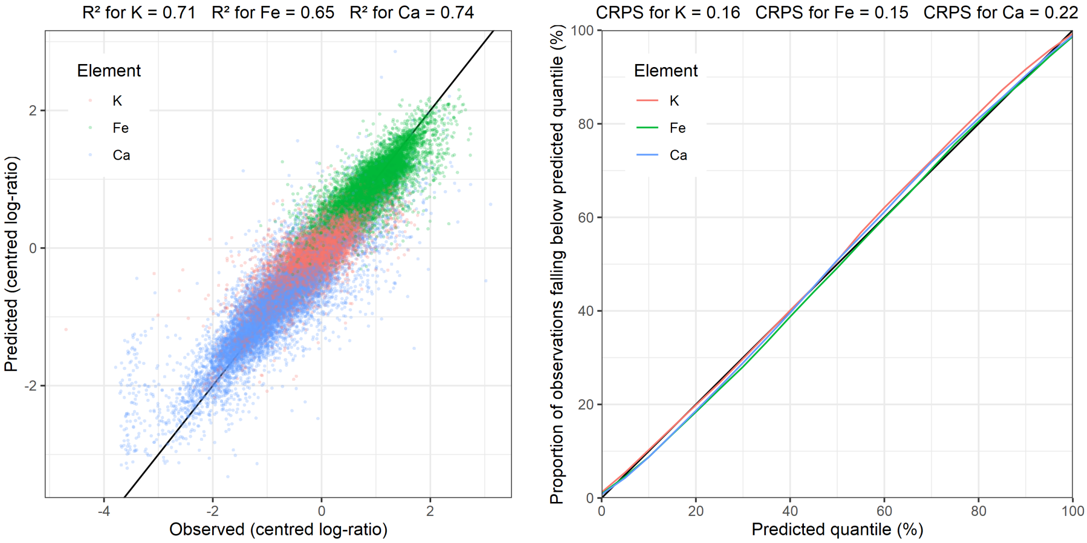


Figure 7.8: Two checks of the quality of the AI map facing the title page, both are made against held-out test observations not seen during the modelling process. Left scatter plot of observed and predicted element concentrations (taking the mean of the AI's predictive distribution – which is what the map shows – as a point prediction). High R^2 values show good fit between the map and unobserved locations. Right quantile-quantile plot assessing the probabilistic calibration of the AI's predictive distribution against observed reality. Calibration is near-perfect, suggesting that as-yet unseen element concentrations will be observed with the same frequencies that the AI's predictive distribution implies.

'laying out on the table' the necessary intellectual machinery required to produce geological maps. Technologically, this will essentially require utilising and further developing modern geostatistical and structural geological methodologies (e.g. Sahu, 2022; Lindsay et al., 2012; Varga and Wellmann, 2016; Wellmann et al., 2018; Hillier et al., 2021; Kirkwood et al., 2022) to incorporate prior knowledge and geological data into geologically plausible models with quantified uncertainties.

By incorporating our geological knowledge into the design of geological learning machines, we would be setting our expertise free from the closed system of our own minds and bringing it out into the open where it can be more effectively utilised. The formula one cars of today are massively more sophisticated (faster, safer, better) compared to their counterparts from 100 years ago. This progress would never have occurred if the idea of a formula one car had remained hidden in the minds of racing enthusiasts rather than turned into real-life machinery whose design could be iterated over and improved upon. Likewise, our current geological maps are the products of thought processes 'hidden' in the minds of geologists. By adopting artificial intelligence approaches to mapping, we can bring our collective mapping expertise out onto the workbench where it can be iterated over and improved upon in a collaborative and transparent way.

It is difficult to predict just how drastic the progress could be if we – the geoscience community – adopt artificial intelligence as an integral part of geological mapping. It seems likely that this transition could be as significant as the transition from horse-drawn wagons to steam locomotives back in the early 1800s. Back then, once we first adopted machine-powered transport, it took only 150 years to go all the way from steam trains to bullet trains and space rockets, with cars and aeroplanes emerging earlier within the first 100 years. What will the equivalent progress in AI powered geological mapping look like? The specifics remain to be seen, but we can be sure that developing AI to improve our ability to distil and convey geological information will benefit humanity, particularly as we face the pressing challenges of the climate crisis. Now more than ever we need to be in tune with our planet, and that means going beyond the opaque polygons of our traditional mapping approaches.

Chapter 8

Summary and Conclusion

This thesis has presented and developed some viable approaches for the application of machine learning methods for probabilistic modelling of the environment. Building on the reasoning and history behind why we should strive to develop and improve our probabilistic models of the environment (Chapter 1), and a background of research in statistical modelling and post-processing in meteorology and geology (Chapter 2, and individual chapter introductions), we have seen how even a relatively simple regression-tree based machine learning system for site-specific weather forecast post-processing (Chapter 3) can improve the calibration of Numerical Weather Prediction ensembles, and also accommodate predictions from pre-calibrated probabilistic forecasts, the likes of which future AI weather forecasting systems are likely to generate.

From this site-specific weather forecast post-processing illustration (Chapter 3), we have expanded the problem space to include spatial (Chapters 4, 5) and spatio-temporal domains (Chapter 6) and have developed a deep neural network architecture that combines computer vision capabilities with geostatistical traits in order to provide 'intelligent' spatial and spatio-temporal interpolation that automatically utilises information contained within auxiliary datasets. We have shown that our deep learning approach is capable of providing well-calibrated predictive distributions on a range of tasks when trained using approximate Bayesian inference (in the form of Monte Carlo dropout, so far). It is perhaps justifiable to suggest that this 'Bayesian deep learning for spatial interpolation in the presence of auxiliary information' approach — now published in the geostatistics and machine learning special issue of Mathematical Geosciences; Kirkwood et al., 2022 — may represent a potential new generation of 'universal' probabilistic interpolation approaches for big data problems where large numbers of environmental observations have been point-sampled over gridded covariates. It is increasingly becoming the case that gridded covariates are available to support environmental modelling, whether from satellite imagery or even from other model output, for example from Numerical Weather Prediction, and so it seems sensible to incorporate computer vision capabilities into the spatial and spatio-temporal interpolation techniques of the future.

Of course, there is no such thing as a true one-size-fits-all approach to modelling (beyond perhaps Bayes' theorem itself) on the grounds that our prior beliefs will (or should) always be different depending on the problem being tackled. As such, model development should always be task-specific. However, as the number of observations increase, the influence that our prior beliefs have on our posterior predictions decreases, which potentially provides more space for the development of 'universal' (or near-universal) machine learning solutions. At the same time, the decision of what covariates to include within a model is already often the strongest statement of prior belief that a practitioner may have (even if they are not formally operating within the Bayesian framework), and so perhaps there is scope for the adoption of near-universal big data modelling methods in the environmental sciences which can be applied to a range of problems involving a range of predictands and covariates. It seems that many opportunities can arise from recognition of the commonalities between different disciplines; for example weather forecasting and geological mapping (Chapter 7), which both aim to model their respective spheres - atmos and lithos - and summarise this information in a useful way. Working through the lens of data science facilitates the combining of ideas from different disciplines in order to develop more capable solutions for the future.

8.1 Future work

In our original work on site-specific weather forecasting (Chapter 3), we referred to the need for 'algorithmic interfaces' to combine and present the information contained within otherwise overwhelming ensembles of weather forecasting models; a problem which will become more important as the number of models available, and their complexity, increases (Kirkwood et al., 2020; Haupt et al., 2021). The quantile regression forest (QRF) based approach that we used was able to improve the calibration of road temperature forecasts on a point-wise basis (i.e., at each forecast location the predictive distribution provided by our approach improved on the calibration of the raw ensemble). However, while the QRF approach is appealing for its simplicity and speed (which is important in an operational environment), it is not capable of providing simulations that are close to realistic due to decision trees not being physically constrained (partitioning of a feature space by feature thresholds is unlike any natural physical process), and the predictive distributions that we derive from the quantile regression forest at each location being independent of one-another.

As explained in the introduction, our overall goal in probabilistic environmental modelling should be to be able to generate predictive distributions that are well-calibrated and as sharp as possible, but ideally these would also be composed of ‘realistic’ or physically plausible individual samples / simulations. For example, these simulations should possess spatial or spatio-temporal autocorrelation structures (or covariance) that are consistent with reality, as well as ideally adhering to physical laws of conservation of mass, momentum and energy (Laubscher and Rousseau, 2022; Wang et al., 2020). Incorporating physical knowledge into machine learning systems is a current hot topic of research that goes by the name of physics-informed machine learning (PIML) (Karniadakis et al., 2021) or knowledge-guided machine learning (Karpatne et al., 2017; Willard et al., 2022), and many avenues have been proposed (e.g. Kashinath et al., 2021) including customised loss functions (Karpatne et al., 2017; Beucler et al., 2019), custom-designed neural network architectures (Mohan et al., 2020; Kashinath, Marcus et al., 2020), and enforcing constraints on the covariance of generative adversarial network (GAN) outputs (Wu et al., 2020).

The logical extension of Chapter 3's site-specific post-processing system is therefore to develop post-processing systems that can produce physically plausible samples whilst being informed by the range of available ensemble models. This requirement has inspired the work of subsequent chapters, which develop Bayesian deep learning architectures for spatial and spatio-temporal modelling which are able to generate predictive distributions that are composed of individual simulations that could each be considered quite plausible - for example see the appendix of Chapter 5, in which we assess the spatial autocorrelation properties of maps simulated by Bayesian deep learning.

The generation of plausible samples from probabilistic machine learning systems is overall the main area of future work following from this thesis. In this research so far we have prioritised modelling methods that are fast and scalable to large datasets, and we have seen how approaches that are relatively simple to apply (e.g. quantile regression forests, Bayesian deep neural networks) can produce good or even excellent results in terms of calibration and sharpness. It was also encouraging to see that our approach to 'Bayesian deep learning for spatial interpolation in the presence of auxiliary information' (Chapter 5) was able to generate simulated maps whose autocorrelation properties match the autocorrelation properties of held-out test data very well (Chapter 5 appendix) despite the model containing no explicit length-scale parameter (unlike typical Gaussian processes, which our deep neural network approach can be seen as approximating inexactly).

This inspires a question of whether the physical plausibility of posterior samples is an emergent property of sufficiently skilful machine learning systems; can maximising the sharpness of a predictive distribution subject to calibration naturally lead to the generation of physically plausible samples? Our results from Chapter 5 suggest that, with the right machine learning systems (i.e., whose prior distribution space does contain the true model, or a sufficiently similar model as to be deemed 'plausible'), the answer may be yes. This could perhaps be tested by training the machine learning system to predict target variables for which we do have exhaustive observations — for example predicting land surface elevation — but to do so by training on datasets of varying size. As the number of training observations increases towards infinity, the posterior distribution should converge on the ‘truth’. Samples from this posterior should therefore become increasingly physically plausible in the process.

Incorporating Gaussian processes into our Bayesian deep learning systems is one approach which may improve the physical plausibility of simulations for a given number of observations, by providing more control over the prior distribution. The approach of ‘Deep Kernel Learning’ (Wilson et al., 2016) for example, which uses deep neural networks to transform the input feature space for Gaussian process interpolation, is a likely candidate for improving control over the physical plausibility of the simulations that our Bayesian deep learning systems generate, by allowing covariance properties to be specified explicitly in the choice of kernel (even though, if these kernels operate on a deep-learned feature space, the interpretation of their parameters becomes more complex).

Additionally, different approaches to (approximate) Bayesian inference in deep neural networks may also impact on the plausibility of generated samples, as well as on the calibration and sharpness of the resultant predictive distributions. This is therefore another area for future research, building on the work presented in this thesis. The development of Bayesian inference techniques for deep models is a hot topic in machine learning research (e.g., Wilson, 2020), and it will be interesting to test and compare different schemes as they emerge. Given that we are still only 10 years into the deep learning revolution, the architectures we develop in this thesis are likely to seem increasingly elementary as time goes on (and to some people they may already seem that way, but the aim has been to develop and apply probabilistic machine learning methods to real-world environmental modelling problems, rather than to progress probabilistic machine learning in a pure sense) and so we can look forward to a future in which our probabilistic models can be made even more effective, although the underlying paradigms will almost certainly remain the same.

8.2 Final remarks

In this thesis we have described the need for probabilistic environmental modelling, and developed and demonstrated some viable machine learning approaches for achieving it in applications ranging from site-specific weather forecast post-processing, to spatio-temporal interpolaton of crowd-sourced weather observations, to a preliminary development of probabilistic geological mapping. We have been motivated by 'the quiet revolution of numerical weather prediction' (Bauer, Thorpe and Brunet, 2015b) and the associated rise (since the early 2000s) of ensemble forecasts for quantifying uncertainty. We have seen how weather forecasts composed of multiple ensemble members bear similarity to Bayesian posterior predictive distributions approximated using Monte Carlo sampling, and have developed a machine learning approach for site-specific weather forecast post-processing that inexpensively corrects and 'fills in' the forecast probability space with more dense sampling in order to produce better calibrated forecasts.

Extending beyond site-specific problems, we have adopted and developed Bayesian deep learning methodologies that combine the strengths of computer vision with approximate Gaussian processes for geostatistical interpolation, and in doing so achieve a new generation of 'intelligent' probabilistic interpolators for simulating continuous maps from point-sampled environmental variables. Our Bayesian deep learning approach for spatial and spatio-temporal interpolation can learn its own features automatically in order to extract relevant information from gridded auxiliary covariates, even where these relationships are potentially complex. As our demonstrations show, the resultant simulated maps (or 'ensemble members') can be highly detailed, and appear capable of capturing structure such that their spatial autocorrelation properties closely match those of held-out test observations. We have also demonstrated how our Bayesian deep learning approach can be expanded into a mixture model, or mixture density network, in order to filter outliers from noisy datasets during the modelling process, making it suitable for use on crowd-sourced datasets in which the quality of observations reported from individual stations is inconsistent.

The work presented in this thesis is intended to inspire further development of machine learning approaches for probabilistic environmental modelling. It is clear that having good-quality probabilistic models of the environment will be important for our future, and despite the explosion in popularity of machine learning over the last decade, it seems that many opportunities for progress within the environmental modelling space remain under-explored. The experience of this PhD has shed light on some of these opportunities, and of the benefits that cross-disciplinary thinking can provide. No doubt the future will see these areas be developed further.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016). 'Tensorflow: A system for large-scale machine learning'. In: *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283 (Cited on page 132).

Abbe, C. (Dec. 1901). 'The physical basis of long-range weather forecasts'. In: *Monthly Weather Review* 29.12, pp. 551–561. ISSN: 0027-0644. (Visited on 08/02/2019) (Cited on pages 53 and 74).

Adadi, A. and Berrada, M. (2018). 'Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)'. In: *IEEE access* 6, pp. 52138–52160 (Cited on page 123).

Akaike, H. (1974). 'A new look at the statistical model identification'. In: *IEEE transactions on automatic control* 19.6, pp. 716–723 (Cited on page 26).

Alzubaidi, F., Mostaghimi, P., Swietojanski, P., Clark, S. R. and Armstrong, R. T. (2021). 'Automated lithology classification from drill core images using convolutional neural networks'. In: *Journal of Petroleum Science and Engineering* 197, p. 107933 (Cited on page 128).

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q. and Chen, G. (2016). 'Deep speech 2: End-to-end speech recognition in english and mandarin'. In: pp. 173–182 (Cited on page 56).

Arrieta, A. B., Diaz-Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R. et al. (2020). 'Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI'. In: *Information fusion* 58, pp. 82–115 (Cited on page 123).

Asanjan, A. A., Yang, T., Hsu, K., Sorooshian, S., Lin, J. and Peng, Q. (2018). 'Short-Term Precipitation Forecast Based on the PERSIANN System and LSTM Recurrent Neural Networks'. en. In: *Journal of Geophysical Research:*

Atmospheres 123.22, pp. 12, 543–12, 563. ISSN: 2169-8996. DOI: 10.1029/2018JD028375. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JD028375 (visited on 25/04/2020) (Cited on page 78).

Athey, S., Tibshirani, J. and Wager, S. (Apr. 2019). 'Generalized random forests'. EN. In: *Annals of Statistics* 47.2, pp. 1148–1178. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/18-AOS1709. URL: https://projecteuclid.org/euclid.aos/1547197251 (visited on 27/04/2020) (Cited on page 81).

Atzori, L., Iera, A. and Morabito, G. (2010). 'The internet of things: A survey'. In: *Computer networks* 54.15, pp. 2787–2805 (Cited on page 166).

Ayadi, A., Ghorbel, O., Obeid, A. M. and Abid, M. (2017). 'Outlier detection approaches for wireless sensor networks: A survey'. In: *Computer Networks* 129, pp. 319–333 (Cited on page 167).

Bakar, K. S. and Sahu, S. K. (2015). 'spTimer: Spatio-temporal Bayesian modeling using R'. In: *Journal of Statistical Software* 63, pp. 1–32 (Cited on page 193).

Bauer, P., Thorpe, A. and Brunet, G. (Sept. 2015a). 'The quiet revolution of numerical weather prediction'. en. In: *Nature* 525.7567, pp. 47–55. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14956. URL: http://www.nature.com/doifinder/10.1038/nature14956 (visited on 23/10/2018) (Cited on pages 55, 57 and 74).

Bauer, P., Thorpe, A. and Brunet, G. (2015b). 'The quiet revolution of numerical weather prediction'. In: *Nature* 525.7567, pp. 47–55 (Cited on pages 64, 183, 184 and 220).

Bayes, T. (1763). 'LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S'. In: *Philosophical transactions of the Royal Society of London* 53, pp. 370–418 (Cited on page 20).

Behrens, T., Schmidt, K., MacMillan, R. A. and Rossel, R. A. V. (2018). 'Multi-scale digital soil mapping with deep learning'. In: *Scientific reports* 8.1, pp. 1–9 (Cited on page 174).

Belkin, M., Hsu, D., Ma, S. and Mandal, S. (Aug. 2019). ‘Reconciling modern machine-learning practice and the classical bias–variance trade-off’. en. In: *Proceedings of the National Academy of Sciences* 116.32, pp. 15849–15854. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1903070116. URL: https://www.pnas.org/content/116/32/15849 (visited on 27/04/2020) (Cited on page 89).

Bell, S., Cornford, D. and Bastin, L. (2015). ‘How good are citizen weather stations? Addressing a biased opinion’. In: *Weather* 70.3, pp. 75–84 (Cited on pages 163 and 172).

Bengtsson, L., Kanamitsu, M., Kallberg, P. and Uppala, S. (1982). ‘FGGE 4-dimensional data assimilation at ECMWF (weather forecasts).’ In: *Bulletin of the American Meteorological Society* 63, pp. 29–43 (Cited on page 183).

Benjamin, S. G., Brown, J. M., Brunet, G., Lynch, P., Saito, K. and Schlatter, T. W. (2019). ‘100 years of progress in forecasting and NWP applications’. In: *Meteorological Monographs* 59, pp. 13–1 (Cited on page 183).

Bentzien, S. and Friederichs, P. (2014). ‘Decomposition and graphical portrayal of the quantile score’. In: *Quarterly Journal of the Royal Meteorological Society* 140.683, pp. 1924–1934 (Cited on page 89).

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media (Cited on page 126).

Berrocal, V. J., Raftery, A. E., Gneiting, T. and Steed, R. C. (June 2010). ‘Probabilistic Weather Forecasting for Winter Road Maintenance’. In: *Journal of the American Statistical Association* 105.490, pp. 522–537. ISSN: 0162-1459. DOI: 10.1198/jasa.2009.ap07184. URL: https://doi.org/10.1198/jasa.2009.ap07184 (visited on 22/10/2019) (Cited on page 52).

Berti, M., Corsini, A. and Daehne, A. (2013). ‘Comparative analysis of surface roughness algorithms for the identification of active landslides’. In: *Geomorphology* 182, pp. 1–18 (Cited on page 103).

Beucler, T., Rasp, S., Pritchard, M. and Gentine, P. (2019). ‘Achieving conservation of energy in neural network emulators for climate modeling’. In: *arXiv preprint arXiv:1906.06622* (Cited on page 217).

Beven, K. (2007). ‘Towards integrated environmental models of everywhere: uncertainty, data and modelling as a learning process’. In: *Hydrology and Earth System Sciences* 11.1, pp. 460–467 (Cited on page 165).

Beven, K., Cloke, H., Pappenberger, F., Lamb, R. and Hunter, N. (2015). ‘Hyperresolution information and hyperresolution ignorance in modelling the hydrology of the land surface’. In: *Science China Earth Sciences* 58.1, pp. 25–35 (Cited on page 165).

Beven, K. J. and Alcock, R. E. (2012). ‘Modelling everything everywhere: a new approach to decision-making for water management under uncertainty’. In: *Freshwater Biology* 57, pp. 124–132 (Cited on page 165).

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X. and Tian, Q. (2022). ‘Pangu-Weather: A 3D High-Resolution Model for Fast and Accurate Global Weather Forecast’. In: *arXiv preprint arXiv:2211.02556* (Cited on page 194).

Bishop, C. M. (1994). *Mixture density networks*. Tech. rep. Aston University (Cited on page 168).

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer (Cited on page 26).

Bjerknes, V. (1904). ‘Das Problem der Wettervorhers-age, betrachtet vom Standpunkte der Mechanik und der Physik’. In: *Meteor. Z.* 21, pp. 1–7 (Cited on pages 53 and 74).

Blair, G. S., Beven, K., Lamb, R., Bassett, R., Cauwenberghs, K., Hankin, B., Dean, G., Hunter, N., Edwards, L., Nundloll, V. et al. (2019a). ‘Models of everywhere revisited: A technological perspective’. In: *Environmental Modelling & Software* 122, p. 104521 (Cited on page 165).

Blair, G. S., Henrys, P., Leeson, A., Watkins, J., Eastoe, E., Jarvis, S. and Young, P. J. (2019b). ‘Data science of the natural environment: a research roadmap’. In: *Frontiers in Environmental Science* 7, p. 121 (Cited on page 166).

Blangiardo, M., Cameletti, M., Baio, G. and Rue, H. (2013). 'Spatial and spatio-temporal models with R-INLA'. In: *Spatial and spatio-temporal epidemiology* 4, pp. 33–49 (Cited on page 193).

Blomqvist, K., Kaski, S. and Heinonen, M. (2019). 'Deep convolutional Gaussian processes'. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 582–597 (Cited on page 120).

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U. and Zhang, J. (2016). 'End to end learning for self-driving cars'. In: *arXiv preprint arXiv:1604.07316* (Cited on page 56).

Bolin, B. (1955). 'Numerical Forecasting with the Barotropic Model 1'. In: *Tellus* 7.1, pp. 27–49. ISSN: 0040-2826 (Cited on page 54).

Bowler, N. E., Arribas, A., Mylne, K. R., Robertson, K. B. and Beare, S. E. (2008). 'The MOGREPS short-range ensemble prediction system'. In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 134.632, pp. 703–722 (Cited on page 58).

Breiman, L. (2001). 'Random forests'. In: *Machine learning* 45, pp. 5–32 (Cited on page 82).

Brier, G. W. (1946). *A study of quantitative precipitation forecasting in the TVA basin*. 26. US Department of Commerce, Weather Bureau (Cited on page 54).

British Geological Survey (2020). *Geology of Britain Viewer*. Accessed through online web interface at mapapps.bgs.ac.uk/geologyofbritain/home.html. URL: `mapapps.bgs.ac.uk/geologyofbritain/home.html` (Cited on page 151).

Byers, H., Landsberg, H., Wexler, H., Haurwitz, B., Spilhaus, A., Willett, H., Houghton, H., Allen, R. and Vernon, E. (1951). 'Objective weather forecasting'. In: *Compendium of Meteorology: Prepared under the Direction of the Committee on the Compendium of Meteorology*, pp. 796–801 (Cited on page 54).

Calanca, P., Bolius, D., Weigel, A. and Liniger, M. (2011). 'Application of long-range weather forecasts to agricultural decision problems in Europe'. In: *The Journal of Agricultural Science* 149.1, pp. 15–22 (Cited on page 52).

Calandra, R., Peters, J., Rasmussen, C. E. and Deisenroth, M. P. (2016). ‘Manifold Gaussian processes for regression’. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 3338–3345 (Cited on page 69).

Cannon, A. J. (Sept. 2011). ‘Quantile regression neural networks: Implementation in R and application to precipitation downscaling’. en. In: *Computers & Geosciences* 37.9, pp. 1277–1284. ISSN: 0098-3004. DOI: 10.1016/j.cageo.2010.07.005. URL: http://www.sciencedirect.com/science/article/pii/S009830041000292X (visited on 06/04/2020) (Cited on page 90).

Cawley, G. C., Janacek, G. J., Haylock, M. R. and Dorling, S. R. (2007). ‘Predictive uncertainty in environmental modelling’. In: *Neural networks* 20.4, pp. 537–549 (Cited on page 152).

Chapman, L., Bell, C. and Bell, S. (2017). ‘Can the crowdsourcing data paradigm take atmospheric science to a new level? A case study of the urban heat island of London quantified using Netatmo weather stations’. In: *International Journal of Climatology* 37.9, pp. 3597–3605 (Cited on page 163).

Chapman, W. E., Subramanian, A. C., Monache, L. D., Xie, S. P. and Ralph, F. M. (2019). ‘Improving Atmospheric River Forecasts With Machine Learning’. en. In: *Geophysical Research Letters* 0.0. ISSN: 1944-8007. DOI: 10.1029/2019GL083662. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GL083662 (visited on 19/09/2019) (Cited on pages 63 and 75).

Chapman, W. E., Haupt, S. E., Kirkwood, C., Lerch, S., Matsueda, M. and Subramanian, A. C. (n.d.). ‘Data from: Towards Implementing AI Post-processing in Weather and Climate: Proposed Actions from the Oxford 2019 Workshop’. In: (). DOI: https://doi.org/10.6075/J08S4NDM (Cited on pages 95 and 97).

Charney, J. G., Fjörtoft, R. and Neumann, J. v. (1950). ‘Numerical integration of the barotropic vorticity equation’. In: *Tellus* 2.4, pp. 237–254. ISSN: 0040-2826 (Cited on page 54).

Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G. and Tang, L. (2017). ‘Hyperspectral images classification with Gabor filtering and convolutional neural network’. In:

IEEE Geoscience and Remote Sensing Letters 14.12, pp. 2355–2359 (Cited on page 132).

Clark, I. (2010). 'Statistics or geostatistics? Sampling error or nugget effect?' In: *Journal of the Southern African Institute of Mining and Metallurgy* 110.6, pp. 307–312 (Cited on page 158).

Colomina, I. and Molina, P. (2014). 'Unmanned aerial systems for photogrammetry and remote sensing: A review'. In: *ISPRS Journal of photogrammetry and remote sensing* 92, pp. 79–97 (Cited on page 126).

Coveney, P. V., Dougherty, E. R. and Highfield, R. R. (2016). 'Big data need big theory too'. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2080, p. 20160153 (Cited on page 104).

Cressie, N. (1990). 'The origins of kriging'. In: *Mathematical geology* 22.3, pp. 239–252 (Cited on pages 126 and 130).

Dabernig, M., Mayr, G. J., Messner, J. W. and Zeileis, A. (2017). 'Spatial ensemble post-processing with standardized anomalies'. In: *Quarterly Journal of the Royal Meteorological Society* 143.703, pp. 909–916 (Cited on pages 79 and 96).

Dale, M., Wicks, J., Mylne, K., Pappenberger, F., Laeger, S. and Taylor, S. (2014). 'Probabilistic flood forecasting and decision-making: an innovative risk-based approach'. In: *Natural hazards* 70.1, pp. 159–172 (Cited on page 52).

Damianou, A. and Lawrence, N. (2013). 'Deep gaussian processes'. In: *Artificial Intelligence and Statistics*, pp. 207–215 (Cited on pages 69 and 120).

Das, A. and Rad, P. (2020). 'Opportunities and challenges in explainable artificial intelligence (xai): A survey'. In: *arXiv preprint arXiv:2006.11371* (Cited on page 123).

Diggle, P. J., Tawn, J. A. and Moyeed, R. A. (1998). 'Model-based geostatistics'. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 47.3, pp. 299–350 (Cited on pages 68, 102, 120 and 131).

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. and Saurous, R. A. (2017). 'Tensorflow distributions'. In: *arXiv preprint arXiv:1711.10604* (Cited on page 132).

Dimitrakopoulos, R. (1998). 'Conditional simulation algorithms for modelling orebody uncertainty in open pit optimisation'. In: *International Journal of Mining, Reclamation and Environment* 12.4, pp. 173–179 (Cited on page 159).

Dimitrakopoulos, R. (2018). 'Stochastic mine planning—methods, examples and value in an uncertain world'. In: *Advances in Applied Strategic Mine Planning*. Springer, pp. 101–115 (Cited on page 159).

Düben, P., Modigliani, U., Geer, A., Siemen, S., Pappenberger, F., Bauer, P., Brown, A., Palkovic, M., Raoult, B., Wedi, N. et al. (Jan. 2021). 'Machine learning at ECMWF: A roadmap for the next 10 years'. In: 878. DOI: 10.21957/ge7ckgm. URL: https://www.ecmwf.int/node/19877 (Cited on page 56).

Economou, T., Stephenson, D. B., Rougier, J. C., Neal, R. A. and Mylne, K. R. (Oct. 2016). 'On the use of Bayesian decision theory for issuing natural hazard warnings'. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 472.2194, p. 20160295. DOI: 10.1098/rspa.2016.0295. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.2016.0295 (visited on 20/02/2020) (Cited on page 74).

Farquhar, S., Osborne, M. and Gal, Y. (2019). *Radial Bayesian Neural Networks: Beyond Discrete Support In Large-Scale Bayesian Deep Learning*. arXiv: 1907.00865 [stat.ML] (Cited on page 118).

Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L. et al. (2007). 'The shuttle radar topography mission'. In: *Reviews of geophysics* 45.2 (Cited on page 169).

Fox, C. R. and Ülkümen, G. (2011). 'Distinguishing two dimensions of uncertainty'. In: *Perspectives on thinking, judging, and decision making* 14 (Cited on page 136).

Friedman, R. M. (1993). *Appropriating the weather: Vilhelm Bjerknes and the construction of a modern meteorology*. Cornell University Press. ISBN: 0-8014-8160-0 (Cited on page 53).

Gal, Y. and Ghahramani, Z. (2016). 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning'. In: *international conference on machine learning*, pp. 1050–1059 (Cited on pages 69, 118, 127, 129, 136, 138, 141 and 177).

Gal, Y., Hron, J. and Kendall, A. (2017). 'Concrete dropout'. In: *arXiv preprint arXiv:1705.07832* (Cited on page 177).

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press (Cited on pages 41, 67 and 160).

Genest, C. (1992). 'Vincentization Revisited'. In: *The Annals of Statistics* 20.2, pp. 1137–1142. ISSN: 0090-5364. URL: https://www.jstor.org/stable/2242003 (visited on 09/03/2020) (Cited on page 87).

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G. and Yacalis, G. (2018). 'Could Machine Learning Break the Convection Parameterization Deadlock?' en. In: *Geophysical Research Letters* 45.11, pp. 5742–5751. ISSN: 1944-8007. DOI: 10.1029/2018GL078202. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL078202 (visited on 16/02/2019) (Cited on page 57).

Gibbs, M. N. (1998). 'Bayesian Gaussian processes for regression and classification'. PhD thesis. Citeseer (Cited on page 120).

Glahn, H. (1982). 'Statistical Weather Forecasting'. In: *Probability, Statistics, and Decision Making in the Atmospheric Sciences*. Ed. by A. H. Murphy and R. W. Katz. Boulder: Westview Press (Cited on pages 54 and 55).

Glahn, H. R. and Lowry, D. A. (1972). 'The use of model output statistics (MOS) in objective weather forecasting'. In: *Journal of applied meteorology* 11.8, pp. 1203–1211 (Cited on page 77).

Gleick, J. (1993). *Genius: The life and science of Richard Feynman*. Vintage (Cited on page 41).

Gneiting, T., Balabdaoui, F. and Raftery, A. E. (2007). ‘Probabilistic forecasts, calibration and sharpness’. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.2, pp. 243–268 (Cited on pages 27, 28, 29, 31, 148, 153 and 181).

Gneiting, T. and Katzfuss, M. (2014). ‘Probabilistic forecasting’. In: *Annual Review of Statistics and Its Application* 1, pp. 125–151 (Cited on page 151).

Gneiting, T. and Raftery, A. E. (2007). ‘Strictly proper scoring rules, prediction, and estimation’. In: *Journal of the American statistical Association* 102.477, pp. 359–378 (Cited on pages 34, 36, 89 and 148).

Gneiting, T., Raftery, A. E., Westveld III, A. H. and Goldman, T. (2005). ‘Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation’. In: *Monthly Weather Review* 133.5, pp. 1098–1118 (Cited on pages 58, 59, 60, 61, 77 and 182).

Gneiting, T. and Ranjan, R. (2013). ‘Combining predictive distributions’. EN. In: *Electronic Journal of Statistics* 7, pp. 1747–1782. ISSN: 1935-7524. DOI: `10.1214/13-EJS823`. URL: `https://projecteuclid.org/euclid.ejs/1372861687` (visited on 27/04/2020) (Cited on pages 87 and 88).

Gödel, K. (1931). *On formally undecidable propositions of Principia Mathematica and related systems*. Courier Corporation (Cited on page 18).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). ‘Generative adversarial nets’. In: pp. 2672–2680 (Cited on page 63).

Gotway, C. A. and Hartford, A. H. (1996). ‘Geostatistical methods for incorporating auxiliary information in the prediction of spatial variables’. In: *Journal of Agricultural, Biological, and Environmental Statistics*, pp. 17–39 (Cited on page 131).

Graves, A. (2011). ‘Practical variational inference for neural networks’. In: *Advances in neural information processing systems*. Citeseer, pp. 2348–2356 (Cited on page 177).

Gringorten, I. I. (1955). ‘Methods of objective weather forecasting’. In: *Advances in geophysics*. Vol. 2. Elsevier, pp. 57–92 (Cited on page 54).

Grose, L., Ailleres, L., Laurent, G., Armit, R. and Jessell, M. (2019). ‘Inversion of geological knowledge for fold geometry’. In: *Journal of Structural Geology* 119, pp. 1–14 (Cited on page 130).

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J. et al. (Dec. 2016). ‘Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs’. en. In: *JAMA* 316.22, pp. 2402–2410. ISSN: 0098-7484. DOI: 10.1001/jama.2016.17216. URL: https://jamanetwork.com/journals/jama/fullarticle/2588763 (visited on 24/04/2020) (Cited on page 75).

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S. and Yang, G.-Z. (2019). ‘XAI—Explainable artificial intelligence’. In: *Science robotics* 4.37, eaay7120 (Cited on page 123).

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. and Lew, M. S. (2016). ‘Deep learning for visual understanding: A review’. In: *Neurocomputing* 187, pp. 27–48 (Cited on page 123).

Hagedorn, R., Doblas-Reyes, F. J. and Palmer, T. (2005). ‘The rationale behind the success of multi-model ensembles in seasonal forecasting—I. Basic concept’. In: *Tellus A: Dynamic Meteorology and Oceanography* 57.3, pp. 219–233. ISSN: 1600-0870 (Cited on page 63).

Han, J., Zhang, D., Cheng, G., Guo, L. and Ren, J. (2014). ‘Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning’. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.6, pp. 3325–3337 (Cited on page 105).

Handcock, M. S. and Stein, M. L. (1993). ‘A Bayesian analysis of kriging’. In: *Technometrics* 35.4, pp. 403–410 (Cited on page 127).

Harff, J. and Davis, J. (1990). ‘Regionalization in geology by multivariate classification’. In: *Mathematical Geology* 22, pp. 573–588 (Cited on page 202).

Hastie, T., Tibshirani, R., Friedman, J. H. and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer (Cited on pages 23 and 24).

Haupt, S. E., Chapman, W., Adams, S. V., Kirkwood, C., Hosking, J. S., Robinson, N. H., Lerch, S. and Subramanian, A. C. (2021). 'Towards implementing artificial intelligence post-processing in weather and climate: proposed actions from the Oxford 2019 workshop'. In: *Philosophical Transactions of the Royal Society A* 379.2194, p. 20200091 (Cited on pages 95, 97 and 217).

Heek, J. and Kalchbrenner, N. (2019). 'Bayesian inference for large scale image classification'. In: *arXiv preprint arXiv:1908.03491* (Cited on page 177).

Heinonen, M., Mannerström, H., Rousu, J., Kaski, S. and Lähdesmäki, H. (2016). 'Non-stationary gaussian process regression with hamiltonian monte carlo'. In: *Artificial Intelligence and Statistics*. PMLR, pp. 732–740 (Cited on page 68).

Hengl, T., Heuvelink, G. B. M., Perčec Tadić, M. and Pebesma, E. J. (Jan. 2012). 'Spatio-temporal prediction of daily temperatures using time-series of MODIS LST images'. en. In: *Theoretical and Applied Climatology* 107.1, pp. 265–277. ISSN: 1434-4483. DOI: 10.1007/s00704-011-0464-2. URL: https://doi.org/10.1007/s00704-011-0464-2 (visited on 25/04/2020) (Cited on page 78).

Hengl, T., Heuvelink, G. B. and Rossiter, D. G. (2007). 'About regression-kriging: From equations to case studies'. In: *Computers & geosciences* 33.10, pp. 1301–1315 (Cited on pages 71 and 131).

Hey, T., Butler, K., Jackson, S. and Thiyagalingam, J. (Mar. 2020). 'Machine learning and big scientific data'. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 378.2166, p. 20190054. DOI: 10.1098/rsta.2019.0054. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2019.0054 (visited on 24/04/2020) (Cited on page 75).

Hijmans, R. J. (2017). *raster: Geographic Data Analysis and Modeling*. R package version 2.6-7. URL: https://CRAN.R-project.org/package=raster (Cited on page 144).

Hillier, M., Wellmann, F., Brodaric, B., Kemp, E. de and Schetselaar, E. (2021). ‘Three-dimensional structural geological modeling using graph neural networks’. In: *Mathematical Geosciences* 53.8, pp. 1725–1749 (Cited on page 213).

Hoeting, J. A., Madigan, D., Raftery, A. E. and Volinsky, C. T. (Nov. 1999). ‘Bayesian model averaging: a tutorial (with comments by M. Clyde, David Draper and E. I. George, and a rejoinder by the authors’. en. In: *Statistical Science* 14.4, pp. 382–417. ISSN: 0883-4237, 2168-8745. DOI: `10.1214/ss/1009212519`. URL: `https://projecteuclid.org/euclid.ss/1009212519` (visited on 21/10/2019) (Cited on page 87).

Hudson, G. and Wackernagel, H. (1994). ‘Mapping temperature using kriging with external drift: theory and an example from Scotland’. In: *International journal of Climatology* 14.1, pp. 77–91 (Cited on page 101).

Hughes, P. (1988). ‘FitzRoy the Forecaster: Prophet without Honor’. In: *Weatherwise* 41.4, pp. 200–204 (Cited on page 53).

Idrissi, M. A. J., Ramchoun, H., Ghanou, Y. and Ettaouil, M. (2016). ‘Genetic algorithm for neural network architecture optimization’. In: *2016 3rd International conference on logistics operations management (GOL)*. IEEE, pp. 1–4 (Cited on page 133).

Johnson, C., Breward, N., Ander, E. and Ault, L. (2005). ‘G-BASE: baseline geochemical mapping of Great Britain and Northern Ireland’. In: *Geochemistry: exploration, environment, analysis* 5.4, pp. 347–357 (Cited on pages 99, 107 and 144).

Journel, A. G. (1974). ‘Geostatistics for conditional simulation of ore bodies’. In: *Economic Geology* 69.5, pp. 673–687 (Cited on page 159).

Journel, A. G. and Rossi, M. (1989). ‘When do we need a trend model in kriging?’ In: *Mathematical Geology* 21.7, pp. 715–739 (Cited on pages 67 and 153).

Kampffmeyer, M., Salberg, A.-B. and Jenssen, R. (2016). ‘Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks’. In: *Proceedings of the IEEE confer-*

ence on computer vision and pattern recognition workshops, pp. 1–9 (Cited on page 129).

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B. and Xing, E. P. (2018). 'Neural architecture search with bayesian optimisation and optimal transport'. In: *Advances in neural information processing systems* 31 (Cited on page 133).

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S. and Yang, L. (2021). 'Physics-informed machine learning'. In: *Nature Reviews Physics* 3.6, pp. 422–440 (Cited on page 217).

Karpatne, A., Watkins, W., Read, J. and Kumar, V. (2017). 'Physics-guided neural networks (pgnn): An application in lake temperature modeling'. In: *arXiv preprint arXiv:1710.11431* 2 (Cited on page 217).

Kashinath, K., Marcus, P. et al. (2020). 'Enforcing physical constraints in cnns through differentiable pde layer'. In: *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (Cited on page 217).

Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A. et al. (2021). 'Physics-informed machine learning: case studies for weather and climate modelling'. In: *Philosophical Transactions of the Royal Society A* 379.2194, p. 20200093 (Cited on page 217).

Katz, R. W. and Murphy, A. H. (2005). *Economic value of weather and climate forecasts*. Cambridge University Press (Cited on page 52).

Kendall, A. and Gal, Y. (2017). 'What uncertainties do we need in bayesian deep learning for computer vision?' In: *Advances in neural information processing systems*, pp. 5574–5584 (Cited on pages 118, 128, 136, 137 and 143).

Kim, S.-M., Choi, Y., Yi, H. and Park, H.-D. (2017). 'Geostatistical prediction of heavy metal concentrations in stream sediments considering the stream networks'. In: *Environmental Earth Sciences* 76.2, p. 72 (Cited on page 113).

Kimura, R. (2002). 'Numerical weather prediction'. In: *Journal of Wind Engineering and Industrial Aerodynamics* 90.12-15, pp. 1403–1414 (Cited on page 182).

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D. and Kim, B. (2019). 'The (un) reliability of saliency methods'. In: *Explainable AI: Interpreting, explaining and visualizing deep learning*, pp. 267–280 (Cited on page 123).

Kingma, D. P. and Ba, J. (2014). 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (Cited on page 147).

Kingma, D. P. and Welling, M. (2013). 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114* (Cited on page 63).

Kirk, P. J., Clark, M. R. and Creed, E. (2021). 'Weather observations website'. In: *Weather* 76.2, pp. 47–49 (Cited on pages 163, 166 and 169).

Kirkwood, C. (2016). *A dropout-regularised neural network for mapping arsenic enrichment in SW England using MXNet*. British Geological Survey. eprint: `http://nora.nerc.ac.uk/id/eprint/514614` (Cited on pages 116 and 126).

Kirkwood, C. (2020). 'Deep covariate-learning: optimising information extraction from terrain texture for geostatistical modelling applications'. In: *arXiv preprint arXiv:2005.11194* (Cited on pages 129, 170 and 174).

Kirkwood, C., Cave, M., Beamish, D., Grebby, S. and Ferreira, A. (2016a). 'A machine learning approach to geochemical mapping'. In: *Journal of Geochemical Exploration* 167, pp. 49–61 (Cited on pages 33, 35, 112, 116 and 126).

Kirkwood, C., Cooper, M., Ferreira, A. and Beamish, D. (2017). 'Unmixing and mapping components of Northern Ireland's geochemical composition using FastICA and random forests'. In: *EarthArXiv preprint http://eartharxiv.org/8k3f7/* (Cited on page 116).

Kirkwood, C., Economou, T., Odbert, H. and Pugeault, N. (2020). 'A framework for probabilistic weather forecast post-processing across models and lead times using machine learning'. In: *Philosophical Transactions of the Royal Society of London: Series A* (Cited on pages 152 and 217).

Kirkwood, C., Economou, T., Pugeault, N. and Odbert, H. (2022). 'Bayesian deep learning for spatial interpolation in the presence of auxiliary information'. In:

Mathematical Geosciences 54.3, pp. 507–531 (Cited on pages 170, 174, 204, 213 and 215).

Kirkwood, C., Everett, P., Ferreira, A. and Lister, B. (2016b). 'Stream sediment geochemistry as a tool for enhancing geological understanding: An overview of new data from south west England'. In: *Journal of Geochemical Exploration* 163, pp. 28–40 (Cited on page 107).

Krige, D. G. (1951). 'A statistical approach to some basic mine valuation problems on the Witwatersrand'. In: *Journal of the Southern African Institute of Mining and Metallurgy* 52.6, pp. 119–139 (Cited on pages 66, 101 and 130).

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). 'Imagenet classification with deep convolutional neural networks'. In: *Advances in neural information processing systems*, pp. 1097–1105 (Cited on pages 51, 55, 75, 104, 127 and 128).

Kurth, T., Subramanian, S., Harrington, P., Pathak, J., Mardani, M., Hall, D., Miele, A., Kashinath, K. and Anandkumar, A. (2022). 'FourCastNet: Accelerating Global High-Resolution Weather Forecasting using Adaptive Fourier Neural Operators'. In: *arXiv preprint arXiv:2208.05419* (Cited on page 56).

Laaha, G., Skøien, J. and Blöschl, G. (2014). 'Spatial prediction on river networks: comparison of top-kriging with regional regression'. In: *Hydrological Processes* 28.2, pp. 315–324 (Cited on page 154).

Lakshminarayanan, B., Pritzel, A. and Blundell, C. (2016). 'Simple and scalable predictive uncertainty estimation using deep ensembles'. In: *arXiv preprint arXiv:1612.01474* (Cited on page 180).

Lamichhane, S., Kumar, L. and Wilson, B. (2019). 'Digital soil mapping algorithms and covariates for soil organic carbon mapping and their implications: A review'. In: *Geoderma* 352, pp. 395–413 (Cited on page 126).

Laplace, P. S. (1814). *Philosophical essay on probabilities*. fre. Paris: Courcier. URL: `http://eudml.org/doc/203193` (Cited on page 17).

Laplace, P. S. de (1820). *Théorie analytique des probabilités*. Vol. 7. Courcier (Cited on page 20).

Lark, R. (2000). ‘Estimating variograms of soil properties by the method-of-moments and maximum likelihood’. In: *European Journal of Soil Science* 51.4, pp. 717–728 (Cited on page 101).

Lark, R. (2012). ‘Towards soil geostatistics’. In: *Spatial Statistics* 1, pp. 92–99 (Cited on page 101).

Laubscher, R. and Rousseau, P. (2022). ‘Application of a mixed variable physics-informed neural network to solve the incompressible steady-state and transient mass, momentum, and energy conservation equations for flow over in-line heated tubes’. In: *Applied Soft Computing* 114, p. 108050 (Cited on page 217).

Le Cam, L. (1990). ‘Maximum likelihood: an introduction’. In: *International Statistical Review/Revue Internationale de Statistique*, pp. 153–171 (Cited on page 20).

LeCun, Y., Bengio, Y. and Hinton, G. (2015). ‘Deep learning’. In: *Nature* 521.7553, pp. 436–444 (Cited on page 128).

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J. and Sohl-Dickstein, J. (2017). ‘Deep neural networks as gaussian processes’. In: *arXiv preprint arXiv:1711.00165* (Cited on pages 69 and 120).

Lee, P. M. (1989). *Bayesian statistics*. Oxford University Press London: (Cited on page 22).

Lei, N., An, D., Guo, Y., Su, K., Liu, S., Luo, Z., Yau, S.-T. and Gu, X. (2020). ‘A geometric understanding of deep learning’. In: *Engineering* 6.3, pp. 361–374 (Cited on page 123).

LeNail, A. (2019). ‘NN-SVG: Publication-ready neural network architecture schematics’. In: *Journal of Open Source Software* 4.33, p. 747 (Cited on pages 127 and 139).

Leung, F. H.-F., Lam, H.-K., Ling, S.-H. and Tam, P. K.-S. (2003). ‘Tuning of the structure and parameters of a neural network using an improved genetic algorithm’. In: *IEEE Transactions on Neural networks* 14.1, pp. 79–88 (Cited on page 133).

Leutbecher, M. and Palmer, T. N. (2008). 'Ensemble forecasting'. In: *Journal of computational physics* 227.7, pp. 3515–3539 (Cited on page 184).

Li, J., Heap, A. D., Potter, A. and Daniell, J. J. (2011). 'Application of machine learning methods to spatial interpolation of environmental variables'. In: *Environmental Modelling & Software* 26.12, pp. 1647–1659 (Cited on page 105).

Li, T., Shen, H., Yuan, Q., Zhang, X. and Zhang, L. (2017). 'Estimating ground-level PM2. 5 by fusing satellite and station observations: a geo-intelligent deep learning approach'. In: *Geophysical Research Letters* 44.23, pp. 11–985 (Cited on page 128).

Li, Y., Sun, Y. and Reich, B. J. (2020). 'DeepKriging: Spatially Dependent Deep Neural Networks for Spatial Prediction'. In: *arXiv preprint arXiv:2007.11972* (Cited on page 134).

Lichtendahl Jr, K. C., Grushka-Cockayne, Y. and Winkler, R. L. (2013). 'Is it better to average probabilities or quantiles?' In: *Management Science* 59.7, pp. 1594–1611 (Cited on page 88).

Lindsay, M. D., Aillères, L., Jessell, M. W., Kemp, E. A. de and Betts, P. G. (2012). 'Locating and quantifying geological uncertainty in three-dimensional models: Analysis of the Gippsland Basin, southeastern Australia'. In: *Tectonophysics* 546, pp. 10–27 (Cited on page 213).

Liu, H., Ong, Y.-S., Shen, X. and Cai, J. (2020). 'When Gaussian process meets big data: A review of scalable GPs'. In: *IEEE Transactions on Neural Networks and Learning Systems* (Cited on page 120).

Lorenc, A. C. (Oct. 1986). 'Analysis methods for numerical weather prediction'. en. In: *Quarterly Journal of the Royal Meteorological Society* 112.474, pp. 1177–1194. ISSN: 00359009, 1477870X. DOI: `10.1002/qj.49711247414`. URL: `http://doi.wiley.com/10.1002/qj.49711247414` (visited on 23/10/2018) (Cited on page 183).

Lorenc, A. C. and Rawlins, F. (2005). 'Why does 4D-Var beat 3D-Var?' In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric* *sciences, applied meteorology and physical oceanography* 131.613, pp. 3247–3257 (Cited on page 64).

Lorenz, E. N. (1963). 'Deterministic nonperiodic flow'. In: *Journal of atmospheric sciences* 20.2, pp. 130–141 (Cited on pages 17 and 18).

Luo, W., Li, Y., Urtasun, R. and Zemel, R. (2016). 'Understanding the effective receptive field in deep convolutional neural networks'. In: *Advances in neural information processing systems*, pp. 4898–4906 (Cited on page 132).

Lynch, P. (2008). 'The origins of computer weather prediction and climate modeling'. In: *J. Comput. Physics* 227, pp. 3431–3444. DOI: `10.1016/j.jcp.2007.02.034` (Cited on pages 53, 54, 57 and 65).

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G. and Johnson, B. A. (2019). 'Deep learning in remote sensing applications: A meta-analysis and review'. In: *ISPRS journal of photogrammetry and remote sensing* 152, pp. 166–177 (Cited on page 105).

MacKay, D. J. (1995). 'Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks'. In: *Network: computation in neural systems* 6.3, pp. 469–505 (Cited on page 177).

Matheron, G. (1969). *Le krigeage universel: cahiers du Centre de Morphologie Mathematique*. École nationale supérieure des mines de Paris (Cited on pages 66 and 101).

Matheron, G. (1962). *Traité de géostatistique appliquée*. Mémoires du Bureau de Recherches Géologiques et Minières. Éditions Technip (Cited on page 130).

Meinshausen, N. (2006). 'Quantile regression forests'. In: *Journal of Machine Learning Research* 7.Jun, pp. 983–999 (Cited on pages 76, 80 and 82).

Menabde, M., Froyland, G., Stone, P. and Yeates, G. (2018). 'Mining schedule optimisation for conditionally simulated orebodies'. In: *Advances in applied strategic mine planning*. Springer, pp. 91–100 (Cited on page 159).

Met Office (2022). *Embedding machine learning and artificial intelligence in weather and climate science and services: a framework for data science in the Met Office (2022-2027)*. URL: `https://www.metoffice.gov.uk/binaries/con` tent/assets/metofficegovuk/pdf/research/foundation-science/data-science-framework-2022-2027.pdf (Cited on page 56).

Miller, G. F., Todd, P. M. and Hegde, S. U. (1989). ‘Designing Neural Networks Using Genetic Algorithms.’ In: *ICGA*. Vol. 89, pp. 379–384 (Cited on page 133).

Miller, R. C. (1967). *Notes on analysis and severe-storm forecasting procedures of the Military Weather Warning Center*. Vol. 200. Air Weather Service (MAC), United States Air Force (Cited on page 52).

Milodowski, D., Mudd, S. and Mitchard, E. (2015). ‘Topographic roughness as a signature of the emergence of bedrock in eroding landscapes’. In: *Earth Surface Dynamics* 3.4, pp. 483–499 (Cited on page 103).

Mohan, A. T., Lubbers, N., Livescu, D. and Chertkov, M. (2020). ‘Embedding hard physical constraints in neural network coarse-graining of 3d turbulence’. In: *arXiv preprint arXiv:2002.00021* (Cited on page 217).

Mosegaard, K. and Tarantola, A. (1995). ‘Monte Carlo sampling of solutions to inverse problems’. In: *Journal of Geophysical Research: Solid Earth* 100.B7, pp. 12431–12447 (Cited on page 129).

Mulder, V., De Bruin, S., Schaepman, M. E. and Mayr, T. (2011). ‘The use of remote sensing in soil and terrain mapping—A review’. In: *Geoderma* 162.1-2, pp. 1–19 (Cited on page 126).

Murphy, A. H. (1966). ‘A note on the utility of probabilistic predictions and the probability score in the cost-loss ratio decision situation’. In: *Journal of Applied Meteorology (1962-1982)* 5.4, pp. 534–537 (Cited on page 33).

Murphy, A. H. (1993). ‘What is a good forecast? An essay on the nature of goodness in weather forecasting’. In: *Weather and forecasting* 8.2, pp. 281–293 (Cited on page 37).

Napoly, A., Grassmann, T., Meier, F. and Fenner, D. (2018). ‘Development and application of a statistically-based quality control for crowdsourced air temperature data’. In: *Frontiers in Earth Science* 6, p. 118 (Cited on page 167).

Neal, R. M. (1995). ‘Bayesian Learning For Neural Networks’. PhD thesis. University of Toronto (Cited on pages 69 and 120).

Neal, R. M. (1996). 'Priors for infinite networks'. In: *Bayesian Learning for Neural Networks*. Springer, pp. 29–53 (Cited on pages 131 and 134).

Neal, R. M. (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media (Cited on page 177).

Nelder, J. A. and Wedderburn, R. W. (1972). 'Generalized linear models'. In: *Journal of the Royal Statistical Society: Series A (General)* 135.3, pp. 370–384 (Cited on page 68).

NERC (2022). *NERC digital strategy 2021 to 2030*. URL: `https://www.ukri.org/wp-content/uploads/2022/05/NERC-170522-NERCDigitalStrategy-FINAL-WEB.pdf` (Cited on page 56).

Nesa, N., Ghosh, T. and Banerjee, I. (2018). 'Outlier detection in sensed data using statistical learning models for IoT'. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, pp. 1–6 (Cited on page 167).

Nundloll, V., Porter, B., Blair, G. S., Emmett, B., Cosby, J., Jones, D. L., Chadwick, D., Winterbourn, B., Beattie, P., Dean, G. et al. (2019). 'The design and deployment of an end-to-end IoT infrastructure for the natural environment'. In: *Future Internet* 11.6, p. 129 (Cited on page 166).

O'Gorman, P. A. and Dwyer, J. G. (2018). *Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events*. en-GB. DOI: `10.1029/2018MS001351`. URL: `https://www.mendeley.com/catalogue/using-machine-learning-parameterize-moist-convection-potential-modeling-climate-climate-change-extre-2/` (visited on 07/02/2019) (Cited on page 58).

Odeh, I. O., McBratney, A. and Chittleborough, D. (1995). 'Further results on prediction of soil properties from terrain attributes: heterotopic cokriging and regression-kriging'. In: *Geoderma* 67.3-4, pp. 215–226 (Cited on page 101).

Olierook, H. K., Scalzo, R., Kohn, D., Chandra, R., Farahbakhsh, E., Clark, C., Reddy, S. M. and Müller, R. D. (2021). 'Bayesian geological and geophysical data fusion for the construction and uncertainty quantification of 3D geological models'. In: *Geoscience Frontiers* 12.1, pp. 479–493 (Cited on page 130).

Padarian, J., Minasny, B. and McBratney, A. B. (2019). ‘Using deep learning for digital soil mapping’. In: *Soil* 5.1, pp. 79–89 (Cited on pages 106, 128 and 174).

Palmer, T. (2017). ‘The primacy of doubt: Evolution of numerical weather prediction from determinism to probability’. In: *Journal of Advances in Modeling Earth Systems* 9.2, pp. 730–734 (Cited on pages 31 and 41).

Palmer, T. and Richardson, D. (2014). ‘Decisions, decisions. . . !’ In: (141), pp. 12–14. DOI: `10.21957/bychj3cf`. URL: `https://www.ecmwf.int/node/17333` (Cited on pages 33 and 37).

Palmer, T. N., Alessandri, A., Andersen, U., Cantelaube, P., Davey, M., Delécluse, P., Déqué, M., Diez, E., Doblas-Reyes, F. J., Feddersen, H. et al. (2004). ‘Development of a European multimodel ensemble system for seasonal-to-interannual prediction (DEMETER)’. In: *Bulletin of the American Meteorological Society* 85.6, pp. 853–872 (Cited on page 182).

Parmentier, B., McGill, B., Wilson, A. M., Regetz, J., Jetz, W., Guralnick, R. P., Tuanmu, M.-N., Robinson, N. and Schildhauer, M. (2014). ‘An assessment of methods and remote-sensing derived covariates for regional predictions of 1 km daily maximum air temperature’. In: *Remote Sensing* 6.9, pp. 8639–8670 (Cited on page 126).

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K. et al. (2022). ‘Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators’. In: *arXiv preprint arXiv:2202.11214* (Cited on page 56).

Pawlowsky-Glahn, V., Egozcue, J. J. and Tolosana-Delgado, R. (2015). *Modeling and analysis of compositional data*. John Wiley & Sons (Cited on page 107).

Perol, T., Gharbi, M. and Denolle, M. (2018). ‘Convolutional neural network for earthquake detection and location’. In: *Science Advances* 4.2, e1700578 (Cited on page 128).

Pilz, J. and Spöck, G. (2008). ‘Why do we need and how should we implement Bayesian kriging methods’. In: *Stochastic Environmental Research and Risk Assessment* 22.5, pp. 621–632 (Cited on page 127).

Poggio, L., Gimona, A. and Brewer, M. J. (2013). 'Regional scale mapping of soil properties and their uncertainty with a large number of satellite-derived covariates'. In: *Geoderma* 209, pp. 1–14 (Cited on page 126).

Pourghasemi, H. R. and Rahmati, O. (2018). 'Prediction of the landslide susceptibility: Which algorithm, which precision?' In: *Catena* 162, pp. 177–192 (Cited on page 105).

Pratt, J. W., Raiffa, H., Schlaifer, R. et al. (1995). *Introduction to statistical decision theory*. MIT press (Cited on page 31).

Quiñonero-Candela, J., Rasmussen, C. E., Sinz, F., Bousquet, O. and Schölkopf, B. (2006). 'Evaluating Predictive Uncertainty Challenge'. en. In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 1–27. ISBN: 978-3-540-33428-6. DOI: 10.1007/11736790_1 (Cited on pages 89 and 90).

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/ (Cited on page 144).

Raatikainen, P. et al. (2020). 'Remarks on the Gödelian Anti-Mechanist Arguments'. In: *Studia Semiotyczne* 34.1, pp. 267–278 (Cited on page 18).

Raftery, A. E., Gneiting, T., Balabdaoui, F. and Polakowski, M. (2005). 'Using Bayesian Model Averaging to Calibrate Forecast Ensembles'. en. In: *Monthly Weather Review* 133, p. 20 (Cited on pages 47, 58, 60, 61, 62, 77, 87 and 184).

Raftery, A. E. and Lewis, S. M. (1996). 'Implementing mcmc'. In: *Markov chain Monte Carlo in practice*, pp. 115–130 (Cited on page 142).

Rahmstorf, S. and Coumou, D. (Oct. 2011). 'Increase of extreme events in a warming world'. en. In: *Proceedings of the National Academy of Sciences*. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1101766108. URL: https://www.pnas.org/content/early/2011/10/18/1101766108 (visited on 24/04/2020) (Cited on page 74).

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K. et al. (2017). 'Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning'. In: *arXiv preprint arXiv:1711.05225* (Cited on page 121).

Rasp, S. and Lerch, S. (Oct. 2018). 'Neural Networks for Postprocessing Ensemble Weather Forecasts'. In: *Monthly Weather Review* 146.11, pp. 3885–3900. ISSN: 0027-0644. DOI: 10.1175/MWR-D-18-0187.1. URL: https://journals.ametsoc.org/doi/full/10.1175/MWR-D-18-0187.1 (visited on 27/04/2020) (Cited on pages 15, 63 and 75).

Rasp, S., Pritchard, M. S. and Gentine, P. (Sept. 2018). 'Deep learning to represent subgrid processes in climate models'. en. In: *Proceedings of the National Academy of Sciences* 115.39, pp. 9684–9689. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1810286115. URL: https://www.pnas.org/content/115/39/9684 (visited on 05/02/2019) (Cited on page 57).

Ratcliff, R. (1979). 'Group reaction time distributions and an analysis of distribution statistics'. In: *Psychological Bulletin* 86.3, pp. 446–461. ISSN: 1939-1455(Electronic),0033-2909(Print). DOI: 10.1037/0033-2909.86.3.446 (Cited on page 88).

Rawlins, F., Ballard, S., Bovis, K., Clayton, A., Li, D., Inverarity, G., Lorenc, A. and Payne, T. (2007). 'The Met Office global four-dimensional variational data assimilation scheme'. In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 133.623, pp. 347–362 (Cited on page 184).

Ren, Y., Zhang, L. and Suganthan, P. (Feb. 2016). 'Ensemble Classification and Regression-Recent Developments, Applications and Future Directions [Review Article]'. In: *IEEE Computational Intelligence Magazine* 11.1, pp. 41–53. ISSN: 1556-6048. DOI: 10.1109/MCI.2015.2471235 (Cited on pages 88 and 96).

Richardson, L. F. (1922). *Weather Prediction by Numerical Process*. en. Cambridge University Press. ISBN: 978-0-521-68044-8 (Cited on pages 54 and 74).

Richardson, L. F. and Glazebrook, R. T. (Jan. 1911). ‘The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam’. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 210.459-470, pp. 307–357. DOI: `10.1098/rsta.1911.0009`. URL: `https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1911.0009` (visited on 08/02/2019) (Cited on page 54).

Riley, S. J., DeGloria, S. D. and Elliot, R. (1999). ‘Index that quantifies topographic heterogeneity’. In: *intermountain Journal of sciences* 5.1-4, pp. 23–27 (Cited on page 103).

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W. et al. (2017). ‘Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure’. In: *Ecography* 40.8, pp. 913–929 (Cited on pages 25 and 146).

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M. and Chica-Rivas, M. (2015). ‘Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines’. In: *Ore Geology Reviews* 71, pp. 804–818 (Cited on pages 105 and 116).

Rougier, J. and Beven, K. J. (2013). ‘Model and data limitations: the sources and implications of epistemic uncertainty’. In: *Risk and uncertainty assessment for natural hazards* 40 (Cited on page 184).

Rougier, J. (2013). ‘‘Intractable and unsolved’: some thoughts on statistical data assimilation with uncertain static parameters’. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1991, p. 20120297 (Cited on page 184).

Rubin, D. B. (1984). 'Bayesianly justifiable and relevant frequency calculations for the applies statistician'. In: *The Annals of Statistics*, pp. 1151–1172 (Cited on page 27).

Ruiz-Arias, J., Pozo-Vázquez, D., Santos-Alamillos, F., Lara-Fanego, V. and Tovar-Pescador, J. (2011). 'A topographic geostatistical approach for mapping monthly mean values of daily global solar radiation: A case study in southern Spain'. In: *Agricultural and forest meteorology* 151.12, pp. 1812–1822 (Cited on page 126).

Sabins, F. F. (1999). 'Remote sensing for mineral exploration'. In: *Ore geology reviews* 14.3-4, pp. 157–183 (Cited on page 153).

Sahu, S. K. (2022). *Bayesian modeling of spatio-temporal data with R*. Chapman and Hall/CRC (Cited on pages 120 and 213).

Sambridge, M. and Mosegaard, K. (2002). 'Monte Carlo methods in geophysical inverse problems'. In: *Reviews of Geophysics* 40.3, pp. 3–1 (Cited on page 129).

Saxe, A., Nelli, S. and Summerfield, C. (2021). 'If deep learning is the answer, what is the question?' In: *Nature Reviews Neuroscience* 22.1, pp. 55–67 (Cited on page 123).

Schaaf, A., Varga, M. de la, Wellmann, F. and Bond, C. E. (2020). 'Constraining stochastic 3-D structural geological models with topology information using Approximate Bayesian Computation using GemPy 2.1'. In: *Geoscientific Model Development Discussions*, pp. 1–24 (Cited on page 130).

Schepen, A. and Wang, Q. J. (2015). 'Model averaging methods to merge operational statistical and dynamic seasonal streamflow forecasts in Australia'. en. In: *Water Resources Research* 51.3, pp. 1797–1812. ISSN: 1944-7973. DOI: `10.1002/2014WR016163`. URL: `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014WR016163` (visited on 06/03/2020) (Cited on page 86).

Shamseldin, A. Y., O'Connor, K. M. and Liang, G. C. (Oct. 1997). 'Methods for combining the outputs of different rainfall–runoff models'. en. In: *Journal of Hydrology* 197.1, pp. 203–229. ISSN: 0022-1694. DOI: `10.1016/S0022-1694(9`

6)03259-3. URL: http://www.sciencedirect.com/science/article/pii/S0022169496032593 (visited on 09/03/2020) (Cited on page 63).

Shamsipour, P., Schetselaar, E., Bellefleur, G. and Marcotte, D. (2014). '3D stochastic inversion of potential field data using structural geologic constraints'. In: *Journal of Applied Geophysics* 111, pp. 173–182 (Cited on page 126).

Shwartz-Ziv, R. and Tishby, N. (2017). 'Opening the black box of deep neural networks via information'. In: *arXiv preprint arXiv:1703.00810* (Cited on page 126).

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. et al. (Oct. 2017). 'Mastering the game of Go without human knowledge'. en. In: *Nature* 550.7676, pp. 354–359. ISSN: 1476-4687. DOI: 10.1038/nature24270. URL: https://www.nature.com/articles/nature24270%3E (visited on 24/04/2020) (Cited on page 75).

Simpson, M., James, R., Hall, J. W., Borgomeo, E., Ives, M. C., Almeida, S., Kingsborough, A., Economou, T., Stephenson, D. and Wagener, T. (Oct. 2016). 'Decision Analysis for Management of Natural Hazards'. In: *Annual Review of Environment and Resources* 41.1, pp. 489–516. ISSN: 1543-5938. DOI: 10.1146/annurev-environ-110615-090011. URL: https://www.annualreviews.org/doi/10.1146/annurev-environ-110615-090011 (visited on 20/02/2020) (Cited on page 75).

Smith, E. (2022). *Making Decisions: Putting the Human Back in the Machine*. HarperCollins Publishers. ISBN: 9780008530167. URL: https://books.google.co.uk/books?id=pkdnEAAAQBAJ (Cited on page 31).

Smith, R. C. (2013). *Uncertainty quantification: theory, implementation, and applications*. Vol. 12. Siam (Cited on page 154).

Snelson, E. and Ghahramani, Z. (2005). 'Sparse Gaussian processes using pseudo-inputs'. In: *Advances in neural information processing systems* 18 (Cited on page 69).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). 'Dropout: a simple way to prevent neural networks from overfitting'.

In: *The journal of machine learning research* 15.1, pp. 1929–1958 (Cited on pages 127, 140 and 143).

Stambaugh, M. C. and Guyette, R. P. (2008). 'Predicting spatio-temporal variability in fire return intervals using a topographic roughness index'. In: *Forest Ecology and Management* 254.3, pp. 463–473 (Cited on page 103).

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging.* Springer Science & Business Media (Cited on pages 101 and 126).

Stephens, E. and Cloke, H. (Dec. 2014). 'Improving flood forecasts for better flood preparedness in the UK (and beyond)'. In: *The Geographical Journal* 180.4, pp. 310–316. ISSN: 0016-7398. DOI: 10.1111/geoj.12103. URL: https://rgs-ibg.onlinelibrary.wiley.com/doi/10.1111/geoj.12103 (visited on 10/04/2020) (Cited on page 76).

Taillardat, M. and Mestre, O. (Jan. 2020). 'From research to applications - Examples of operational ensemble post-processing in France using machine learning'. English. In: *Nonlinear Processes in Geophysics Discussions*, pp. 1–27. ISSN: 1023-5809. DOI: https://doi.org/10.5194/npg-2019-65. URL: https://www.nonlin-processes-geophys-discuss.net/npg-2019-65/ (visited on 28/02/2020) (Cited on pages 79 and 96).

Taillardat, M., Mestre, O., Zamo, M. and Naveau, P. (June 2016). 'Calibrated Ensemble Forecasts Using Quantile Regression Forests and Ensemble Model Output Statistics'. en. In: *Monthly Weather Review* 144.6, pp. 2375–2393. ISSN: 0027-0644, 1520-0493. DOI: 10.1175/MWR-D-15-0260.1. URL: http://journals.ametsoc.org/doi/10.1175/MWR-D-15-0260.1 (visited on 23/10/2018) (Cited on pages 63, 75, 81 and 96).

Tarantola, A., Valette, B. et al. (1982). 'Inverse problems= quest for information'. In: *Journal of geophysics* 50.1, pp. 159–170 (Cited on page 129).

Tian, D. p. (2013). 'A review on image feature extraction and representation techniques'. In: *International Journal of Multimedia and Ubiquitous Engineering* 8.4, pp. 385–396. ISSN: 1975-0080 (Cited on page 51).

Tobler, W. R. (1970). ‘A computer movie simulating urban growth in the Detroit region’. In: *Economic geography* 46.sup1, pp. 234–240 (Cited on page 101).

Van Zyl, J. J. (2001). ‘The Shuttle Radar Topography Mission (SRTM): a breakthrough in remote sensing of topography’. In: *Acta Astronautica* 48.5-12, pp. 559–565 (Cited on pages 107 and 144).

Vann, J., Bertoli, O. and Jackson, S. (2002). ‘An overview of geostatistical simulation for quantifying risk’. In: *Proceedings of Geostatistical Association of Australasia Symposium” Quantifying Risk and Error*. Vol. 1. Citeseer, p. 1 (Cited on page 159).

Varga, M. de la and Wellmann, J. F. (2016). ‘Structural geologic modeling as an inference problem: A Bayesian perspective’. In: *Interpretation* 4.3, SM1–SM16 (Cited on pages 130 and 213).

Venter, Z. S., Brousse, O., Esau, I. and Meier, F. (2020). ‘Hyperlocal mapping of urban air temperature using remote sensing and crowdsourced weather data’. In: *Remote Sensing of Environment* 242, p. 111791 (Cited on page 168).

Vincent, S. (1912). ‘The function of the vibrissae in the behavior of the white rat’. In: *Animal Behavior Monographs* 1, 5, pp. 84–84 (Cited on page 87).

Vladimirova, M., Verbeek, J., Mesejo, P. and Arbel, J. (2019). ‘Understanding priors in bayesian neural networks at the unit level’. In: *International Conference on Machine Learning*. PMLR, pp. 6458–6467 (Cited on page 23).

Wadhams, G. H. and Armitage, J. P. (2004). ‘Making sense of it all: bacterial chemotaxis’. In: *Nature reviews Molecular cell biology* 5.12, pp. 1024–1037 (Cited on page 16).

Wadoux, A. M. J., Padarian, J. and Minasny, B. (2019). ‘Multi-source data integration for soil mapping using deep learning’. In: *Soil* 5.1, pp. 107–119 (Cited on pages 106 and 128).

Wadoux, A. M.-C. (2019). ‘Using deep learning for multivariate mapping of soil with quantified uncertainty’. In: *Geoderma* 351, pp. 59–70 (Cited on pages 106, 129 and 174).

Wadoux, A. M.-C., Heuvelink, G. B., De Bruin, S. and Brus, D. J. (2021). 'Spatial cross-validation is not the right way to evaluate map accuracy'. In: *Ecological Modelling* 457, p. 109692 (Cited on pages 25, 26 and 147).

Wang, H., Bah, M. J. and Hammad, M. (2019). 'Progress in outlier detection techniques: A survey'. In: *Ieee Access* 7, pp. 107964–108000 (Cited on page 167).

Wang, R., Kashinath, K., Mustafa, M., Albert, A. and Yu, R. (2020). 'Towards physics-informed deep learning for turbulent flow prediction'. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466 (Cited on page 217).

Wasserstein, R. (2010). 'George Box: a model statistician'. In: *Significance* 7.3, pp. 134–135. DOI: https://doi.org/10.1111/j.1740-9713.2010.00442.x. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1740-9713.2010.00442.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2010.00442.x (Cited on page 41).

Wellmann, J. F., De La Varga, M., Murdie, R. E., Gessner, K. and Jessell, M. (2018). 'Uncertainty estimation for a geological model of the Sandstone greenstone belt, Western Australia–insights from integrated geological and geophysical inversion in a Bayesian inference framework'. In: *Geological Society, London, Special Publications* 453.1, pp. 41–56 (Cited on pages 130 and 213).

Weyn, J. A., Durran, D. R., Caruana, R. and Cresswell-Clay, N. (2021). 'Subseasonal forecasting with a large ensemble of deep-learning weather prediction models'. In: *Journal of Advances in Modeling Earth Systems* 13.7, e2021MS002502 (Cited on page 56).

White, C., Neiswanger, W. and Savani, Y. (2021). 'Bananas: Bayesian optimization with neural architectures for neural architecture search'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12, pp. 10293–10301 (Cited on page 133).

Wilford, J., Caritat, P. de and Bui, E. (2016). 'Predictive geochemical mapping using environmental correlation'. In: *Applied geochemistry* 66, pp. 275–288 (Cited on page 112).

Wilks, D. S. and Hamill, T. M. (June 2007). 'Comparison of Ensemble-MOS Methods Using GFS Reforecasts'. In: *Monthly Weather Review* 135.6, pp. 2379–2390. ISSN: 0027-0644. DOI: 10.1175/MWR3402.1. URL: https://journals.ametsoc.org/doi/full/10.1175/MWR3402.1 (visited on 21/02/2020) (Cited on page 77).

Willard, J., Jia, X., Xu, S., Steinbach, M. and Kumar, V. (2022). 'Integrating scientific knowledge with machine learning for engineering and environmental systems'. In: *ACM Computing Surveys* 55.4, pp. 1–37 (Cited on page 217).

Wilson, A. G. (2020). 'The case for Bayesian deep learning'. In: *arXiv preprint arXiv:2001.10995* (Cited on pages 69, 141 and 219).

Wilson, A. G., Hu, Z., Salakhutdinov, R. and Xing, E. P. (2016). 'Deep kernel learning'. In: *Artificial intelligence and statistics*. PMLR, pp. 370–378 (Cited on pages 69, 70 and 219).

Wright, M. N. and Ziegler, A. (2017). 'ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R'. In: *Journal of Statistical Software* 77.1. ISSN: 1548-7660. DOI: 10.18637/jss.v077.i01. URL: http://arxiv.org/abs/1508.04409 (visited on 31/10/2019) (Cited on page 80).

Wu, J.-L., Kashinath, K., Albert, A., Chirila, D., Xiao, H. et al. (2020). 'Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems'. In: *Journal of Computational Physics* 406, p. 109209 (Cited on page 217).

Wu, X., Liang, L., Shi, Y. and Fomel, S. (2019). 'FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation'. In: *Geophysics* 84.3, pp. IM35–IM45 (Cited on page 128).

Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (Sept. 2018). 'Using Stacking to Average Bayesian Predictive Distributions (with Discussion)'. EN. In: *Bayesian Analysis* 13.3, pp. 917–1007. ISSN: 1936-0975, 1931-6690. DOI: 10.1214/17-BA1091. URL: https://projecteuclid.org/euclid.ba/1516093227 (visited on 04/11/2019) (Cited on page 96).

Yoe, C. (2011). *Principles of risk analysis: decision making under uncertainty*. CRC press (Cited on page 136).

Young, D. M., Parry, L. E., Lee, D. and Ray, S. (2018). 'Spatial models with covariates improve estimates of peat depth in blanket peatlands'. In: *PLoS ONE* 13.9 (Cited on page 126).

Youssef, A. M., Pourghasemi, H. R., Pourtaghi, Z. S. and Al-Katheeri, M. M. (2016). 'Landslide susceptibility mapping using random forest, boosted regression tree, classification and regression tree, and general linear models and comparison of their performance at Wadi Tayyah Basin, Asir Region, Saudi Arabia'. In: *Landslides* 13.5, pp. 839–856 (Cited on page 126).

Zaherpour, J., Mount, N. J., Gosling, S. N., Dankers, R., Eisner, S., Dieter, G., Liu, X., Masaki, Y., Müller Schmied, H. and Tang, Q. (2019). 'Exploring the value of machine learning for weighted multi-model combination of an ensemble of global hydrological models'. In: *Environmental Modelling and Software*. ISSN: 1873-6726 (Cited on page 63).

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2021). 'Understanding deep learning (still) requires rethinking generalization'. In: *Communications of the ACM* 64.3, pp. 107–115 (Cited on page 123).

Zhang, L., Zhang, L. and Du, B. (2016). 'Deep learning for remote sensing data: A technical tutorial on the state of the art'. In: *IEEE Geoscience and Remote Sensing Magazine* 4.2, pp. 22–40 (Cited on pages 105, 127 and 128).

Zhang, Y., Wang, J. and Wang, X. (2014). 'Review on probabilistic forecasting of wind power generation'. In: *Renewable and Sustainable Energy Reviews* 32, pp. 255–270 (Cited on page 52).

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F. and Fraundorfer, F. (2017). 'Deep learning in remote sensing: A comprehensive review and list of resources'. In: *IEEE Geoscience and Remote Sensing Magazine* 5.4, pp. 8–36 (Cited on pages 105, 127 and 128).

Zou, Q., Ni, L., Zhang, T. and Wang, Q. (2015). ‘Deep learning based feature selection for remote sensing scene classification’. In: *IEEE Geoscience and Remote Sensing Letters* 12.11, pp. 2321–2325 (Cited on page 105).

Zumwald, M., Knüsel, B., Bresch, D. N. and Knutti, R. (2021). ‘Mapping urban temperature using crowd-sensing data and machine learning’. In: *Urban Climate* 35, p. 100739 (Cited on page 168).

Zuo, R. (2017). ‘Machine learning of mineralization-related geochemical anomalies: A review of potential methods’. In: *Natural Resources Research* 26.4, pp. 457–464 (Cited on page 116).

Zuo, R., Xiong, Y., Wang, J. and Carranza, E. J. M. (2019). ‘Deep learning and its application in geochemical mapping’. In: *Earth-science reviews* 192, pp. 1–14 (Cited on pages 128 and 154).

Appendices

Appendix A

Associated posters

Find here copies of posters presented at various conferences during the PhD. These are:

1. “A probabilistic post-processing framework for blending all available weather forecast” presented at the University of Exeter’s Environmental Intelligence Conference 2020.

2. “Harnessing auxiliary information for probabilistic spatial interpolation using Bayesian deep learning” presented at the University of Exeter’s Environmental Intelligence Conference 2020.

3. “Bayesian deep learning for large scale environmental data modelling” presented at the Alan Turing Institute’s AI UK research showcase 2021.

4. “Harnessing auxiliary information for probabilistic environmental modelling using Bayesian deep learning” presented at the University of Exeter’s Environmental Intelligence Conference 2021.

5. “Deep geostatistics: incorporating computer vision for intelligent Bayesian interpolation between environmental observations” presented at the AI Innovations Summit 2022, Exeter and Berlin.

The posters are targeted to the different types of audiences at these different events. Talks were also given (e.g. to the Met Office Data Science Community of Practice, the Alan Turing Institute’s Environmental Monitoring group, the University of Cambridge’s Environmental Data Science Group) but including the associated slides would hugely extend this thesis, and so I have not.

A probabilistic post-processing framework for blending all available weather forecasts

Problem statement

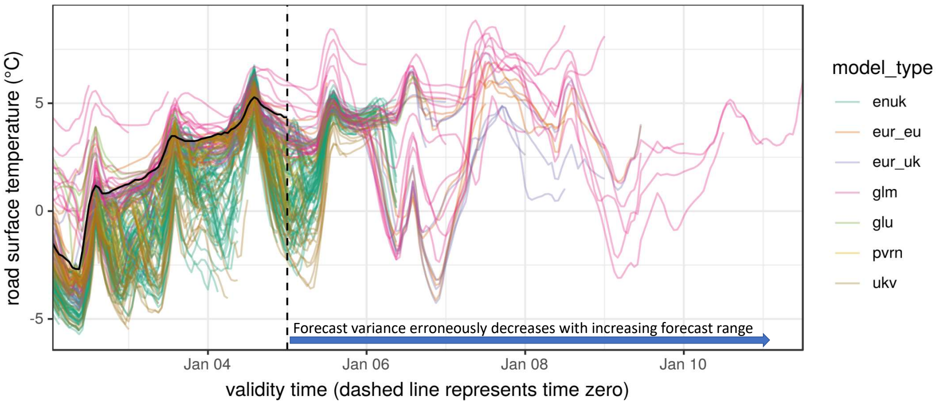
Much effort is placed in the development of ever more sophisticated, high-resolution, and accurate weather forecasting models – both by advancing numerical weather prediction and more recently also by utilising machine learning techniques.

However - regardless of the quality of individual models - with the increasing complexity and number of models in operation, a different challenge emerges: that of assimilating the information contained within the pool of available forecasts in order to provide well-calibrated probabilities on which to base decision making.

For risk-critical applications the task of assimilating information from the forecast pool is currently carried out by expert operational meteorologists, who assess the spread of forecasts produced by different models in order to provide statements about the most likely outcomes and reasonable worst case scenarios.

However, automating this process via the development of an ‘algorithmic interface’ to forecast information has the potential to improve the calibration and consistency of probabilities that are provided to stakeholders – as well as ease of access.

Below: Example of the forecast pool for road surface temperature forecasting. Even in this simple site specific case the pool of different model forecasts is quite overwhelming to human interpreters and difficult to translate into well-calibrated probabilities of future weather outcomes.



Our solution (applied to road surface temperature forecasting)

We propose a framework by which to probabilistically combine the forecasting power of different weather models and their respective ensemble members, in order to automatically produce a ‘best estimate’ predictive distribution of future weather outcomes that has well-calibrated probabilities. Our approach borrows from model stacking concepts in machine learning, and is designed for fast operational use, including compatibility with any type of forecast model.

1. Convert individual deterministic forecasts to probabilistic forecasts by modelling their error profile:

We cannot rely on the inter-model variance of the available forecasts for our uncertainty estimates, because more ensemble members are available at short lead times than long. In our example, inter-model variance decreases to zero at the longest ranges – where we would expect uncertainty to be highest!

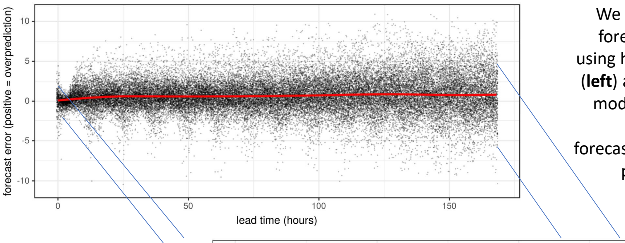
To overcome this, we instead simulate intra-model variance, by modelling the error profile of each forecast:

$$\epsilon_{t,m} = y - x_{t,m}$$

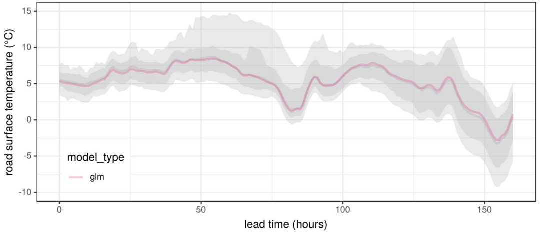
Where y is the observed value corresponding to the prediction of the forecast x by model m at lead time t. We then use these modelled error profiles to ‘dress’ each forecast with its own (intra-model) variance:

$$\hat{y}_{t,m} = x_{t,m} + \epsilon_{t,m}$$

We now have a set of debiased and individually well-calibrated probabilistic forecasts. In this study, we used quantile regression forests (QRF), based on the random forest algorithm, to model these error profiles – mostly due to the high speed and non-parametric flexibility.

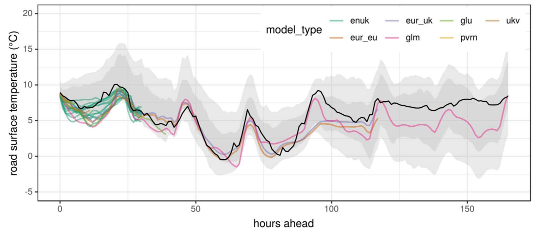


We model each forecast’s error using historic data (**left**) and use this model to ‘dress’ the current forecast, making it probabilistic (**below**)



2. Combine individual probabilistic forecasts by quantile averaging:

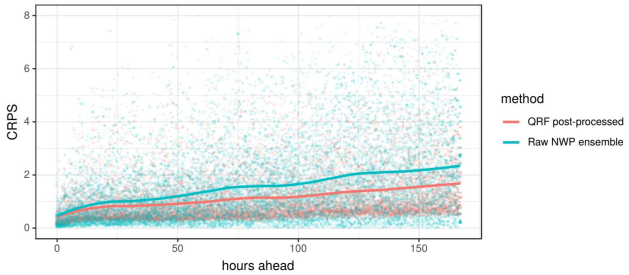
Our set of forecasts are now individually well-calibrated, and we wish to retain this calibration for our final ‘best estimate’ predictive distribution. To do this, we use quantile averaging, which results in an aggregate predictive distribution with mean, variance, and shape all approximately equal to the average mean, variance, and shape of the individual probabilistic forecasts. We now have our ‘best estimate’ predictive distribution (**below**):



3. Results:

Our post-processing framework outperforms the raw ensemble in both probabilistic metrics (coverage, continuous rank probability score (CRPS), log score) and deterministic metrics (mean absolute error, mean squared error), and is more suitable for use in decision support.

We can see from plotting CRPS by forecast range (**right**), that our post-processed output provides more consistent, and consistently better forecasts than the raw ensemble



Charlie Kirkwood, Theo Economou, Henry Odbert, Nicolas Pugeault

UNIVERSITY OF EXETER

correspondence to: c.kirkwood@exeter.ac.uk

Met Office

See more in the full paper – “A framework for probabilistic weather forecast post-processing across models and lead times using machine learning”, Philosophical Transactions of the Royal Society A, 2021

Harnessing auxiliary information for probabilistic spatial interpolation using Bayesian deep learning

1. Problem statement

We are often unable to observe exhaustively the environmental phenomena in which we are interested, and are limited to the collection of point-sampled observations, from which we must interpolate in order to build up a complete picture.

Traditional spatial interpolation techniques (e.g., ordinary kriging, inverse distance weighted (IDW) interpolation) consider only the two or three dimensions of physical space, and as a result produce spatially smooth outputs 'in isolation' which fail to consider any other available information.

In order to capture auxiliary information, established practice is to include a 'regression on covariates' component in the model, as in regression kriging:

$$Z = X\tau + u + \varepsilon,$$

where Z, a random vector corresponding to the target variable at n sites, is predicted as the sum of regression on covariates component $X\tau$ (where X is an n × p design matrix containing the values of any covariates, for which τ are the corresponding coefficients), Gaussian Process spatial component u, and error term ε.

However, it is difficult or even impossible to know a priori what would constitute optimal covariates for a given problem. This issue is especially stark in environmental modelling, where derivatives of satellite imagery and other remotely sensed data have the potential to be highly informative, but attempts to harness this information through manual filtering (e.g., using off-the-shelf terrain analyses) are destined to fall short of optimality.

Developing precise maps, models and forecasts requires capturing the complex interactions between all the environmental factors at play

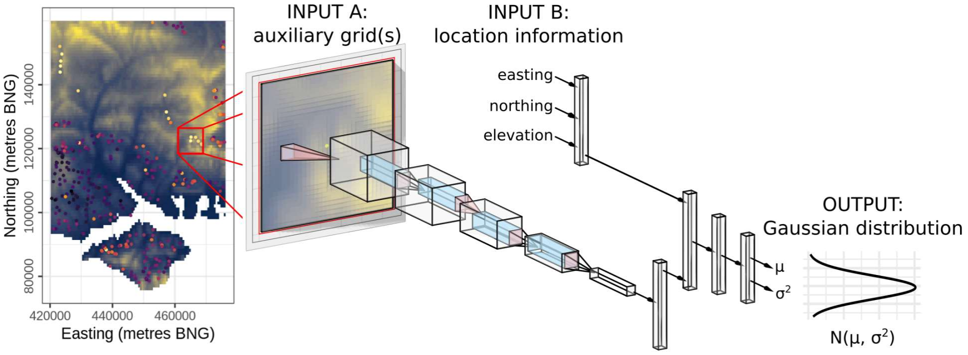
2. Our solution

To avoid the top-down trial and error of manual feature engineering and feature selection, we propose a bottom-up approach to utilising auxiliary information via Bayesian deep learning. Our methodology combines feature learning capabilities drawn from the field of computer vision with probabilistic spatial interpolation capabilities drawn from the field of (Bayesian) geostatistics. In addition, owing to the efficiencies of neural networks and modern deep learning frameworks (Tensorflow and Tensorflow Probability in this case), our method is scalable to very large datasets.

To demonstrate the ability of our approach to model complex environmental phenomena we present a case study using stream sediment calcium concentration as our point-sampled target variable, and SRTM digital elevation data (terrain topography) as our source of auxiliary information. We intend a similar approach for downscaling weather forecasts to continuous space.

3. Experimental setup

We used a 3CV train/evaluate/test data split for our neural network training, with a total of 109 201 point-sampled calcium observations from all across the UK. For data preparation, each calcium observation was appended with a vector of its location information in 3D space (easting, northing, and elevation) and an image of surrounding terrain topography, centred on the observation.

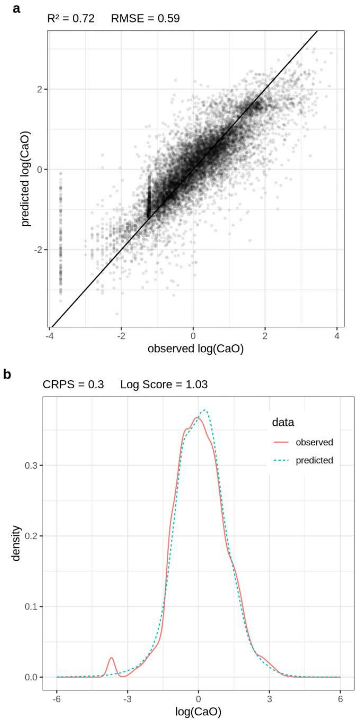


INPUT A: auxiliary grid(s)

INPUT B: location information

OUTPUT: Gaussian distribution

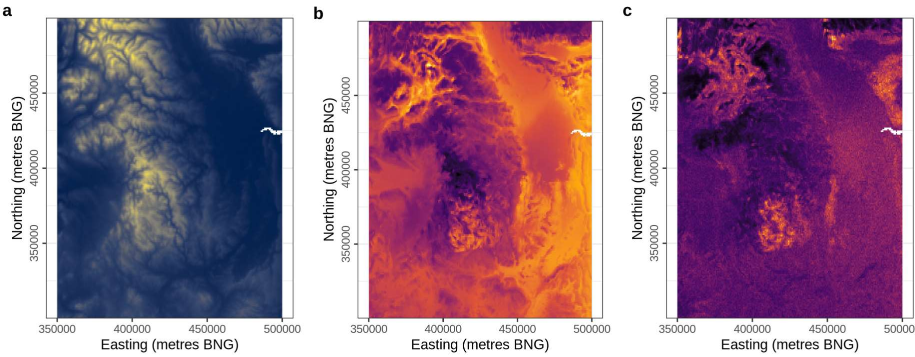
Our neural network architecture (**above**) has two branches – a convolutional branch takes the terrain topography image as input and trains convolutional filters to extract information about the context of the observation within the landscape. Meanwhile, the fully connected branch behaves like (deep) Gaussian Process spatial interpolation, or kriging. Combining these two branches allows our neural network to interpolate not simply in the geographic space, but in a self-constructed hybrid space that optimally combines global location information with local contextual information. *In addition, we use Monte Carlo dropout (Gal and Ghahramani 2015) to provide a tuneable distribution on the network weights, which, in conjunction with our Gaussian likelihood, allows us to estimate uncertainty and approximate a full posterior predictive distribution.*



4. Results

Quantitative (**right**): After training on 8 tenths of the data, we test final performance on the 1 tenth test set (after parameter tuning on the eval set). Our deep learning approach achieves impressive explanatory power for such a complex natural system (R^2 of 0.72) and near-perfect coverage (94.9% of held-out observations were found to fall within the 95% prediction interval)

Qualitative (**below**): As we can see, our neural network makes good use of the auxiliary information within the topography data (**a**) in order to provide detailed predictions of the mean calcium concentration (**b**) and standard deviation of the predictive distribution (**c**) for all points in space. We can see by the precision of (**c**) that our predictive distribution is not just well-calibrated, but sharp too.



Charlie Kirkwood, Theo Economou, Henry Odbert, Nicolas Pugeault

correspondence to: c.kirkwood@exeter.ac.uk

See more in the full paper – preprint available on arXiv – https://arxiv.org/abs/2008.07320

Bayesian deep learning for large scale environmental data modelling

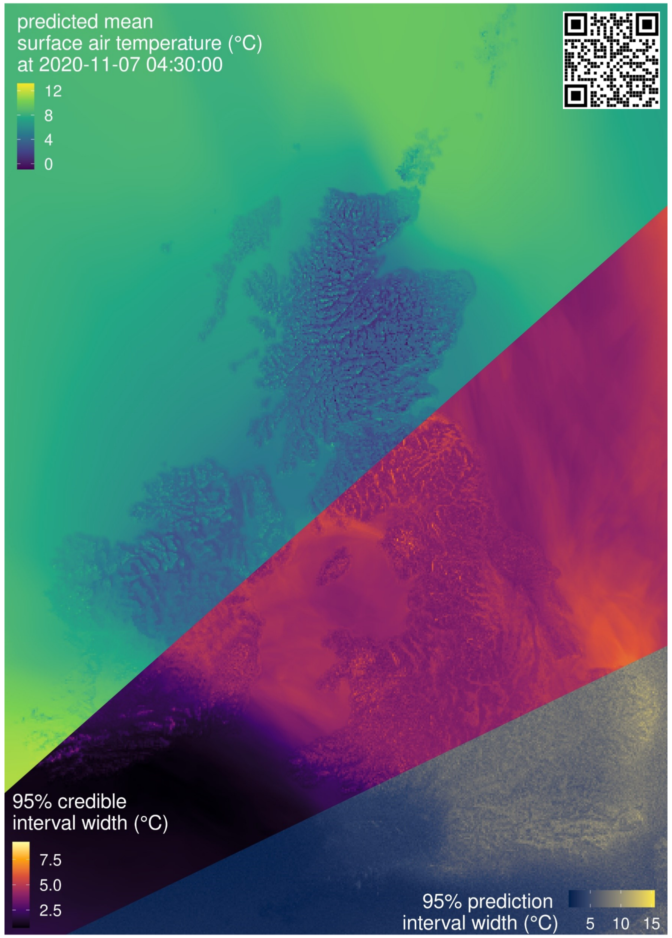
Charlie Kirkwood[1]*, Theo Economou[1], Henry Odbert[2], Nicolas Pugeault[3]

University of Exeter (1), UK Met Office (2), University of Glasgow (3) *contact: c.kirkwood@exeter.ac.uk

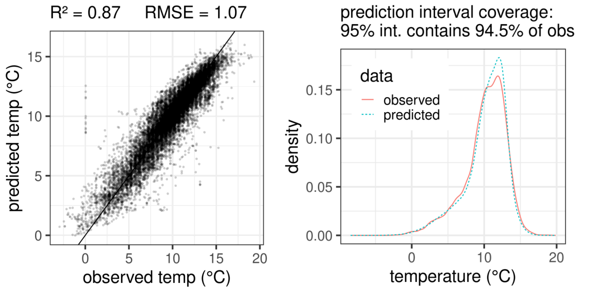
Outcomes

- Deep learning – machine learning using deep neural networks – is an efficient way to discover patterns in data that may be more complex than we could manually hypothesise.
- Here we learn spatio-temporal models that harness information from gridded auxiliary datasets, such as digital terrain models and satellite imagery, by learning task-relevant derivatives of these with no requirement for manual feature engineering.
- By operating within the Bayesian probabilistic framework, we can learn well-calibrated deep models that quantify epistemic and aleatoric uncertainties and avoid overfitting despite the capacity of deep models to do so.
- This allows us to simulate infinite ‘ensemble members’ from the posterior distribution, approximating the range of plausible hypotheses (data generating functions) that can explain the environmental phenomena we observe.
- In addition, we can make data collection more efficient by targeting regions for which the variability between plausible hypotheses – the epistemic uncertainty – is greatest.

Example: modelling air temperature from Internet-of-Things sensors

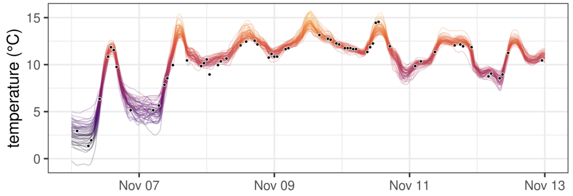


Over 1500 unofficial weather stations across the UK provide over 8000 observations per hour. We interpolate these and assess against observations at held-out locations to reveal the predictive performance:



Above L: deterministic results **Above R:** probabilistic results

Below: 50 samples, or ensemble members, from the posterior



Wide applicability

Making the right decisions requires having good information – including modelled uncertainty.

We can use deep learning as an algorithmic interface to distill information from otherwise overwhelming sources, allowing us to manage the unmanageable.

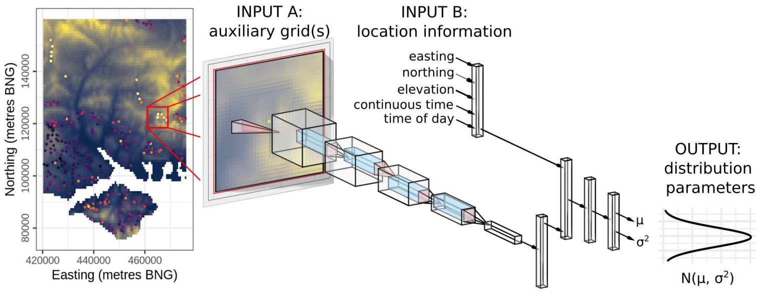
Harnessing auxiliary information for probabilistic environmental modelling using Bayesian deep learning

The concept

With the rise of remote sensing and satellite imagery, we often have auxiliary information available that we would like to make use of in our models, but it is not necessarily clear how to do so.

Traditional approaches, such as regression kriging, tend to require the user to manually engineer and select features to be included within the model. For geospatial applications, this means conducting manual filtering of auxiliary grids e.g. to generate slope angles, roughness, and other derivatives to be input into the model. However, these ‘off the shelf’ features are unlikely to be optimal for the modelling task at hand – and do we really know what would be?

Instead, with our approach, we incorporate computer vision capabilities into the model using deep learning to automatically derive complex filters for extracting relevant information from auxiliary grids. This is achieved directly as part of the process to interpolate point-sampled target variable observations:



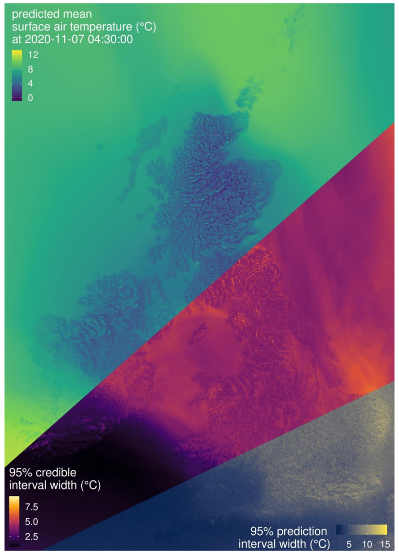
The flexibility of deep learning opens up a richer space of possible model fits compared to traditional approaches. This could lead to ‘overfitting’, but by adopting a Bayesian rather than frequentist approach, we compute the posterior distribution of plausible model fits rather than committing to just one. The Bayesian approach provides a means to quantify model uncertainty (in addition to data uncertainty) and improves the generalisation and calibration of predictions against out-of-sample observations. In addition, model uncertainty provides a useful guide for prioritising the collection of further observations.

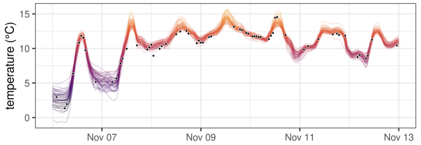
Examples

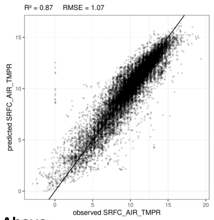
Our proposed approach is applicable to both spatial and spatio-temporal problems, here we briefly share one of each:

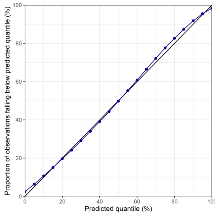
Modelling crowd-sourced weather observations

Unofficial weather stations across the UK provide a high density of weather observations, though with some quality issues. By augmenting our approach to include outlier detection and filtering, we can use these to learn high resolution probabilistic spatio-temporal interpolations of past weather conditions, e.g. surface air temperature shown below. Expanding the approach to harness numerical weather and climate model outputs as ‘auxiliary information’ may provide a suitable framework to learn efficient probabilistic downscaling of discrete forecast grids back to continuous space.







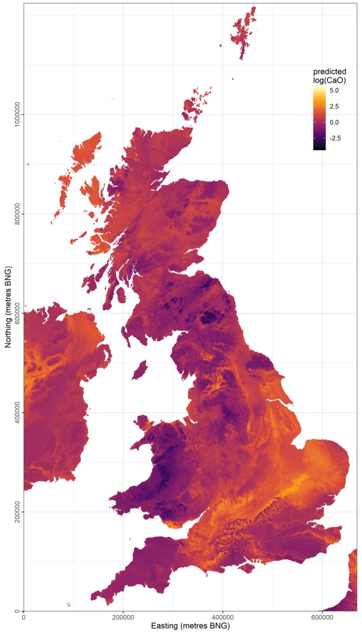


Above: observed-predicted scatter

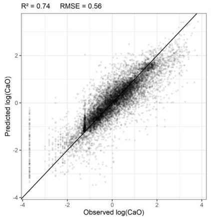
Right: Q-Q calibration

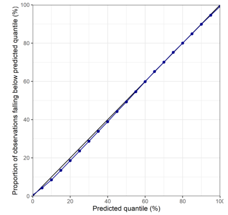
Left: posterior samples time series

Modelling surface geochemistry



Surface geochemistry captures effects of both surface and subsurface processes, and so provides a good test-bench for geospatial machine learning as well as being of stand-alone environmental importance.





Above: observed-predicted scatter

Right: Q-Q calibration

The future

Modern environmental AI approaches like ours enable more sources of information to be harnessed in order to learn predictive distributions that are as sharp as possible while remaining honest about uncertainty, whatever the target variables are.

Sharp and well-calibrated probabilistic predictions are the keystone of effective decision making, and by using new intelligence to improve the efficiency with which we make environmental decisions, we can improve our chances of achieving good outcomes on the road to net zero.

Charlie Kirkwood*, Theo Economou, Henry Odbert, Nicolas Pugeault

For more information see our upcoming paper **Bayesian deep learning for spatial interpolation in the presence of auxiliary information** to appear in the Mathematical Geosciences special issue on geostatistics and machine learning (in print 02/2022)

*contact: c.kirkwood@exeter.ac.uk

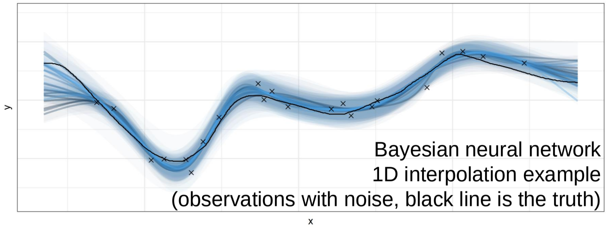
Deep geostatistics: incorporating computer vision for intelligent Bayesian interpolation between environmental observations

Charlie Kirkwood[1]*, Theo Economou[1,4], Henry Odbert[2], Nicolas Pugeault[3]

University of Exeter (1), UK Met Office (2), University of Glasgow (3), Cyprus Institute (4) *contact: c.kirkwood@exeter.ac.uk

Premise

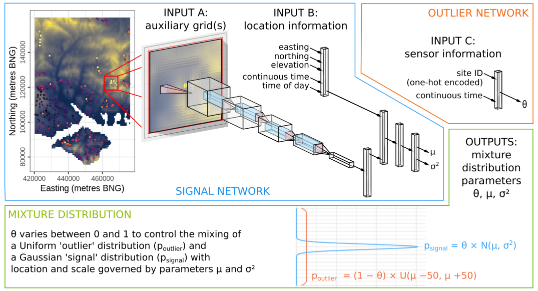
- Deep learning – machine learning using deep neural networks – is an efficient way to discover patterns in data that may be more complex than we could manually hypothesise.
- Here we learn spatial and spatiotemporal models that automatically harness information from gridded auxiliary datasets, such as digital terrain models and satellite imagery.
- This enables the production of detailed, accurate, and ‘natural’ looking maps which capture the effects of task-relevant structure in the landscape.
- Using approximate Bayesian inference, we can learn well-calibrated deep models that quantify epistemic and aleatoric uncertainties and avoid overfitting despite the capacity of deep models to do so.
- We can efficiently simulate ‘ensemble members’ from the posterior distribution, approximating the range of plausible hypotheses (data generating functions) that can explain the environmental phenomena we observe.
- In addition, we can make data collection more efficient by targeting regions for which the variability between plausible hypotheses – the epistemic uncertainty – is greatest.



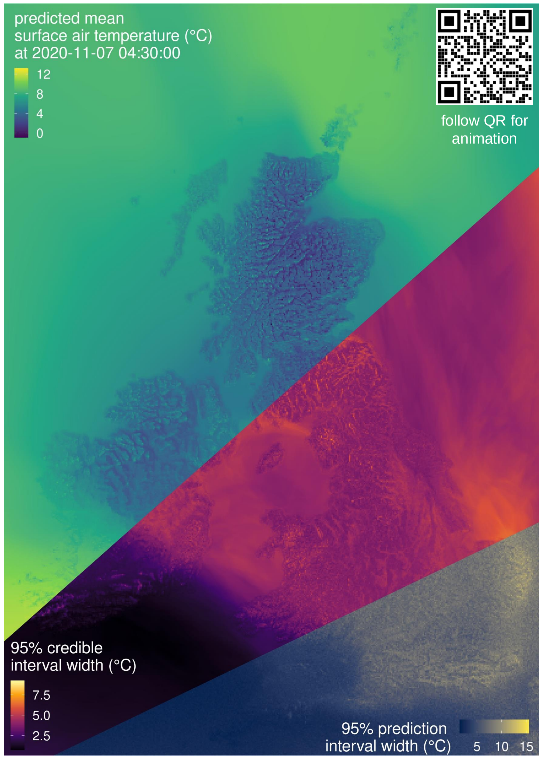
Bayesian neural network 1D interpolation example (observations with noise, black line is the truth)

Examples

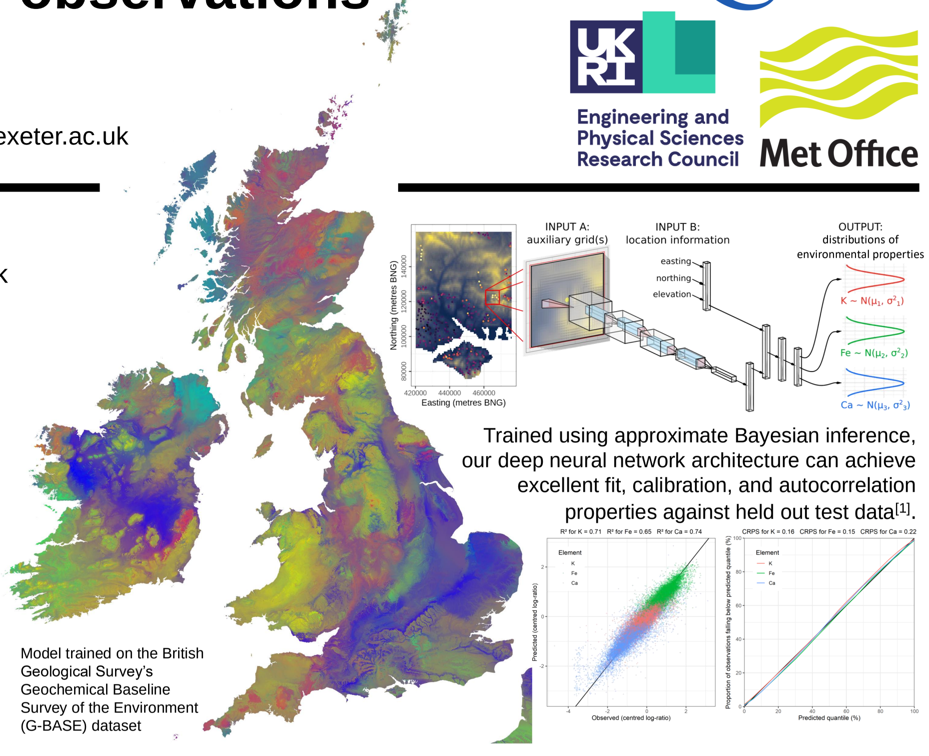
Below: A spatiotemporal mixture density network for interpolating crowd-sourced weather data[2]



θ varies between 0 and 1 to control the mixing of a Uniform 'outlier' distribution ($p_{outlier}$) and a Gaussian 'signal' distribution (p_{signal}) with location and scale governed by parameters μ and σ²



Model trained on the Met Office’s Weather Observation Website (WOW) dataset



Trained using approximate Bayesian inference, our deep neural network architecture can achieve excellent fit, calibration, and autocorrelation properties against held out test data[1].

Model trained on the British Geological Survey’s Geochemical Baseline Survey of the Environment (G-BASE) dataset

Above: Deep-learned spatial interpolation of land-surface geochemistry[1]

Past and future

The field of geostatistics arose in the latter half of the 20th century as a response to the problem of interpolating between point-sampled observations in geographic space. ‘Kriging’ (Gaussian process regression in two or three dimensions developed by Georges Matheron in the 60s after mining engineer Danie Krige) became the ubiquitous solution but has required restrictive assumptions of stationarity and lacks feature learning abilities.

Since 2012, deep learning has revolutionised computer vision and other applications by enabling automatic feature learning. Our results indicate that by developing suitable architectures and inference schemes it can also give rise to a new generation of geostatistical interpolators that are able to learn the potentially complex ways that observations relate to the landscape itself.

[1] Kirkwood et al. (2022) Bayesian deep learning for spatial interpolation in the presence of auxiliary information. Mathematical Geosciences 54, 507–531
[2] Kirkwood et al. (2022) A deep mixture density network for outlier-corrected interpolation of crowd-sourced weather data. arXiv preprint arXiv:2201.10544.

Appendix B

Conference photographs

Figure B.1: Attendees of the machine learning for weather and climate modelling workshop that took place at Corpus Christi College in Oxford in the first week of September 2019, organised by Matthew Chantry, Hannah Christensen, Tim Palmer, and Peter Dueben. A most interesting and inspiring conference - it felt like a historic event. I am stood in the top right corner of this photo - directly in front of Scott Hosking (in black) and left of Sebastian Lerch (in blue, in front of bench).

Figure B.2: Attendees of the Eumetnet workshop on AI for weather and climate that took place at the Royal Meteorological Institute of Belgium in the last week of February 2020, organised by Steven Dewitte and colleagues. Another extremely interesting conference, though along with new ideas I also brought home suspected COVID-19 (but thankfully was better in about a week). I am third from the left.