UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

AIRTON FERREIRA DE SOUZA NETO

SPATIO-TEMPORAL WIND SPEED FORECASTING WITH BAYESIAN UNCERTAINTY QUANTIFICATION

FORTALEZA

2023

AIRTON FERREIRA DE SOUZA NETO

SPATIO-TEMPORAL WIND SPEED FORECASTING WITH BAYESIAN UNCERTAINTY QUANTIFICATION

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Lógica e Inteligência Artificial. Área de Concentração: Ciência da Computação.

Orientador: Prof. Dr. César Lincoln Cavalcante Mattos.
Coorientador: Prof. Dr. João Paulo Pordeus Gomes.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S713s Souza Neto, Airton Ferreira de.
Spatio-temporal wind speed forecasting with Bayesian uncertainty quantification / Airton Ferreira de Souza Neto. – 2023.
69 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2023.
Orientação: Prof. Dr. César Lincoln Cavalcante Mattos.
Coorientação: Prof. Dr. João Paulo Pordeus Gomes.

1. Quantificação de incerteza Bayesiana. 2. Apredizado profundo. 3. Modelagem espaço-temporal. 4. Predição de vento. I. Título.

CDD 005

AIRTON FERREIRA DE SOUZA NETO

SPATIO-TEMPORAL WIND SPEED FORECASTING WITH BAYESIAN UNCERTAINTY QUANTIFICATION

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Lógica e Inteligência Artificial. Área de Concentração: Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. César Lincoln Cavalcante Mattos (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. João Paulo Pordeus Gomes (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. João Paulo do Vale Madeiro
Universidade Federal do Ceará (UFC)

Prof. Dr. Leonardo Ramos Rodrigues
Instituto Tecnológico de Aeronáutica (ITA)

Dedico o presente trabalho a todos que me apoiaram e me ajudaram ao longo dessa caminhada.

AGRADECIMENTOS

Ao Prof. Dr. César Lincoln Cavalcante Mattos e ao Prof. Dr. João Paulo Pordeus Gomes por me orientarem na construção do presente trabalho. O apoio de vocês foi essencial para a conclusão dessa dissertação.

À minha família, minha namorada, meus amigos e parentes, que me apoiaram em todos esses anos de estudo.

À Delfos, pelo apoio e disponibilidade dos dados e da infraestrutura para realização desse estudo.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

"O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado."

(Ariano Suassuna)

RESUMO

A predição de séries temporais de vento de curto e longo prazo possui grande utilidade para a indústria, sobretudo a de geração de energia eólica, tendo várias aplicações práticas no dia a dia operacional dos parques. Os resultados da predição são ainda mais poderosos e confiáveis quando associados a estimativas de incerteza, trazendo um maior apoio à tomada de decisão. Neste trabalho, uma modelagem orientada a dados, baseada em redes neurais profundas, é apresentada. A quantificação de incerteza associada à distribuição preditiva pode ser feita a partir de uma abordagem de aprendizagem Bayesiana. No entanto, no contexto de redes neurais e aprendizagem profunda, a abordagem Bayesiana convencional é intratável e computacionalmente custosa. Por outro lado, tem havido vários avanços recentes em técnicas de inferência Bayesiana aproximada em aprendizado profundo, em que destacam-se aquelas que não modificam os algoritmos de treinamento tradicionais. O presente trabalho propõe o uso de redes neurais profundas para a modelagem espaço-temporal do vento a partir de medições presentes nos sistemas de aquisição de dados de turbinas eólicas. São incluídas ainda as predições de modelos de previsão climática global, amplamente usados pela indústria energética. As predições realizadas são acompanhadas da quantificação da incerteza, extraída a partir de técnicas de inferência Bayesiana aproximada. A solução desenvolvida é avaliada em dados coletados de um parque eólico no sul do Brasil. Diferentes combinações de modelos e aproximações são comparadas a partir da acurácia alcançada e de métricas e gráficos de calibração da incerteza. Os experimentos executados indicam que a utilização de redes neurais convolucionais recorrentes (ConvLSTM) em comitês profundos (Deep Ensembles) proporciona os melhores resultados para a distribuição preditiva, podendo auxiliar a operação de parques eólicos.

Palavras-chave: quantificação de incerteza bayesiana; aprendizado profundo; modelagem espaço-temporal; predição de vento.

ABSTRACT

The prediction of short and long-term wind time series has great utility for the industry, especially for wind energy generation, with various practical applications in the day-to-day operation of parks. The results are even more powerful and reliable when associated with uncertainty estimates, providing greater support for decision-making. In this work, a data-driven modeling approach based on deep neural networks is presented. The quantification of uncertainty associated with the predictive distribution can be done using a Bayesian learning approach. However, in the context of neural networks and deep learning, the conventional Bayesian approach is intractable and computationally expensive. On the other hand, there have been several recent advances in approximate Bayesian inference techniques in deep learning, particularly those that do not modify traditional training algorithms. This work proposes the use of deep neural networks for the spatio-temporal modeling of wind based on measurements collected from wind turbine data acquisition systems. It also includes predictions from widely used global climate forecasting models in the energy industry. The predictions made are accompanied by the quantification of uncertainty, extracted using approximate Bayesian inference techniques. The developed solution is evaluated using data collected from a wind farm in South of Brazil. Different combinations of models and approximations are compared based on the achieved metrics and graphs of uncertainty calibration. The conducted experiments indicate that the use of recurrent convolutional neural networks (ConvLSTM) with Deep Ensembles provides the best results for the predictive distribution, potentially assisting the operation of wind farms.

Keywords: Bayesian uncertainty quantification; deep learning; spatio-temporal modeling; wind speed forecast.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BMA	Bayesian Model Averaging
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
ConvLSTM	Convolutional LSTM Network
CRPS	Continuous Ranked Probability Score
ECMWF	European Centre for Medium-Range Weather Forecasts
ERA5	ECMWF Reanalysis Dataset
GFS	Global Forecast System
KL Divergence	Kullback-Leibler Divergence
LSTM	Long Short-Term Memory
MC Dropout	Monte Carlo Dropout
MCMC	Markov-Chain Monte Carlo
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
MultiSWAG	Multi Stochastic Weight Averaging - Gaussian
NeurIPS	Neural Information Processing Systems
NLL	Negative Log-Likelihood
NOAA	National Oceanic and Atmospheric Administration
NWP	Numerical Weather Prediction
PDF	Probability Distribution Function
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SCADA	Supervisory Control And Data Acquisition
SGD	Stochastic Gradient Descent
SVR	Support Vector Regressor
SWA	Stochastic Weight Averaging
SWAG	Stochastic Weight Averaging - Gaussian
WTG	Wind Turbine Generator

LIST OF SYMBOLS

$p(...)$	probability distribution functions, in general.
$P-50, P-90$	50% and 90% Percentiles, respectively.
$\mathbf{w}$	Neural network weights, in general.
$\mathbf{X}, \mathbf{y}$	Input and output data sets, in general.
$\mathbf{S}_{k,t}$	Array with SCADA features, for turbine k, time t.
$\mathbf{F}_{m,t}$	Array with NWP features, for spatial coordinate m, at time t.
u, v	Wind speed components along latitude and longitude axis, respectively.
$\odot$	Hadamard (element-wise) product operator.
$v(...)$	Flattening (vectorizing) operation to reduce matrices dimensions.

TABLE OF CONTENTS

1 INTRODUCTION

Weather forecasting is a well-known problem with applications in several areas and a wide bibliography concerning various climate variables and different prediction time ranges. Meteorology and forecasting have been considered important since the antiquity, when people related climate phenomena to mythology and faith, but remained for many years with no significant scientific improvements, until the first measurement systems were made in the renaissance (GOLDSTEIN, 2002).

Although it has been a highly pursued task for decades, the efforts on weather forecasting only started to show reasonable and practical results recently, when computer processing came up with computational power to do fast mathematics. The complex relationship among climate variables and their future is commonly described by complex mathematical equations and a large amount of data, and only modern computers could do that work in an acceptable time (Jain e Mallick (2016) and Ongoma (2022)).

There are several important applications for weather forecasting. Perhaps the most important one is protecting people's lives by forecasting natural disasters (COUNCIL, 1991). One example is forecasting and preventing impacts from wildfires in forests (COEN, 2014). But there are many other applications, such as rain forecasting and frost probability for agricultural planning and helping air traffic management in airports (ISEH.A.; WOMA.T., 2013).

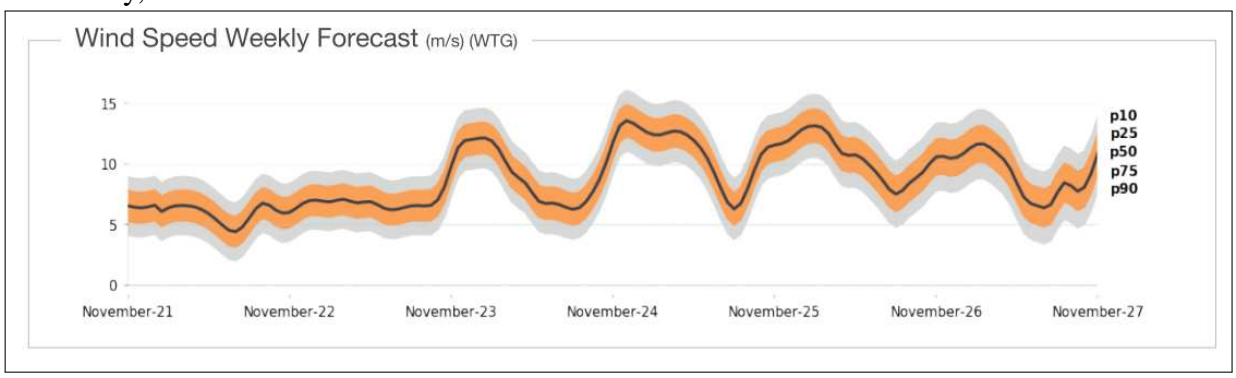
1.1 Wind Resource as a Source of Energy

Wind power has been used as a source of energy for centuries. They have evolved over time into highly efficient machines that can produce electricity on a large scale. The first recorded wind turbine was built in Scotland in 1887. However, the commercialization of wind power did not occur until the 70s when oil prices rose drastically, making wind power more attractive (MANWELL *et al.*, 2010).

One major limitation of wind power is that it is an intermittent resource. Storage options, such as batteries, are not common, so this kind of energy can hardly be stored (QUASCHNING; HANKE, 2019). Sites with strong winds are also necessary, otherwise efficiency decreases (PAVEZ *et al.*, 2021).

Wind power has seen tremendous growth in recent years, with 93.6 GW new istalled capacity. By the end of 2021, there was a total of 837 GW of installed wind power capacity

Figura 1 – Wind speed forecasts with its distribution percentiles. Interventions in turbines must not be scheduled on windy days. Forecasts support decisions in provisioning personnel and machinery, such as cranes for the interventions.



Fonte: Elaborated by the author.

globally, with China leading with almost half of that capacity. Next, there are United States generating 13.6%, and Brazil with 4%. By the year 2030, wind power could supply up to 30% of global electricity needs. Despite that, all this growth, wind energy's 5% generation share still remains very low, compared to other energy sources (COUNCIL, 2021).

1.2 Wind Speed Forecasts for the Energy Industry

This study will focus on wind speed forecasting, which holds significant importance for the energy industry, particularly for wind farms. Energy players deal every day with the stochastic nature of the wind. When looking at long-term wind speed prediction, one can have the advantage of assuring a generation predictability, which reduces the uncertainty of any investment decision (RODRIGO *et al.*, 2017). With short-term wind speed predictions, the owner is able to schedule turbine preventive and corrective maintenance in an intelligent way (see Fig. 1). It is interesting to focus the interventions in periods of low wind speed, arguing about non-business hour request for repairs, and staying out of the way of strong wind gusts, when it is impossible to do any work at the turbine. There is also a need of short and long term weather prediction for energy traders in financial market, as the energy demand and supply are determinant factors for the price composition (CHEN *et al.*, 2018). Prices usually rise with higher demand and higher cost of dispatch, which is mainly affected by low renewable resources. Hong *et al.* (2020) present a recent review of the most influential studies in the energy forecasting area, summarizing research trends, focused in power, demand and price forecasting, for both solar and wind plants.

When dealing with wind farm's specific operational concerns, there is an additional

Figura 2 – Photos of wake effect captured in an offshore wind farm. Generation is higher in turbines with little intervention coming from its neighborhood

Fonte: Hasager *et al.* (2017)

effect that can also become very important for decision making: the wake effect (see Jenkins (2021) for a detailed explanation). It is common in wind energy projects to have turbines affecting each other's generation depending on wind direction, due to the kinetic energy conversion of the wind. This turns out to have great impact in the energy production, forcing the operators to use strategies to reduce these effects in order to maximize the whole park's production. Yet, such issues still exist and cannot be ignored. These effects are also important when dealing with interventions in a specific wind turbine. Thus, the wind direction and the position of the turbine relative to the others are important features when using forecasts for operational purposes. A turbine may generate at full power while another one at the same moment will not. Hasager *et al.* (2017) present photos of this effect in a real world wind farm, such as the one depicted in Fig. 2.

Still, statistical uncertainty quantification is also highly important in wind forecasting. Gathering statistical values such as P-50 and P-90 (which are equivalent to 50% and 10% distribution percentiles) of the forecast distribution can leverage the power plant's owner decision making to a much more reliable place. It is common in energy industry to deal with percentile metrics, due to the stochastic nature of renewable energy sources. Thus, probabilistic aspects come up even on the very first viability studies.

1.3 General and Site-specific Data

Regarding weather forecasting, there are some well-known models to describe and predict weather data around all the world. The so-called Numerical Weather Prediction (NWP) models are driven by physics and fluid mechanics laws and present good results in a grid covering all the globe, in scales less than 1 degree (latitude and longitude measurements). These short-term models predict more than a week ahead. The short-time prediction often is presented in an hourly basis. Two of the most commonly used NWP models are the American Global Forecast System (GFS) model, from National Oceanic and Atmospheric Administration (NOAA) (NOAA, 2021), and the European Centre for Medium-Range Weather Forecasts (ECMWF) model (ECMWF, 2021).

Another important source of data is the Supervisory Control And Data Acquisition (SCADA) system of the wind farm. This system controls the turbine's operation and has special databases for the turbines, which store data from Wind Turbine Generator (WTG) sensors. These measurements include wind speed, wind direction, ambient temperature measured on the turbine's exterior, among others.

Regarding wind speed forecasting applied to the energy industries, it is mandatory to have a reliable accuracy, which cannot be well achieved only by a generic numerical model, such as GFS or ECMWF. It is also worth taking advantage of the power plant's site-specific source of data, in order to improve the model's accuracy for the wind farm being monitored. One can mix SCADA site-specific data with Numerical Weather Prediction (NWP) models in order to make new site-adapted models.

The study will be based on a wind farm in the South of Brazil. The ECMWF's ERA5 data is taken as the source of global forecasts, delimiting the area of the wind farm to simplify the data acquisition. It is publicly available on the climate data service from European Centre for Medium-Range Weather Forecasts (ECMWF) website[1]. The ECMWF Reanalysis Dataset (ERA5) data set maintains hourly historical data for a wide variety of climate variables. The data include some of the main features from the website, such as expectations of wind speed in both longitude and latitude axis, pressure measurements, relative humidity, dew point temperatures, among other information. For site-specific data, the SCADA system of the wind farm's turbines is also available. For this work, it is considered 2 years of data coverage from both sources.

[1] Data available in https://cds.climate.copernicus.eu

1.4 Bayesian Spatio-Temporal Forecasts

Deep learning is a subset of machine learning techniques comprising neural networks in general, which are mathematical entities based on human neurons. Goodfellow *et al.* (2016) introduce the main principles and algorithms regarding deep learning and neural networks. Today, there are plenty of applications of neural networks, going from image recognition, disease prevention and fraud detection to self-driving cars (ALZUBAIDI *et al.*, 2021).

Spatio-temporal models are complex models which aims to capture both spatial and temporal aspects of the data. It has been applied in many different areas, such as in epidemiology (SCIANNAMEO *et al.*, 2022) and traffic in cities (TASCIKARAOGLU, 2018). Wikle e Zammit-Mangion (2023) reviews some of the research in spatial and spatio-temporal models and discusses the best models and future work.

Machine Learning techniques are especially useful in the task of predicting wind time series, as one can add many features to the model without the need of describing explicitly the nature of climate changes, allowing the model to learn how to proceed with the data itself. Bochenek e Ustrnul (2022) present a critical review of some data-driven methods for weather prediction, discussing about future work in this area and concluding machine learning is now a key feature in this context. Moshrefi-Torbati *et al.* (2014) also summarize some methods, more specifically on the task of wind speed forecasting, and discuss different applications and time ranges. Zhang *et al.* (2014) aggregate probabilistic models applied directly to the task of wind generation forecasting, opposed to deterministic estimates. The article also states Bayesian techniques have been subject of much of the recent research, presenting some related work, such as Pinson e Madsen (2009) and Yang *et al.* (2012).

This study proposes a data-driven approach to predict the expected value for wind speed in a certain short-term period, pursuing applications on wind energy industries, especially on the wind farm's daily operation. Importantly, turbines' spatial positions were included to achieve better turbine-specific accuracy.

Deep learning has been showing great results when dealing with large amounts of data. Esteva *et al.* (2019) show examples in many healthcare areas. Although such network's complexity may bring better generalization in terms of point estimates, standard deep learning methods have little to show in the matter of confidence about their predictions.

Uncertainty estimates are provided to the models, through a Bayesian uncertainty quantification approach. A classical statistical Bayesian method to model predictive distributions

basically consists in assuming *a priori* distributions over the model's parameters, adding prior beliefs to them, and then yielding an *a posteriori* distribution, a process which statisticians call marginalization over the distribution parameters (GELMAN *et al.*, 2003). For example, even if the true value of a parameter is totally unknown, it is possible, for instance, to assume it follows a prior Gaussian distribution, since a very large absolute value would drive the model to worse generalization (GERON, 2019).

However, traditional Bayesian modeling, based on marginalization, becomes mathematically intractable in deep neural networks, which are usually complex models, having a huge number of parameters. Izmailov *et al.* (2021) discuss about this issue, arguing about how challenging it is to train a pure Bayesian model and to use sampling methods to yield a posterior distribution. To overcome these computing issues, some recent Bayesian uncertainty quantification techniques using approximate techniques can be applied, in the context of neural networks. Gawlikowski *et al.* (2021) and Abdar *et al.* (2021) review some Bayesian frameworks that can be applied to deep neural networks.

Indeed, these methods yield more reliable predictions without increasing the computational cost too much to train the models. Some of them are even capable of quantifying uncertainty estimates from pre-trained models. With these techniques, it is possible to add Bayesian uncertainty quantification on totally data-driven complex neural networks. Chapter 3 discusses more about Bayesian modeling and comments on concerns some authors have with this terminology.

This study focuses on Bayesian spatio-temporal forecasts, comparing the usage of recent simple to use approximate Bayesian frameworks, such as SWAG (MADDOX *et al.*, 2019), Monte Carlo Dropout (MC Dropout) (GAL; GHAHRAMANI, 2016) and Deep Ensembles (LAKSHMINARAYANAN *et al.*, 2017). The aim is to apply them on a totally data-driven wind speed forecast framework and to discuss the obtained results. All models will be compared using scoring metrics (GNEITING; RAFTERY, 2007) and calibration studies, which will be described in Chapter 3.

1.5 General and Specific Objectives

In this section, it is presented the general and specific objectives of this work.

1.5.1 General Objective

The main goal of this work is to model and evaluate spatio-temporal short-term site-specific wind speed forecasting, along with uncertainty quantification, comparing approximate Bayesian uncertainty quantification frameworks. The modeling process will take advantage of data extracted primarily from two sources: the power plant's data acquisition system (SCADA) and a global forecast model's predictions.

1.5.2 Specific Objectives

In order to pursue the goal above, the following efforts are performed:

1. To extract and process data from raw sources. Both SCADA data and ERA5 historical forecast data are considered.
2. To apply machine learning models to the task of short-term spatio-temporal wind speed forecasting. Hyperparameter tuning will be applied to sharpen the forecasts.
3. To leverage Bayesian deep learning frameworks to quantify uncertainty for the models' forecasts.
4. To compare the models' posterior beliefs using probabilistic metrics.
5. To discuss the results obtained from the models regarding their generalization capacity and their usability in practical daily operations.

1.6 Document Organization

As follows, some related work will be presented in Chapter 2. In Chapter 3, it is described the theoretical foundation behind the Bayesian uncertainty quantification models to be deployed and the metrics to be evaluated. In Chapter 4, it is presented the methodology around the data patterns and the models. Chapter 5 shows the results concerning these models and their usage. Finally, at Chapter 6 there is a final conclusion, with some guidance for further work.

2 RELATED WORK

Data-driven machine learning models are well-fitted for wind speed forecasts due to the amount of available data, both local and global, as stated in Chapter 1. In this chapter, it is presented recent related studies in Bayesian Deep Learning and in data-oriented weather prediction, especially wind speed forecasting.

2.1 Bayesian Deep Learning

Deep neural networks have achieved many advances in several areas, such as healthcare (ESTEVA *et al.*, 2019) and language models (BROWN *et al.*, 2020). Soniya *et al.* (2015) and Alzubaidi *et al.* (2021) present a summary of some of these applications. Within the area, the task of quantifying uncertainty in deep neural networks is still a common research objective. Standard Bayesian inference consists in marginalizing over a set of model parameters. In summary, first it considers prior distributions for the parameters. Then, after the data are observed, a posterior distribution is computed. Finally, the posterior distribution is used to marginalize the parameters and obtain a predictive distribution. This will be better discussed in chapter 3. Early works in Bayesian Deep Learning includes MacKay (1992a), MacKay (1992b), MacKay (1995), in which the author describes a framework to regularize and compare neural networks through a Bayesian perspective, relating Bayesian approach to Occam's Razor principle applied in the modeling process. Neal (1996) is another important early work which performs Bayesian inference over neural networks via Markov Chain Monte Carlo sampling methods.

However, for large complex neural networks, standard Bayesian inference is often computationally expensive and cannot be used in practical situations, which turns it to be intractable for common problems. In order to achieve uncertainty quantification measurements, many approximate Bayesian inference approaches have been introduced, including frameworks focusing both on scalability and generalization capabilities (SMITH; JOHNSON, 2020; DAXBERGER *et al.*, 2021; MADDOX *et al.*, 2019; GAL; GHAHRAMANI, 2016). Some of these methods can even be effortless adapted for production-ready models (MADDOX *et al.*, 2019).

Izmailov *et al.* (2021) argue about the performance of approximate Bayesian methods, compared to the classical Bayesian approach. Leading a computationally costly experiment, the authors trained a rigorously Bayesian network, applying Markov-Chain Monte Carlo (MCMC) models to sample the posterior predictive distribution in a classification problem. The work

showed that Bayesian approximate frameworks also result in precise data generalization, being able to approximate the true posterior distributions.

Regarding Bayesian Deep Learning, Gawlikowski *et al.* (2021) gather some recent state-of-the-art research, summarizing and classifying them in certain groups of study: *Variational Inference*, which approximates the posterior by tractable well-behaved functions, *Sampling*, which yields the posterior distribution by sampling, and *Laplace Approximations*, which simplifies the target distribution by approximating it, deriving a normal distribution for the network weights. Abdar *et al.* (2021) also summarize several Bayesian strategies, including Monte Carlo methods, variational inference for neural networks, auto-encoders and many others.

Presenting a Bayesian approximate approach for neural networks, Gal e Ghahramani (2016) state a simple technique that became a baseline in terms of Bayesian Deep Learning. One can estimate a posterior predictive distribution with a dropout strategy over neurons of a model, sampling outputs and performing Bayesian Model Averaging (BMA). Dropout (SRIVASTAVA *et al.*, 2014) is the process of turning off some neurons on each training step, forcing the model not to depend on any of them directly and preventing overfitting the data set. It is mathematically proven the relation between dropout's inferred distribution and the posterior of a deep Gaussian Process (DAMIANOU; LAWRENCE, 2013), showing that the objective function is actually equal to minimizing the Kullback-Leibler divergence, which is a measurement of the difference between the true posterior and the inferred distribution. It is a seminal study that can be used as a baseline for Bayesian approximate frameworks, due to its simplicity. However, this process should attach dropout layers to the model in the training process, as the author states. Details about it will be presented at Chapter 4.

Another well-known method, perhaps currently one of the most recognized ones due to its impressive results, is Deep Ensembles (LAKSHMINARAYANAN *et al.*, 2017). It has a simple procedure: (1) choose a proper scoring rule, as defined by Gneiting e Raftery (2007); (2) train some models, optionally using adversarial (GOODFELLOW *et al.*, 2015) training, as suggested in the original work; (3) build an ensemble with these models. Izmailov *et al.* (2021) give credit to this framework for being highly similar, in terms of model performance, to a Bayesian Neural Network yielded from sampling methods, which is the gold standard for rigorous Bayesian inference. On the other hand, Abe *et al.* (2022) mention that good calibration and results can be obtained from single networks alone, arguing that Deep Ensembles is more like a convenient tool than a "superior" technique. Although Deep Ensembles is presented as a

non-Bayesian probabilistic method, Hoffmann e Elster (2021) discuss about how it can as well be seen as a Bayesian method. Wilson e Izmailov (2020) also argue about Deep Ensembles being a Bayesian Model Averaging technique.

Maddox *et al.* (2019) present Stochastic Weight Averaging - Gaussian (SWAG), a simple method that can be used on pre-trained models. It is an enhancement over the ideas presented in Izmailov *et al.* (2018). After considering the weight's distribution as being Gaussian, the output's posterior predictive distribution can then be obtained by a sampling strategy. The authors argue about the fact that the Stochastic Gradient Descent (SGD) captures the geometry of the loss space, using its trajectory to infer a distribution over the parameters set. Recently, Wilson e Izmailov (2020) presented another variety for SWAG, called MultiSWAG, extending this inference approach to multiple independently trained SWAG models to achieve additional performance gains. The authors also argue about how this procedure can be viewed as an approximate marginalization method.

Daxberger *et al.* (2021) recently revisited Laplace approximation methods and argue that they can be competitive to sampling and other approximate inference strategies. The first studies involving Laplace approximations in neural networks date from the early 90s (MACKAY, 1992a; NEAL, 1996). In order to improve practical usage, this work also presents *'laplace'* library for PyTorch (PASZKE *et al.*, 2019), one of the most used machine learning frameworks for the Python coding language, containing scalable implementations for the Laplace approximation method. The idea consists in approximating the posterior using a Taylor series expansion around the *maximum a posteriori* estimate. The posterior is approximated by a Gaussian distribution, with its covariance being the Hessian matrix of the loss function relatively to its parameters. The Hessian component can be calculated on post-trained models. This approach also includes regularization terms to represent the log-prior component of the loss function.

General methods for performing approximate Bayesian inference over neural networks were presented in this section. Interestingly, some of them can be used even on pre-trained models. Later, the aforementioned methods will be used to extract uncertainty quantification for the models evaluated in the task of wind speed forecasting.

2.2 Weather Forecasting

Numerical Weather Prediction (NWP) models are widely used by forecasting services around the world. Regarding accurate site-specific predictions, which is the focus of this work,

there is plenty of study on data-driven weather forecasting approaches. As highlighted below, some works include uncertainty estimates as a goal and consider site-specific spatial coordinates as important features, similar to what this work proposes.

Liu *et al.* (2020) follow a Bayesian approach to perform spatio-temporal wind speed forecasts by using an image recognition network in a grid of measurement points. The proposed network combines a recurrent neural network and a convolutional network. Variational inference is employed to overcome the intractable marginalization of the parameters. The approximate posterior is chosen to be a mixture of two Gaussians. By targeting the maximization of the Evidence Lower Bound (ELBO), the network minimizes the KL divergence between the approximation and the true posterior. The posterior distribution, after some assumptions, is approximated by Monte Carlo sampling and Kernel Density Estimation (KDE). However, the modeling process assumes well distributed turbines over the area, to form an approximately well-defined grid, which is not the real scenario in most wind farms.

Yu *et al.* (2019) use a similar grid embedding idea. In this case, the turbines are attached to the grid based on their locations, following a simple algorithm that yields a possibly sparse grid over the plant's area. It states the concept of spatio-temporal scenes as multi-channel images and compares the usage of deep Convolutional Neural Network (CNN) to well-known machine learning models, such as Support Vector Regressors (SVRs) and Multi-Layer Perceptrons (MLPs). The results show great accuracy gain compared to other baseline deep learning methods.

A. Sanandaji (2022) aim to achieve spatio-temporal forecasting with graphs using a Long Short-Term Memory (LSTM) neural network. The nodes in the graphs represent the data generating stations and the edges between nodes indicate the relationship between them. No initial assumptions are made about the stations relationships. The predicted values for every node are retrieved at the same time. Results show significant improvement in short-term prediction performance, compared to other methods.

Zhu *et al.* (2018) present a CNN based network to tackle the task of spatio-temporal correlation, stating that the experimental results outperform common machine learning models, such as SVR, MLP and Decision Trees.

Taking advantage of powerful Markov-Chain Monte Carlo (MCMC) methods, He *et al.* (2014) use data from the power output of turbines, instead of the wind speed. The target remains correlated, as these two quantities are intrinsically dependent of each other. The work

takes into accounts the seasonality and diurnal aspects of the wind speed behavior, focusing on a specific time range of the day. This sounds consistent, since wind speed depends on several climate variables that are affected by daily, monthly or even yearly seasonality.

In opposition to traditional physical modeling for global forecasts, Espeholt L. (2022) propose a data-driven solution based on neural networks. According to the article, NWP physical models, which are complex and computationally costly, could be surpassed using deep learning for the task of global predictions itself. The network would do the work of learning the relationships between weather measurements over space and future. The trained model yields a 12-hour ahead prediction for precipitation that outperformed numerical predictors in a region in United States.

Wang *et al.* (2019) apply the strategy of combining historical data and using NWP data as a prior knowledge, in the context of weather forecasting. Their focus is forecasting in weather stations in China. Their modeling process includes information fusion, NLL-based training and ensembling. They focus on modeling different climate measurements gathered from the stations, such as temperatures, relative humidity and scalar wind speed.

In this study, it is presented a practical model combining both NWP data and historical SCADA data to achieve wind speed forecasting within a wind farm. Uncertainty quantification is attached through simple but effective approximate Bayesian inference techniques, while modeling both spatial and temporal aspects of the wind speed. Several neural network architectures are compared in terms of model generalization, using different metrics and calibration methods, as detailed in Section 3.3.

From the above, there is plenty of recent work on Bayesian Deep Learning applied to wind speed forecasting and other weather-related tasks. Chapter 3 details the approaches pursued in this research.

3 THEORETICAL FOUNDATION

Data-driven forecasting, in general, aims to find a function $f(\cdot)$ that estimates future patterns $\mathbf{Y}_{fut} \in \mathbb{R}^{L \times D_1}$ based on the past data $\mathbf{X}_{past} \in \mathbb{R}^{L \times D_2}$, along with other available information $\mathbf{Z}$, with L being time steps in future and past data and D the dimension of the attributes involved in the task. Thus, it is possible to write

$$\mathbf{Y}_{fut} = f(\mathbf{X}_{past}, Z). \tag{3.1}$$

Wind speed forecasting, as other environmental forecasting with applications in the energy industry, has the purpose of supporting decision making in operational situations. For example, take management use cases in a wind farm. Maintenance interventions in wind turbines, for security and regulatory reasons, cannot be done in winds higher than a critical value. Thus, cranes, for example, are required to repair some damaged machinery, which results in an expensive operational cost. Besides, paying for keeping on hold personnel and other equipment to wait for low wind days makes it even more expensive. Furthermore, when wind speed increases in the middle of an operation, it could lead to dangerous accidents.

Forecasting weather with point estimates becomes inadequate in this situation. Uncertainty quantification must be gathered to model outcomes for the predictions to show how certain they are, based on the data they were presented to. In order to enhance predictive analysis and quantify uncertainty estimates, this work takes advantage of Bayesian inference to yield a posterior predictive analysis.

In this chapter the theoretical background behind the models used for weather forecasting is described, in section 3.1. It is also detailed how one can compare them, measuring generalization capacity and verifying the models calibration. An overview of Bayesian statistics, along with a discussion about the concerns when applying some methods in the context of deep learning, will be also discussed further, in section 3.2.

3.1 Time Series Modeling

Data with both spatial and temporal features are considered. From the perspective of deep learning modeling, for comparison purposes, it feels natural to add the data behaviors separately to the models, and see the results yielded when adding complexity to the models evaluated. In other words, the models will increase complexity with temporal capabilities, and

then with both temporal and spatial capabilities. In this section, it is introduced three different networks, and the difference between them, in terms of learning process, is detailed.

The first one is the Multi-Layer Perceptron (MLP), a fully connected network, which is not capable to directly learn neither spatial nor temporal features from data, due to its limited architecture. However, temporal and spatial data can still be extracted from global forecast models, turning this model applicable as a baseline. These features will be added, still, using an auto-regressive technique. As follows, it is also analyzed two other networks: the Long Short-Term Memory (LSTM), a recurrent neural network capable of learning temporal relationship from data, and the Convolutional LSTM Network (ConvLSTM), a convolutional network which can learn both spatial and temporal relationship from the data.

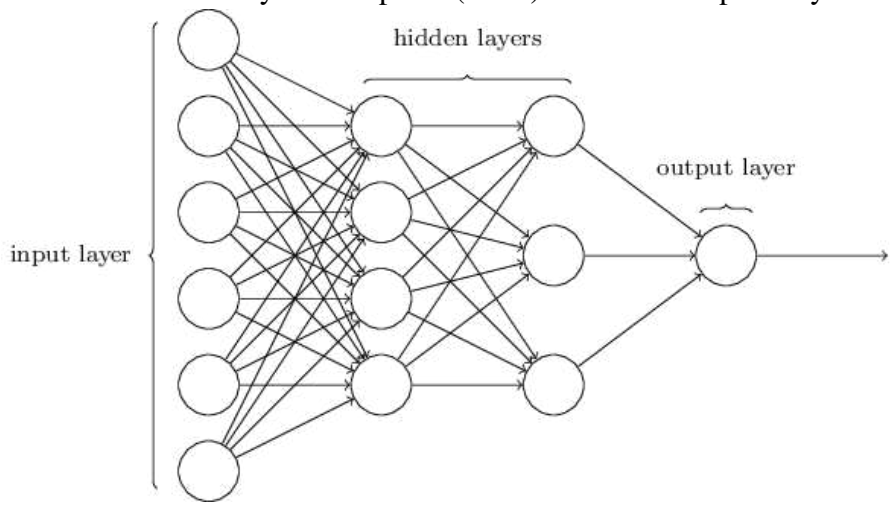
3.1.1 Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is one of the first proposed artificial neural networks. Perceptrons, which are mathematical entities inspired by human's neurons, were first presented by Rosenblatt (1958). MLPs extended that initial approach, having fully-connected layers of neurons, organized hierarchically. This new model became recognized a few decades later, at the 80s, with many practical applications, for example speech recognition (BOURLARD; WELLEKENS, 1989). Since then, MLP has been widely used as a powerful tool for many regression and classification problems.

This network consists in having layers with several neurons, all of them fully connected to the neurons in the previous and next layer, linked with an activation function, such as the hyperbolic tangent, the sigmoid function or the *ReLU* (rectified linear unit). The information is propagated without any cyclical step, which describes a *feedforward network*. The network's parameters are fitted to the data usually by the means of the backpropagation algorithm, which consists in a *gradient descent* method applied to the neural network. Goodfellow *et al.* (2016) go deeper in the explanation of the network. Fig. 3 shows the representation of a neural network with its input, output and hidden layers.

A simple example of this neural network, with one layer, can be described by (3.2). The architecture can be repeated to define more layers, producing a more complex network. In this equation, subscript h refers to the hidden layers parameters, and o to the output layer parameters, $\mathbf{W_h} \in \mathbb{R}^{D_1 \times H}$ and $\mathbf{B_h} \in \mathbb{R}^{D_1 \times H}$ are the coefficient and bias matrices in the input, respectively, and $\mathbf{W_o} \in \mathbb{R}^{H \times D_2}$ and $\mathbf{B_o} \in \mathbb{R}^{H \times D_2}$ are the coefficient and bias matrices of the

Figura 3 – Scheme of Multi-Layer Perceptron (MLP) with its multiple fully-connected layers.



Fonte: Extracted from Nielsen (2015).

output, with D_1 and D_2 representing input and output dimensions, respectively, and H being a defined hidden layers' dimension. ϕ is an arbitrary activation function. It has to be the same on each layer. $\mathbf{X} \in \mathbb{R}^{\cdots \times D_1}$ and $\mathbf{Y} \in \mathbb{R}^{\cdots \times D_2}$ are the input and output data being modelled.

$$\begin{aligned} \mathbf{H} &= \phi\left(\mathbf{X}\mathbf{W}_h^T + \mathbf{B}_h^T\right) \\ \mathbf{Y} &= \phi\left(\mathbf{H}\mathbf{W}_o^T + \mathbf{B}_o^T\right) \end{aligned} \tag{3.2}$$

3.1.2 Long Short-Term Memory

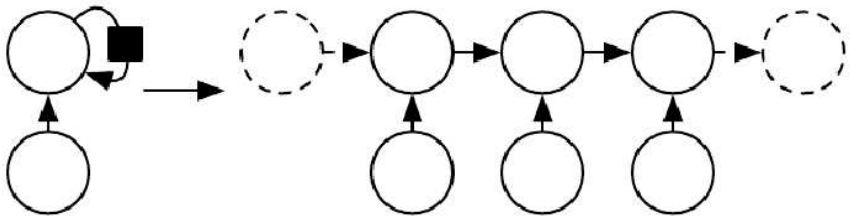
Hochreiter e Schmidhuber (1997) first presented the Long Short-Term Memory (LSTM), one example of Recurrent Neural Network capable of updating hidden states in order to learn relationships between sequential data. This network became acknowledged for tackling problems such as speech recognition (LIU *et al.*, 2018).

A recurrent neural network can be viewed as multiple copies from the same neural network, each one of them propagating knowledge for its successors in a series. Fig. 4 represents a common scheme for RNNs. $\mathbf{x}_t$ represents the pattern's features over time, and $\mathbf{h}_t$ is the hidden state responsible for carrying knowledge throughout the time steps.

The architecture of a LSTM cell (see Fig. 5) is composed by intern neurons called gates. There is particularly three of them:

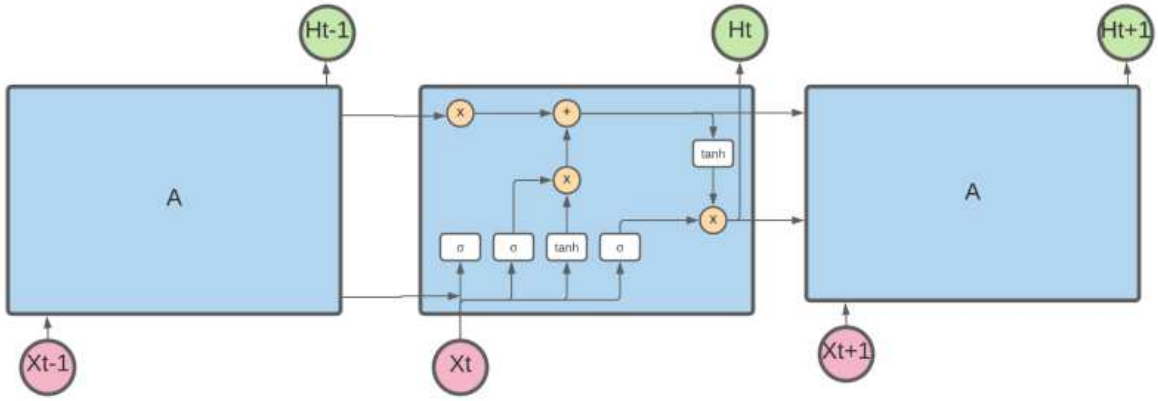
- Forget Gate: This gate throws out unnecessary information from the cell.
- Input Gate: Useful information is added to the cell by this gate.
- Output Gate: This gate yields useful information from the cell as an output.

Figura 4 – The idea of Recurrent Neural Network is to copy multiple times the same network, each one of them propagating knowledge for its successors.



Fonte: Extracted from Academy (2022).

Figura 5 – The structure of an LSTM cell. Here are represented the input, output and forget gates.



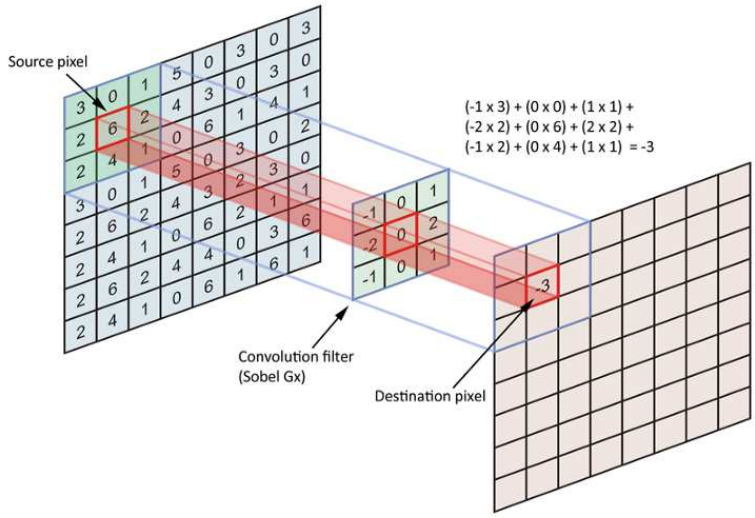
Fonte: Extracted from Moreira *et al.* (2022).

The mathematical representation of a LSTM network is defined in (3.3), where the symbol $\odot$ denotes Hadamard (element-wise) product operator. One can write the network as

$$\begin{aligned}
\mathbf{f}_t &= \sigma_g(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f),\\
\mathbf{i}_t &= \sigma_g(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i),\\
\mathbf{o}_t &= \sigma_g(\mathbf{W}_o\mathbf{x}_t\mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o),\\
\hat{\mathbf{c}}_t &= \sigma_c(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t,\\
\mathbf{h}_t &= \mathbf{o}_t \odot \sigma_h(\mathbf{c}_t),
\end{aligned} \tag{3.3}$$

where $\mathbf{x}_t \in \mathbb{R}^D$ represents a pattern at time t, $\mathbf{f}_t$, $\mathbf{i}_t$ and $\mathbf{o}_t$ the forget, input and output's activation vectors, both of them with values between 0 and 1, $\mathbf{h}_t$ and $\hat{\mathbf{c}}_t$ the hidden state and cell input activation vectors, both with values between -1 and 1, $\mathbf{c}_t \in R^H$ the cell input activation vector, and $\mathbf{W} \in \mathbb{R}^{H\times D}$, $\mathbf{U} \in \mathbb{R}^{H\times H}$, $\mathbf{b} \in \mathbb{R}^H$ are the parameters and bias sets to be learned along the training steps. Superscripts D and H represent the number of input features and hidden units, respectively. Goodfellow *et al.* (2016) goes deeper in the explanation of this network and other

Figura 6 – An example of convolution operation in matrices. This operation is designed to extract relevant features from input data.



Fonte: Extracted from this article.

recurrent networks.

3.1.3 Convolutional LSTM Network

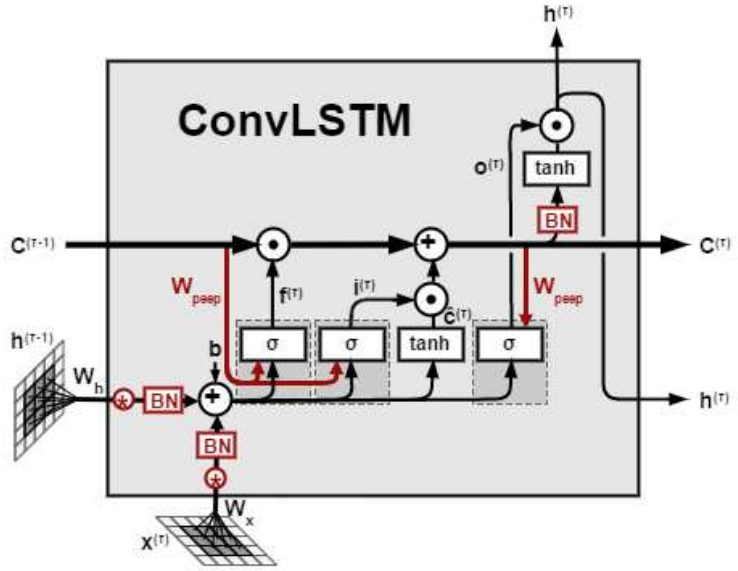
The ConvLSTM, first presented in Shi *et al.* (2015), is a network capable of learning both spatial and temporal relationship. This is achieved extending the behavior in LSTM network by adding convolutional structures in state transitions. Fig. 6 is an example of how a convolution happens in a matrix. This operation is designed to help extracting relevant features from the input data by capturing spatial patterns and dependencies. Many convolution layers with different filters can be combined, enhancing the learning capabilities of these structures. Fig. 7 represents the structure of one ConvLSTM cell, which is enhanced with convolution operators on the cell's gates. Combining spatial and temporal learning features allows these networks to be applied in many problems, such as prediction of the next frame in videos (DESAI *et al.*, 2022).

3.2 Bayesian Inference in Deep Learning

Point estimates can be highly uninformative, especially in cases where statisticians do not have much information about the real behavior of the data, or where the models disagree too much between the data being modelled and the ground truth values. Bayesian inference, however, indicates not only an estimate for what the real value is most likely to be, but also a measurement of confidence one can have for this information.

When predictive distributions are wider, the expected beliefs become much more

Figura 7 – An illustration of a ConvLSTM cell. The cell's gates are now enhanced with convolution operations.



Fonte: Extracted from this article.

unlikely to become true, and the model tells, in qualitative and quantitative ways, that the stakeholders cannot rely on them in this region of the data. On the other hand, when the models have reliable information and support to enforce their beliefs, Bayesian inference carries good support to what they are claiming, through thinner predictive distributions. Statistical models can tell when the data indeed indicates, with great confidence, that ground truth value should be close to what it is expected to be.

In terms of uncertainty types, Bayesian approaches, in general, are able to capture both epistemic uncertainty, the one that comes from the lack of data and from the unknowledge of the analyzed phenomena, and aleatoric uncertainty, the one coming from the intrinsic variation from data generation sources (GAL *et al.*, 2022). It is possible to reduce epistemic uncertainty by gathering more data or mastering the modeling process, but aleatoric uncertainty cannot be reduced, although it can be quantified, still (MUKHOTI *et al.*, 2022).

In the context of the energy industry, regarding wind speed forecasting, it is common to associate an estimate for wind speed, for example the 50% data percentile P-50, along with other percentiles, to add uncertainty information extracted from the predictive modeling procedure (PINSON *et al.*, 2006). It is very important to show information about uncertainty, indeed, because climate variables are highly unpredictable, due to the vast possibilities of unusual phenomena that can be involved in this matter. Rainy days can be difficult to predict in a small area, for example.

When talking about statistics, there are two different schools of studies: the frequentist and the Bayesian. The frequentist school focuses on the frequency of the observed data, inferencing statistics from an estimation approach (CASELLA; BERGER, 2002). In frequentist statistics, parameters are considered to be fixed but unknown values, and conclusions are drawn based on observed data by using methods such as hypothesis testing and confidence intervals.

Bayesian statistics, in opposition to a frequentist approach, consists in adding prior beliefs to the model, and then, after incorporating evidence from the observed data, yielding a predictive distribution combining both, which is finally called posterior distribution (BISHOP, 2006).

The likelihood function, with respect to the parameter set, is a function assigning higher values to parameter sets which are more likely to describe the data in a certain statistical model. From a frequentist approach, the most well-fitted parameter set would be the one that maximizes the likelihood, which is the same as minimizing the negative log-likelihood.

Let $\mathbf{w}$ be the parameters of the predicted distribution, $\mathbf{y}$ and $\mathbf{X}$ the output and input from observed data, respectively. The likelihood function L, defined for a probability distribution p parameterized by $\mathbf{w}$, can be written as

$$L(\mathbf{w}) = p(\mathbf{y}|\mathbf{X},\mathbf{w}). \tag{3.4}$$

From a Bayesian viewpoint, there is not only one ”correct” parameter set. Instead, it is usually assumed that the parameter set is also represented by a distribution $p(\mathbf{w})$. The integration of the likelihood function, over all the parameter set distribution, is called the marginal likelihood, or the evidence of the model, which is a measurement representing how strong observed data is connected to the particular statistical model being evaluated. Importantly, it considers the whole parameter distribution, instead of a particular point estimate, a process called marginalization (GELMAN *et al.*, 2003). It is given by

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})d\mathbf{w}. \tag{3.5}$$

By the Bayes theorem, the posterior distribution of the parameter set, which combines the prior information and the knowledge gathered from the data, can be described mathematically by

$$p(\mathbf{w}|\mathbf{X},\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}. \tag{3.6}$$

Usually $p(\mathbf{y}|\mathbf{X})$ is not available. It can be understood, though, as a normalization factor, assuring the mathematical integrity of the posterior (BISHOP, 2006). It is worth emphasizing that the marginal likelihood has important applications in hyperparameter tuning, such as in Gaussian Processes models (WILLIAMS; RASMUSSEN, 2006). On the other hand, some algorithms are able to ignore it, for example, sampling from the posterior (WANG; PARK, 2020) and Naive Bayes classifiers (MURPHY, 2012), which consider only the numerator of the Bayes theorem.

After obtaining the posterior distribution for the parameter set, the predictive distributions for a new input and unobserved output, denoted respectively by $\mathbf{x}_*$ and $\mathbf{y}_*$, can be computed by

$$p(\mathbf{y}_*|\mathbf{x}_*) = \int p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{y}) d\mathbf{w}. \tag{3.7}$$

In this work, the Bayes rule is not used directly to compute the posterior distribution. Considering different Bayesian approximate methods, described in the next sections, K samples are taken from the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$, without computing the marginal likelihood. Then, these samples are used to approximate $p(\mathbf{y}_*|\mathbf{x}_*)$:

$$\begin{aligned} \tilde{\mathbf{w}}_k &\sim p(\mathbf{w}|\mathbf{X}, \mathbf{y}), \\ p(\mathbf{y}_*|\mathbf{x}_*) &\approx \frac{1}{K} \sum_{k=1}^{K} p(\mathbf{y}_*|\mathbf{x}_*, \tilde{\mathbf{w}}_k). \end{aligned} \tag{3.8}$$

There are different workarounds to deal with the difficulty to work with the equations involved, often analytically intractable. One can assume, for example, conjugate prior and posterior distributions to simplify mathematics, a process which leads to analytical solutions (BISHOP, 2006). This may not be suitable for all models though, due to its strong hypothesis. There is also methods for approximating this posterior, such as Laplace approximations (SMITH; JOHNSON, 2020; DAXBERGER *et al.*, 2021), or forming its distribution by sampling methods, such as in Markov Chain Monte Carlo and Hamiltonian Monte Carlo (NEAL, 2011; WILSON; IZMAILOV, 2020).

Deep learning models, however, have several parameters to marginalize in complicated nonlinear structures, which turns the above equations and the training process to be often intractable and computationally costly. Much has been discussed abut this issue in the recent years, with the raise in the number of data-driven models, as the amount of available data keeps growing (IZMAILOV *et al.*, 2021). Still, the Bayesian paradigm has been applied over

approximations and adaptations of machine learning inference techniques. Gal *et al.* (2022) present a great discuss between some qualified and influential researchers on statistics and machine learning about this matter of quantifying predictive uncertainty and the many concerns about how this uncertainty is being extracted, what it represents and how rigorous the theoretical background behind them is.

Many recent researches state how to handle Bayesian inference and extract uncertainty quantification from deep neural networks. Some of them can be applied even on pre-trained models, and require much less time and resource to extract uncertainty quantification to enhance the model. This work particularly focus on four of them: Monte Carlo Dropout (MC Dropout), Deep Ensembles, Stochastic Weight Averaging - Gaussian (SWAG) and its successor Multi Stochastic Weight Averaging - Gaussian (MultiSWAG), described as follows.

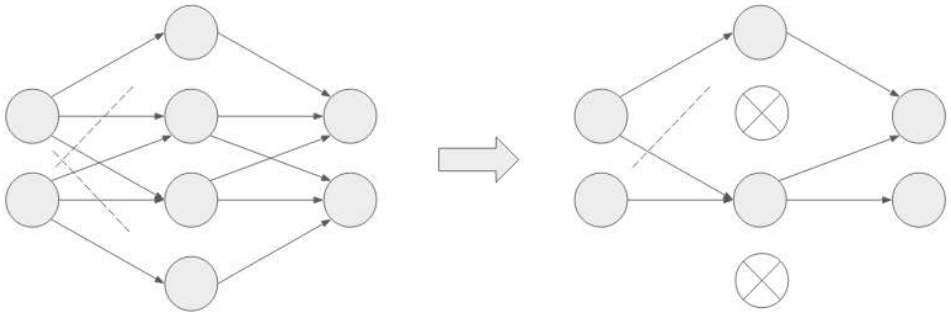
3.2.1 Monte Carlo Dropout

Monte Carlo Dropout (MC Dropout) was first presented by Gal e Ghahramani (2016) and it is particularly important as a practical baseline. This method takes advantage of the Dropout strategy (SRIVASTAVA *et al.*, 2014), which is simply randomly ”turning off” some of the neurons in the process of training. This process is commonly used as regularization method: (1) At training time, some neurons are turned off. By doing this, the neurons are trained to be more “independent” from each other, forcing the network to generalize better and prevent overfitting. (2) In prediction time, no neurons are turned off. The outputs from the model are rescaled to compensate that. Fig. 8 illustrates the process of deactivating neurons using dropout.

However, the dropout procedure can be thought as a method of sampling from neural networks, after which a predictive distribution can be inferred. This way, Gal e Ghahramani (2016) describe Monte Carlo Dropout (MC Dropout), presenting dropout as a Bayesian approximate inference method. The authors also show that the dropout strategy is equivalent to minimizing the Kullback-Leibler Divergence between an approximate distribution and a Deep Gaussian Process (DAMIANOU; LAWRENCE, 2013), which is purely Bayesian. This method has no constraints or particularities in the matter of usage, so it can be applied in any neural network with dropout layers before neuron layers.

Be the random variable $\mathbf{z} \sim \text{Bernoulli}(r)$, with r representing the dropout rate of the neurons, $\mathbf{z_k}$ a sample from $\mathbf{z}$, and $\mathbf{w}$ the parameter set of the trained model. The element-wise product of these vectors sampled from $\mathbf{z}$ and the parameter set defines another random variable $\mathbf{w}_z$.

Figura 8 – Illustrating the Dropout strategy. Neurons are dropped randomly in the process of training. Sampling with this strategy is the center idea of Monte Carlo Dropout (MC Dropout) framework.



Fonte:
Elaborated by the author.

Let $\mathbf{w}_k$ be samples from this random variables. Approximating the output posterior distribution $\mathbf{y}_*$ by a Gaussian, it is possible to write

$$\begin{aligned} \mathbf{w}_k &\sim \mathbf{w}_z = \mathbf{z} \odot \mathbf{w}, \\ \mathbf{y}_{*,k} &= f(\mathbf{X}_*, \mathbf{w}_k), \\ \mu_* &= \frac{1}{K}\sum_{k=1}^{K} \mathbf{y}_{*,k}, \\ \sigma_*^2 &= \frac{1}{K-1}\sum_{k=1}^{K} (\mathbf{y}_{*,k} - \mathbf{y}_*)(\mathbf{y}_{*,k} - \mathbf{y}_*)^T, \\ p(\mathbf{y}_*|\mathbf{x}_*) &\approx \mathcal{N}(\mathbf{y}_*|\mu_*, \sigma_*^2). \end{aligned} \tag{3.9}$$

with f representing the neural network function. μ_* and σ_*^2 represent the mean and the variance of the Gaussian predictive distribution approximating the posterior. The symbol $\odot$ represent the Hadamard (element-wise) product operator. The predictive distribution is sensitive to the hyperparameter r, that must be tuned in the modeling process.

3.2.2 Deep Ensembles

Acknowledged for the good results achieved in diverse areas, Deep Ensembles, presented by Lakshminarayanan *et al.* (2017), is a widely used method when pursuing approximate Bayesian inference in deep learning models. The author suggests a framework for this method, consisting in three steps:

1. Choose a proper scoring rule. The concept of proper scoring rules will be presented later in Section 3.3. Negative Log-Likelihood (NLL), for example, is a proper scoring rule.

2. Train models using adversarial training (GOODFELLOW *et al.*, 2015). The core idea is to input slightly different training data in order to force the model to generalize. Adversarial training at this phase of the process is optional, as the author says.
3. Train an ensemble with the models and infer from its outputs.

At step 3, there are have several ways to proceed. The authors focus on training the models using the full training data set and takes advantage of random shuffling of data and random initialization to yield different models. Ensemble is then treated as a uniformly-weighted mixture model. Although there are other ensemble methods, this work focus on this simple one. In order to use this ensemble method, each model has to be trained yielding two outputs: the mean and variance of the output data, as state Nix e Weigend (1994). The article is not precise whether to double the whole network or to fork the final layer into two different outputs. In this study, all the networks layers are doubled to output variance. It means the variance output has its own network.

Consider a mixture model composed by K Gaussian models, with predicted means and predicted variances given by μ_m and σ_m^2, respectively. The output $\mathbf{y}_*$ of the ensemble model is described by a Gaussian distribution, with predicted mean and variance μ_* and σ_*^2 given by

$$\begin{aligned} \mu_* &= K^{-1} \sum_{m=1}^{M} \mu_m(\mathbf{x}), \\ \sigma_*^2 &= K^{-1} \sum_{m=1}^{M} (\sigma_m^2(\mathbf{x}) + \mu_m^2(\mathbf{x})) - \mu_*^2(\mathbf{x}), \\ p(\mathbf{y}_*|\mathbf{x}_*) &\approx \mathcal{N}(\mathbf{y}_*|\mu_*, \sigma_*^2). \end{aligned} \tag{3.10}$$

Although it is presented as a non-Bayesian probabilistic method, Hoffmann e Elster (2021) discuss about how it can as well be seen as an approximate Bayesian method. Wilson e Izmailov (2020) also argue about Deep Ensembles being a Bayesian Model Averaging technique.

3.2.3 Stochastic Weight Averaging - Gaussian

More recently, Maddox *et al.* (2019) presented a simple method, applicable even in pre-trained neural networks, with a particular framework based on Stochastic Weight Averaging (IZMAILOV *et al.*, 2018).

Stochastic Weight Averaging (SWA) takes advantage of the route traced by Stochastic Gradient Descent (SGD) training in the loss function's surface. After the commons training phase, with the network already in a convergence region, one can take an average of different

possible parameter values, instead of taking only the final parameters estimates. The parameters samples can be extracted through SGD iterations in this region of the loss function. Izmailov *et al.* (2018) state that using a high constant learning rate force the model to explore other areas and ensure not to be around only one estimate. This can be viewed as a regularization method for deep neural networks.

Considering the training steps over a pre-trained model, SWA is the process of taking the parameters vector $\mathbf{w}$ as an average over M of these SGD iterates:

$$\mathbf{w}_{\mathbf{SWA}} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{w}_m. \tag{3.11}$$

Stochastic Weight Averaging - Gaussian (SWAG)'s main idea is to generalize the SWA approach by taking not only the average, but inferring a posterior Gaussian distribution over the parameter's set. The output's posterior distribution can be achieved by just sampling from the pre-trained Bayesian model. This framework performs BMA with little additional computational cost. The inferred Gaussian distribution of the parameters, whose mean and variance are represented by μ_w and σ_w^2, is given by

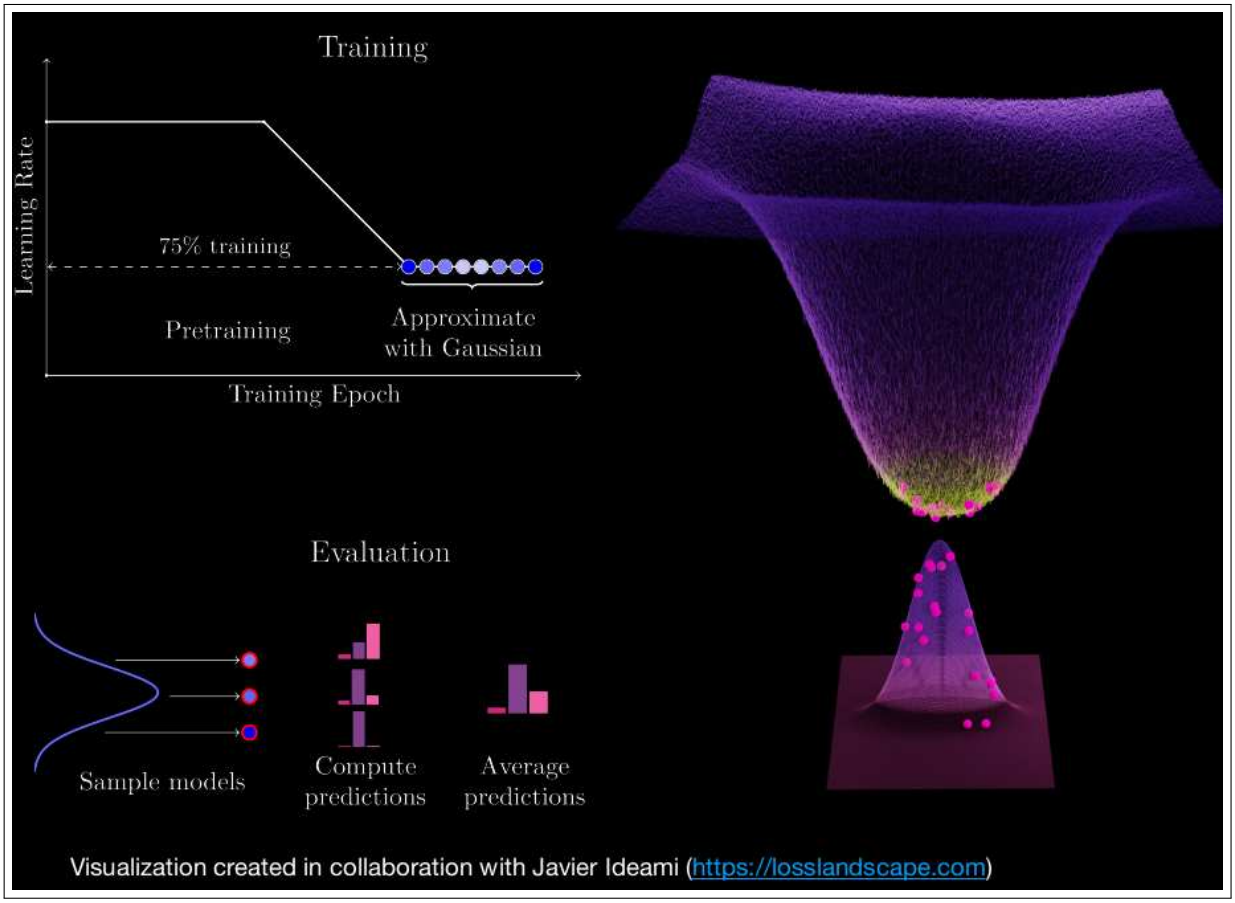
$$\begin{aligned}
\mu_w &= \frac{1}{M}\sum_{m=1}^{M}\mathbf{w}_m,\\
\sigma_w^2 &= \frac{1}{M-1}\sum_{m=1}^{M}(\mathbf{w}_m-\mu_w)(\mathbf{w}_m-\mu_w)^T,\\
p(\mathbf{w}|\mathbf{X},\mathbf{y}) &\approx \mathcal{N}\left(\mathbf{w}|\mu_w,\sigma_w^2\right).
\end{aligned} \tag{3.12}$$

The output distribution $\mathbf{y}_*$ can be achieved from K network outputs $f(\mathbf{X}_*,\mathbf{w}_k)$, using parameters sampled from the $p(\mathbf{w}|\mathbf{X},\mathbf{y})$ distribution. Approximating $\mathbf{y}_*$ by a Gaussian distribution, it is written

$$\begin{aligned}
\mathbf{w}_k &\sim p(\mathbf{w}|\mathbf{X},\mathbf{y}),\\
\mathbf{y}_{*,k} &= f(\mathbf{X}_*,\mathbf{w}_k),\\
\mu_* &= \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}_{*,k},\\
\sigma_*^2 &= \frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{y}_{*,k}-\mathbf{y}_*)(\mathbf{y}_{*,k}-\mathbf{y}_*)^T,\\
p(\mathbf{y}_*|\mathbf{x}_*) &\approx \mathcal{N}\left(\mathbf{y}_*|\mu_*,\sigma_*^2\right),
\end{aligned} \tag{3.13}$$

where μ_* and σ_*^2 represents the mean and the variance of the Gaussian distribution approximating the posterior. The predictive distribution is sensitive to the SWAG learning rate hyperparameter,

Figura 9 – NeurIPS 2019 presentation of the SWAG method. Here are the loss function's surface and the samples in which the posterior the parameter's distribution inference is evaluated.



Fonte: The presentation can be found on https://github.com/wjmaddox/swa_gaussian.

which is the one defined after the training phase is already over and the process of collecting parameters for SWAG takes place. It must be tuned in the modeling process.

Fig. 9 is part of the presentation at Neural Information Processing Systems (NeurIPS), 2019 edition, that took place in Vancouver, Canada. It represents graphically the idea of the SGD iterates through the surface of the loss function.

3.2.4 Multi Stochastic Weight Averaging - Gaussian

In Wilson e Izmailov (2020), the authors show the results for some Bayesian approximate methods, comparing them to computationally heavy Bayesian sampling networks, and showing one can achieve good results without the need to train this kind of networks. They also present Multi Stochastic Weight Averaging - Gaussian (MultiSWAG), an extension to SWAG, by generalizing the idea proposing a multimodal posterior inference, opposed to one simple Gaussian approximation. It can be evaluated in a simple way, averaging multiple independently trained SWAG Bayesian model outputs.

Let be K Gaussian distributions $p(\mathbf{w_k}|\mathbf{X},\mathbf{y})$, defined as in Eq. (3.12), for the parameters of the trained models. The posterior distribution of the parameters $p(\mathbf{w}|\mathbf{X},\mathbf{y})$ can now be represented by a mixture of these distributions

$$p(\mathbf{w}|\mathbf{X},\mathbf{y}) = \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{w_k}|\mathbf{X},\mathbf{y}). \tag{3.14}$$

The predictive distribution of $\mathbf{y}_*$ is similar to the one described in Eq. (3.13).

3.3 Probabilistic Evaluation Metrics

In order to compare the approximate Bayesian frameworks for deep neural networks, it is necessary to use proper scoring rules (GNEITING; RAFTERY, 2007). A proper scoring rule is a metric that rewards accurate and reliable predictions by assigning higher values to forecasts that are closer to the true probabilities. It encourages forecasters to provide calibrated probability estimates rather than biased or overconfident predictions. As follows, two important proper scoring rules are presented. They will be used for comparison purposes in the evaluated models.

3.3.1 Negative Log-Likelihood (NLL)

The likelihood function was described in Eq. (3.4). The negative logarithm of the likelihood is commonly taken, for simplicity and convenience. Defined this way, NLL is a proper score. For the special case of a Gaussian distribution, with $\sigma^2_{n,d}$ being the variance for N patterns with D dimensions, $y_{n,d}$ the true value and $\hat{y}_{n,d}$ the predicted value, the NLL function is given by

$$L(\mathbf{y}) = \sum_{n=1}^{N}\sum_{d=1}^{D}\frac{1}{2}\left(-\log(\sigma^2_{n,d}) - \frac{(\hat{y}_{n,d}-y_{n,d})^2}{\sigma^2_{n,d}}\right) + \frac{ND}{2}\log(2\pi). \tag{3.15}$$

3.3.2 Continuous Ranked Probability Score (CRPS)

Gneiting e Raftery (2007) state the Continuous Ranked Probability Score, a proper score designed for cumulative distributions. It was first designed to deal with CDFs with intractable PDFs. This score is much used for probabilistic forecasts. The lower the value, the better the probabilistic model performs.

Let $F(\mathbf{x})$ be a Cumulative Distribution Function of a given random variable $\mathbf{x}$. The CRPS of a particular value $\mathbf{y}$ of this variable over F is defined by

$$l(F,\mathbf{y}) = \int (F(\mathbf{x}) - 1_{\mathbf{x}\geq\mathbf{y}})^2 \mathrm{d}\mathbf{x}, \tag{3.16}$$

where the expression $1_{x\geq y}$ represents the value 1, if $x \geq y$, and 0 otherwise.

3.3.3 Root Mean Squared Error (RMSE)

Despite not being a probabilistic quantity, the Root Mean Squared Error (RMSE) metric is well known as one of the simplest ways to measure the difference between the predicted estimates and the true values of the data set. Let y_i be the output value for the pattern i and $\hat{y}_i$ the predicted output from the model for this pattern, the RMSE for a set of N patterns is the given by

$$RMSE = \sqrt{\sum_{i}^{N} \frac{(\hat{y}_i - y_i)^2}{N}}. \tag{3.17}$$

3.4 Calibration Graphs

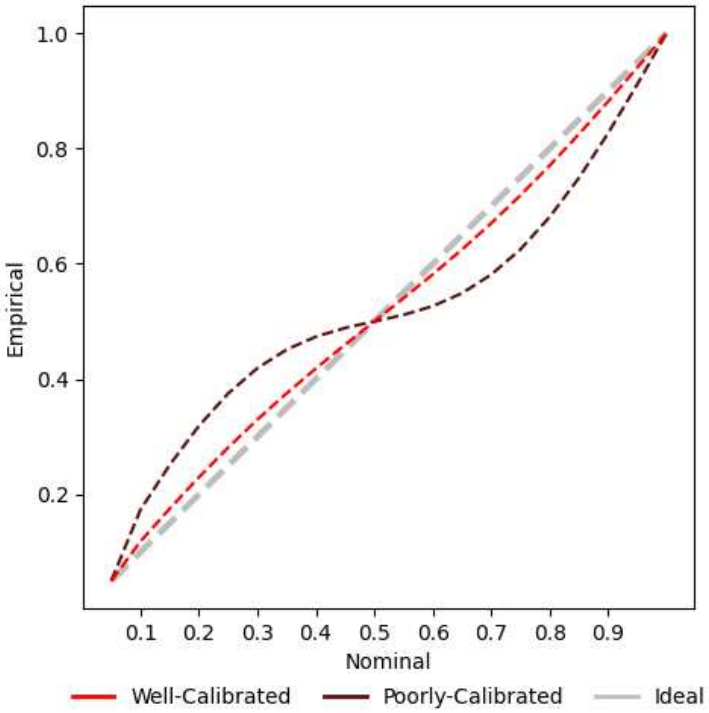
In statistics, calibrating predictive distributions means verifying uncertainty quantification through a frequentist approach over the test data set. A calibrated predictive distribution should ”follow” the true ground truth distribution of the real data. One can do this by using graphical methods to compare predictive and ”true” cumulative sample rates and verify the model behavior on a test data set.

3.4.1 Cumulative Sample Rates

Calibrating probabilistic forecasts means respecting the ”contract” between real data and its predictive distribution: if a given value is meant to be under $P-90$ percentile of the distribution, it is expected that it will indeed be there in 90% of the observations, for a large amount of unobserved data. It means the random values must follow the distribution that is assigned to them. If the predictive distribution is poorly calibrated, sample rates in each quantile is far from ideal, and the uncertainty estimates obtained from them are not trustful.

The Fig. 10 shows cumulative percentages of empirical data in each data percentile (KULESHOV *et al.*, 2018). Well-suited models would yield equal amount of samples in each equispaced quantile of the posterior distribution, as real data would do. It can be tested it by assuming data from test data set indeed represent the real data’s distribution, a frequentist approach. For example, if the model consistently yields biased samples, the graph would show a higher percentage being covered on the first percentiles. In terms of numbers, a biased model, for example, would yield 40% of the samples in the 20% percentile region of the test data set.

Figura 10 – Cumulative sample percentages, from real data, for each percentile. It is expected that real data matches the predictive distribution percentiles properly. The red line represent a well-calibrated predictive distribution, and the darker line a poorly-calibrated one.



Fonte: Elaborated by the author.

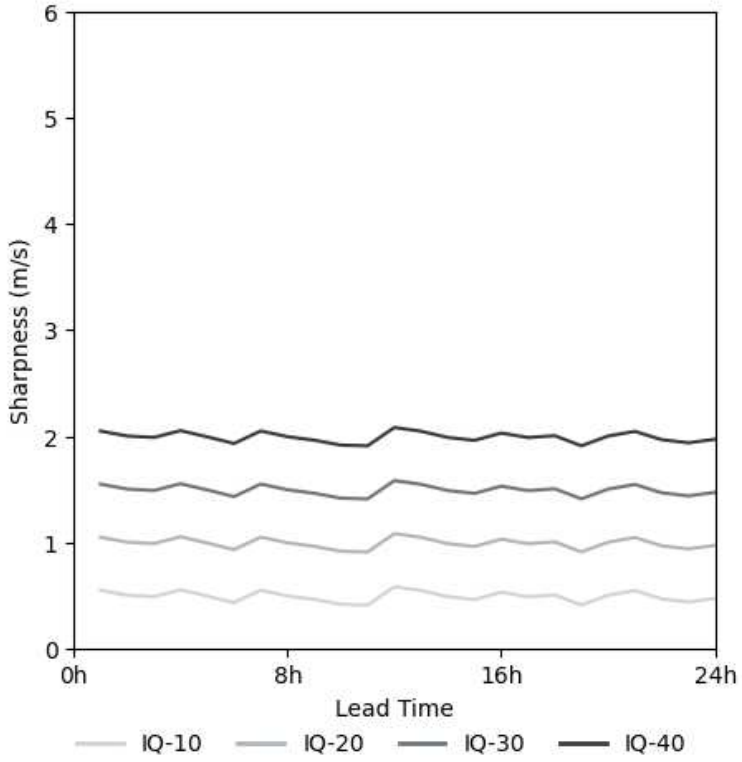
3.4.2 Sharpness × Lead Time

In the context of weather forecasting, one can argue predicted values could have a distribution varying along time. Let IQ-T be the interquantile distance between P-$(50+T)$ and P-$(50-T)$ percentiles, as follows:

$$IQ\text{-}10 = P\text{-}60 - P\text{-}40, \quad IQ\text{-}20 = P\text{-}70 - P\text{-}30, \ldots \tag{3.18}$$

This represent an “error range” of posterior samples expected to be between P-60 and P-40 values. The *sharpness* of a distribution means how concentrated the posterior probabilities are. A ”perfect” forecast would assign 100% probability to a single value, equal to the observed value. Sharp models have narrowly distributed predictions. These inter-quantile distances are a measurement of the sharpness of a predictive distribution. Fig. 11 shows how sharpness varies along the forecast lead time, which is the distance between forecast time and the occurrence of the phenomena that were predicted. Thin distribution, with low sharpness values, do not necessarily mean a calibrated predictor.

Figura 11 – This graph shows the sharpness of the posterior distribution, shown in terms of inter-quantile distances, over different forecast lead times.



Fonte: Elaborated by the author.

3.4.3 Error Plot

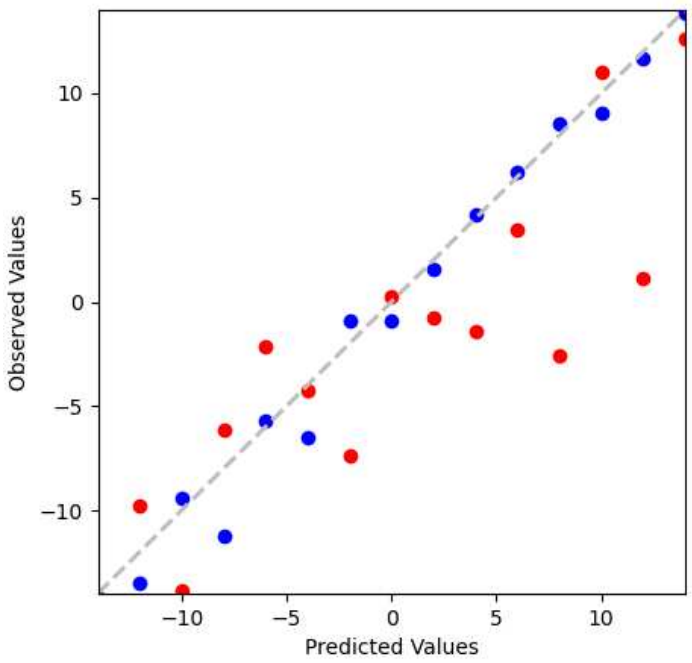
Another way to graphically present sharpness and calibration is plotting observed and predicted values, extracted from the test data set, in a scatter plot (TRAN *et al.*, 2020). The dispersion can show concentrated biases in a region of the data and the predictive distribution sharpness. Fig. 12 presents the difference between a sharp and calibrated model and one which is not. When dealing with a large amount of data dispersion, it is also worth adding binned confidence intervals summarizing the error distribution in each region of the data.

3.5 After All, Are These Methods Truly Bayesian?

Traditionally, Bayesian approach consists in considering a *prior* belief, usually by assuming a prior distribution over the parameter set, mathematically speaking. Point estimates are not used anymore, therefore a predictive distribution from the *posterior* of the parameters, obtained after observing the data.

One could question if the methods described in this chapter are actually Bayesian methods in the manner how they quantify uncertainty. This section argues in favor of each one

Figura 12 – This graph plots observed and predicted values, extracted from the test data set. Blue points represent a sharp and calibrated model, while red points show an erratic and biased model.



Fonte: Elaborated by the author.

of them.

The first one, Monte Carlo Dropout, is a method based on the dropout strategy, as stated before. Gal e Ghahramani (2016) argue it is actually comparable to a deep Gaussian Process (DAMIANOU; LAWRENCE, 2013), with no simplifying assumptions, which is a Bayesian machine learning method. It is possible to support it is indeed Bayesian, due to its nature.

In Wilson e Izmailov (2020), the authors add some discussion about academical criticism to Bayesian concept, in a way that some researchers undermine ”non-Bayesian” results, criticizing terminology the most, which does no good for the research itself. In Izmailov *et al.* (2018), the authors claim SWAG and MultiSWAG, extensions to Stochastic Weight Averaging (SWA), perform “*approximate fully Bayesian inference*”.

According to Lakshminarayanan *et al.* (2017), Deep Ensembles is a non-Bayesian approach to obtain predictive distributions. However, it can still be seen as a Bayesian Model Averaging technique (WILSON; IZMAILOV, 2020), and it is considered as a practical way to surpass the complexity of fully Bayesian Neural Networks. Thus, it can be seen as a Bayesian approximation (HOFFMANN; ELSTER, 2021).

Above all, this work uses the terminology “Bayesian uncertainty quantification”, a general term, not getting attached to any Bayesian method itself, a terminology which is used, for instance, by Gal *et al.* (2022). In that article, the authors also emphasize Monte Carlo Dropout (GAL; GHAHRAMANI, 2016) and Deep Ensembles (LAKSHMINARAYANAN *et al.*, 2017) as being approximate Bayesian methods.

3.6 Concluding Remarks

In this chapter, it is described the details behind the models that are going to be used, the Bayesian frameworks attached to them, and the metrics evaluated for comparison. Chapter 4 explains how to proceed in the task of forecasting wind speed, with the data gathered.

4 METHODOLOGY

This chapter details the machine learning problem that are going to be modeled, in terms of data definition, deep learning architectures, approximate Bayesian strategies, and evaluation metrics for comparison purposes.

4.1 Data Definition

To accomplish the goal of achieving short-term wind speed predictions, there are two main data sources: the wind farm's SCADA system and the publicly available NWP models. There are also general information from the park, like latitude and longitude for each Wind Turbine Generator (WTG). Combining these different data sources, one could have site-specific predictions for each turbine.

4.1.1 The Wind Farm's SCADA System

Every modern hard asset has several analog sensors measuring the most important operational variables. Among other variables, there are some important ones that can explain climate site-specific behavior. Regarding wind turbines, this huge amount of data is often stored as 10-minute timestamps in a Supervisory Control And Data Acquisition (SCADA) system (DANEELS, 1999). It is cataloged a data set including measurements of wind speed, wind direction, ambient temperature, pressure and humidity, for each turbine. Data is always being processed and stored, allowing the update of the model on a daily basis.

This study will be based on a wind farm containing eighteen turbines, localized at the South of Brazil. SCADA data cover two years of measurements for wind speed and wind direction variables. It is taken hourly measurements, down-sampled by picking always the first interval value to match NWP data format. Fig. 13 shows the turbine's relative positions.

4.1.2 Numerical Weather Forecast Systems

The models take advantage of already existent wide-spread Numerical Weather Prediction forecast models. These complex models are based on data stored from multiple measurement sites around the world, making them a great source of information to improve the solution. The most complete models include hourly predictions for wind speed, ambient

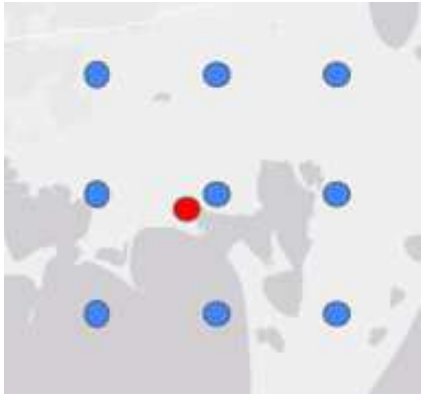
Figura 13 – There are 18 turbines in the wind farm focus of this work. The red points represent the location of each one of them.

Fonte: Elaborated by the author.

temperature, wind direction, rain and storm probabilities, and many other variables, for different heights and geographical locations.

For the purpose of this work, it is considered data collected from ERA5 historical data. Details of this data set were presented in Chapter 1. Data is collected from a 3×3 grid located around the same region of the park. Fig. 14 shows these positions relative to the wind farm.

Figura 14 – ERA5 provides historical data for a grid over the whole globe. In the photo, the blue points represent the subset used in this work. The red point is the wind farm's location.



Fonte: Elaborated by the author.

4.1.3 Pattern Building

Let k be the index to represent wind turbine WTG_k. $[t+1, t+L_{fut}]$ is the interval in the forecast future and $[t-L_{past}+1, t]$ the interval in the past used as input pattern to the model, with L_{past} and L_{future} being a length of time steps in the past and in the "future", that is, in the forecast at that moment. At each time t, $\mathbf{S}_t$ represents SCADA information, and $\mathbf{F}_t$ represents the global forecast model's information.

The model will be trained using wind speed in each direction as features. For each timestamp t in WTG_k, SCADA information, which consists of wind speed in each direction u (along latitude axis) and v (along longitude axis), is represented by a vector $\mathbf{S}_{k,t} \in R^{D_S}$, with D_S representing SCADA features dimension. The spatial coordinates lat and lon are added to this vector, for simplicity. Each of these vectors can be represented by the equation:

$$\mathbf{S}_{k,t} = [lat, lon, u_{k,t}, v_{k,t}]. \tag{4.1}$$

The global forecast source of data also has many different climate variables, such as wind speed components u and v, pressure p, temperature T, and humidity ρ in many different altitudes above sea level. The model training will use only wind speed information, for simplicity. The global forecast model's information u and v in the coordinate m of the grid at time t are represented by $u_{m,t}, v_{m,t}$, for example. At this point of the grid, the information vector is written as

$$\mathbf{F}_{m,t} = [u_{m,t}, v_{m,t}]. \tag{4.2}$$

Each batch of information from the global source of data will then have the shape

$$\mathbf{F}_t \in R^{D_F \times Lat \times Lon}. \tag{4.3}$$

where D_F is the climate variables dimension, and $Lat \times Lon$ represents the portion of the global grid used in the model.

In the experiments, for simplicity, since only wind speed information and geographic coordinates are used as features, $D_S = 4$ and $D_F = 2$.

Respecting the differences between the models and the data to be used, each modeling approach will have a different data conformation procedure, allowing them to use spatial and temporal aspects of the data when they are able to do this.

When it comes to the MLP model, which is neither temporal nor a spatial model, all spatial and temporal data are just flattened, turning them into features of the model. All turbine

Tabela 1 – Dimensions of the feature vectors, for each network. Notation $\nu(...)$ represents the resulting dimension when flattening the others.

Model	dim($\mathbf{S}$)	dim($\mathbf{F}$)
MLP	$\nu(L_{past}, D_S)$	$\nu(L_{fut}+L_{past}, D_F, Lat, Lon)$
LSTM	$\mathrm{L}_{past}\times\mathrm{D}_S$	$(\mathrm{L}_{fut}+L_{past})\times\nu(D_F, Lat, Lon)$
ConvLSTM	$\mathrm{L}_{past}\times\mathrm{D}_S$	$(\mathrm{L}_{fut}+L_{past})\times\mathrm{D}_F\times Lat\times Lon$

Fonte: Elaborated by the author.

and forecast data can be concatenated before applying the network, as they now have the same dimensions. LSTM model, in opposition to it, has short-memory temporal modeling capacities, so only spatial information from forecast model has to be flattened into features. When it comes to the Convolutional LSTM Network, both temporal and spatial dimensions of the NWP forecast data set are kept. The turbine SCADA data, however, do not have any spatial dimensions the way they have been described (latitude and longitude information were put as features, as stated before), forcing the model to use separately two neural networks this time: a LSTM network will be applied on the turbine's data, and a ConvLSTM layer will be applied separately in the forecast data. All of the three schemes will have a final fully-connected dense layer after network layers. The dimensions for S and F, for each pattern, are shown in Table 1, where notation $\nu(...)$ represents the resulting dimension after flattening others, as described above.

The goal is to yield wind speed forecasts $\mathbf{y}_t$, in both directions u_t, v_t, for each timestamp t in the interval $[t+1, t+L_{fut}]$ in *future* time, using data from $[t-L_{past}, t]$, which is in the *past*. It is important to observe that there are information from SCADA referring to the *past*, while Numerical Weather Prediction (NWP) model's information exists both in the *past* and the *future*. It is important to emphasize that there is no information leakage, as NWP future information is a prediction used as a feature. In order to add global forecast information to enhance the models, all $[t-L_{past}, t+L_{fut}]$ interval from the forecast will be included as input for timestamp t.

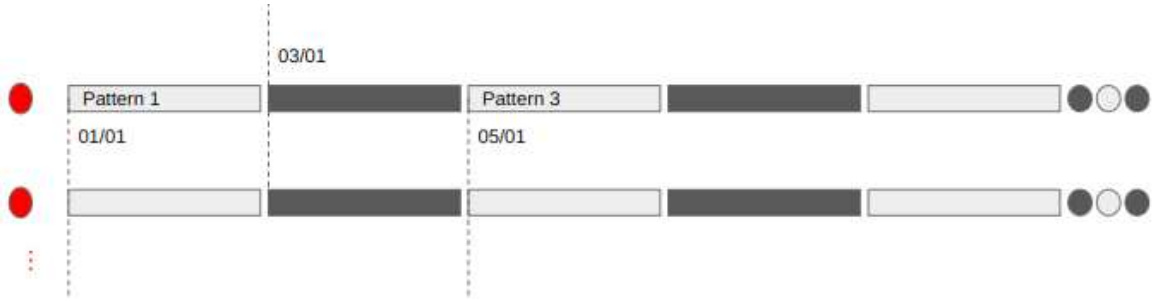
The data can be expressed in terms of these vectors. Each forecast output $\mathbf{Y} \in \mathbb{R}^{L_{fut}\times 2}$ can be defined as matrices, as in Eq. (4.4). The dimensions for input patterns are shown in Table 1, as described above.

$$\mathbf{y}_k(t) = \begin{bmatrix} u_{k,t+1} & v_{k,t+1} \\ \dots & \dots \\ u_{k,t+L_{fut}} & v_{k,t+L_{fut}} \end{bmatrix}^{L_{fut}\times 2} . \tag{4.4}$$

4.2 Training Scenario

For this work's purpose, it is stated one day (24 hourly steps) for past data and one day for future data. In order to take non-intersected, consecutive periods in time, the interval of the study is partitioned into many 2-day pieces, each one of them representing a pattern and its output. This way, the full data set will have around 182 patterns per turbine per year, depending on data quality, because the data sources may have invalid values to be removed.

Figura 15 – The red points represent different turbines. For each turbine, the daily partitioned time series is shown. 01/01, 03/01 and 05/01 are dates in January.



Fonte: Elaborated by the author.

Four turbines will be separated to validate the model, while the others will be used to train the model. Fig. 16 illustrates this split. One of the validation turbines is taken at the top of the farm on purpose, to include patterns affected by wind at the border.

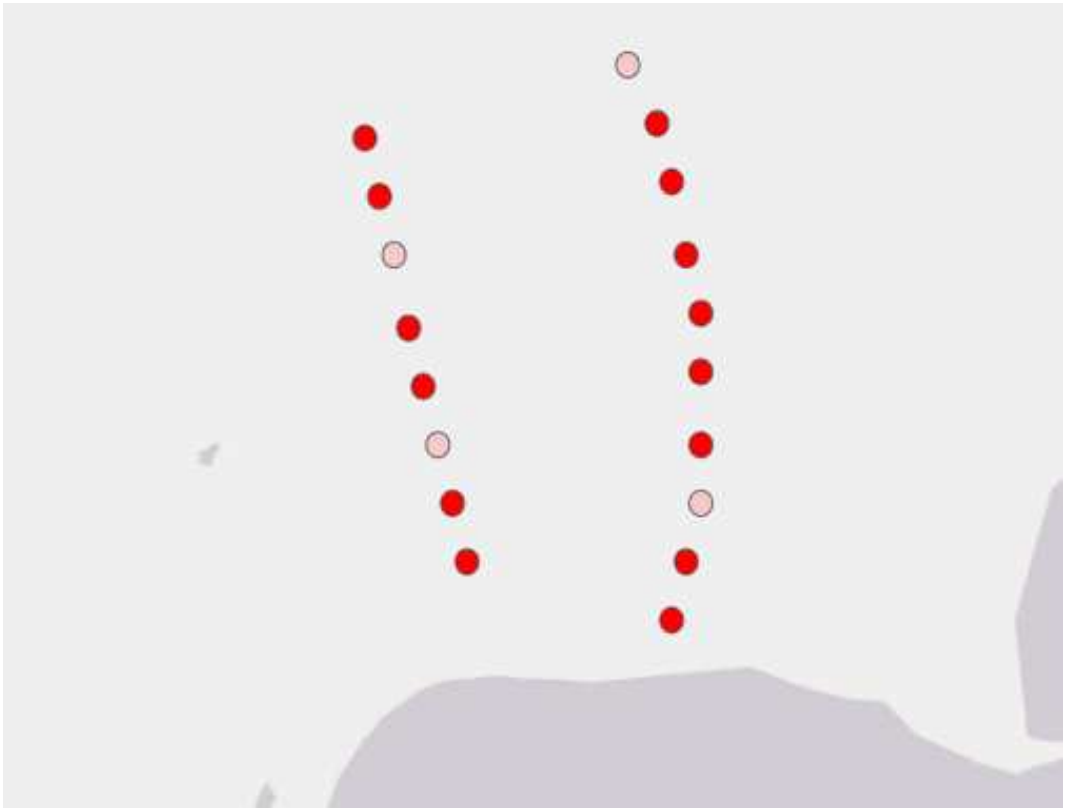
4.3 Neural Network Architectures

Three different neural networks are considered to capture the data behavior: MLP, LSTM and ConvLSTM. It is expected that the ConvLSTM performs better, capturing both temporal and spatial aspects of the data. As follows, the model's configuration is detailed. Bayesian inference, with the purpose of extracting uncertainty estimates, will be also applied using Monte Carlo Dropout, Stochastic Weight Averaging - Gaussian, Multi Stochastic Weight Averaging - Gaussian and Deep Ensembles inference methods.

There will also be a simple NLL-trained model, as a baseline for uncertainty quantification comparison. It is trained similarly to one of the models from the Deep Ensemble, which are trained following Nix e Weigend (1994) guidelines. This model is trained without any adversarial training technique, however.

All the Bayesian models, except the Deep Ensembles, will be pre-trained with

Figura 16 – The soft red points represent the turbines used to validate the model in the Turbine-Driven data set. Note that they are spread along the area.



Fonte: Elaborated by the author.

regular MSE loss. Deep Ensembles have to be trained using the NLL loss due to the ensembling technique used, based on Lakshminarayanan *et al.* (2017) and Nix e Weigend (1994). All models yield the same output architecture shown in (4.4), except Deep Ensembles and the NLL baseline, which models two output time series, both dimensionally equal to (4.4), but with the second one standing for the output's variances (see subsection 3.2.2). Deep Ensembles will also follow the guidelines described in Subsec. 3.2.2, using adversarial training, as suggested.

It is included one more model, called Dummy, that will "forecast" wind speed by just copying the past data to the future. This model will be used only as a naive baseline for the RMSE metric of the main neural networks.

4.4 Evaluation Metrics

Evaluation metrics RMSE, NLL and CRPS are applied in order to compare the results of the different models in the both train and validation data sets. The last two, which are probabilistic metrics, are expected to show lower values to the models which generalize best at the domain space. There are also some calibration plots. Section 3.3 has more details on all

these comparison metrics and techniques.

This chapter described how the models are designed. Chapter 5 shows the results obtained from the experiments with respect to its different model strategies and data set configurations.

5 RESULTS

This chapter is divided in four sections. The first one discusses the training phase and states some general additional information about the models. The second subsection shows the results obtained from the models, which was described in Chapter 4. This part comprises the models results, including how they compare to each other in terms of probabilistic and deterministic metrics and calibration graphics. The third subsection shows practical examples of how to use the model, showcasing them with a hypothetical situation in a wind farm operational context. The fourth is an experiment to see how the model would perform if information from only one source, either NWP global forecasts or SCADA data, was provided.

5.1 Model Training

In order to prevent overfitting the data, the training phase of these models included weight decay and dropout as regularization strategies. Dropout, as well as being a method of sampling for Bayesian approximate inference and uncertainty quantification, is used here to regularize (SRIVASTAVA *et al.*, 2014) all networks parameters as well, independently on which Bayesian inference method they are aiming to perform. Weight decay is also applied in all training procedures, performing *L2* regularization and preventing overfitting (BISHOP, 2006). In the process of choosing the best models, hyperparameter tuning was done by using grid search strategy, applied to dropout rates (with values varying from 0.15 to 0.5 dropout rate (GAL; GHAHRAMANI, 2016)), hidden layers size (with values varying between 24, 48 and 96 neurons) and SWAG learning rates (with values varying from 0.01 to 0.5 for learning rate, based in Maddox *et al.* (2019) guidelines).

Now, it is time to determine which one of the model combinations got better results and discuss about these results and the practical situations in which they can be evaluated, in terms of the metrics defined to compare them. The combinations are made by choosing:

1. The machine learning model to train, chosen between MLP, LSTM, ConvLSTM and the Dummy model [1];
2. The Bayesian model used to extract posterior uncertainty estimates, chosen between Deep Ensembles, MC Dropout, SWAG, MultiSWAG, and the NLL-trained model (see Sec. 4.3 for details).

[1] The Dummy model is not probabilistic, thus, it cannot be compared using probabilistic metrics. However, it can still be calculated the RMSE metric to compare it to the other models.

Tabela 2 – Metrics for all evaluated models. As CRPS was chosen to be the main value to compare the forecasts, the ConvLSTM network along with the Deep Ensembles strategy got the best results. Models are trained multiple times, and the standard deviation of each metric is attached.

Model	Bayesian Framework	CRPS	NLL	RMSE
ConvLSTM	MC Dropout	0.925 ± 0.004	1.447 ± 0.018	$\mathbf{1.728 \pm 0.005}$
	Deep Ensembles	$\mathbf{0.923 \pm 0.001}$	$\mathbf{0.979 \pm 0.001}$	1.981 ± 0.003
	MultiSWAG	1.116 ± 0.014	1.740 ± 0.039	2.065 ± 0.030
	NLL Baseline	1.160 ± 0.006	1.281 ± 0.004	2.339 ± 0.011
	SWAG	1.221 ± 0.031	1.847 ± 0.046	2.245 ± 0.064
LSTM	MC Dropout	1.303 ± 0.007	2.501 ± 0.018	2.301 ± 0.010
	Deep Ensembles	1.295 ± 0.003	1.455 ± 0.003	2.667 ± 0.013
	MultiSWAG	1.552 ± 0.026	2.207 ± 0.053	2.871 ± 0.060
	NLL Baseline	1.420 ± 0.017	1.592 ± 0.020	2.816 ± 0.046
	SWAG	1.605 ± 0.050	2.314 ± 0.074	2.959 ± 0.093
MLP	MC Dropout	1.620 ± 0.085	3.091 ± 0.023	2.801 ± 0.123
	Deep Ensembles	1.370 ± 0.003	1.529 ± 0.003	2.797 ± 0.004
	MultiSWAG	1.615 ± 0.018	2.751 ± 0.089	2.898 ± 0.034
	NLL Baseline	1.617 ± 0.018	1.841 ± 0.033	3.134 ± 0.039
	SWAG	1.682 ± 0.031	2.849 ± 0.168	3.003 ± 0.056
Dummy	—	—	—	6.122

Fonte: Elaborated by the author.

5.2 Model Results

Table 2 present the metrics for all evaluated models. The metrics shown are extracted as an average among all output data for test data set. The models were trained multiple times, so as all metrics are presented with an error estimate, which is represented by the standard deviation. The symbol $\pm$ is used here to separate the metrics and their deviations. All evaluation metrics are calculated on the real scaled data set. The best results for each metric are highlighted with bold letters.

In addition to the numerical model comparison, which can be done with the metrics scalar values, the calibration graphs in Fig. 17 present cumulative sample rates, compared to the ground truth values in test data set, and models sharpness over the lead time for predictions. Note that a sharper model is not necessarily well-calibrated. It is also worth noting a trend of uncertainty increasing along with the lead time.

Fig. 18 presents a graph showing error dispersion between real and predicted values. 95% confidence intervals are shown for binned real values. It indicates ConvLSTM models are more well-calibrated than other models, especially when paired with the Deep Ensemble strategy. The predicted values are closer to the real ones, and the model biases are much lower than the ones yielded by MLP and LSTM, especially in extreme values of wind. Explanations about these

calibration graphs were detailed in Section 3.4.

As expected, the three neural networks yielded much lower RMSE values than the Dummy predictor. ConvLSTM neural network achieved better results, in general, both for predictive and point estimate metrics. LSTM metrics were lower than MLP ones, too. Adding temporal and spatial capabilities to the model helped to enhance their generalization capacity, proving the benefits of using convolutional recurrent neural networks.

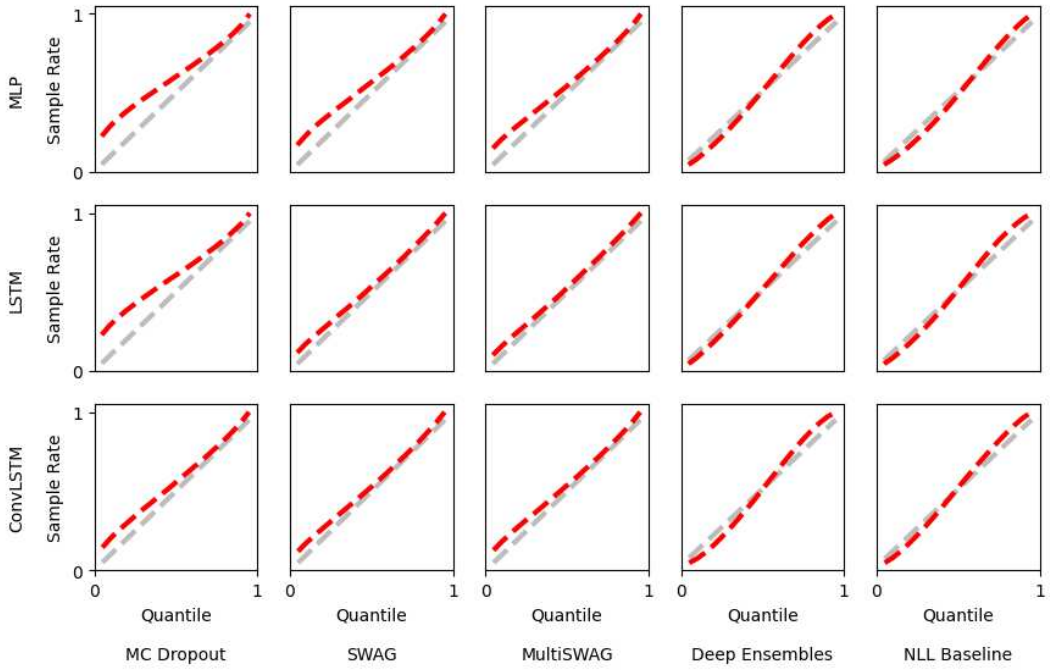
Take CRPS to be the main metric, for comparison purposes. It is designed to measure the difference between the whole predictive distribution and empirical CDFs. It is also a common choice to compare predictive forecasts. Deep Ensembles framework yielded better predictive generalization at the validation data set, due to the lower CRPS. Comparing all these combinations, the one that generalizes the best the data being modelled is the ConvLSTM combined with Deep Ensembles Bayesian uncertainty quantification method. The network also had lower NLL values. MC Dropout method, however, achieved lower RMSE metric.

One unexpected behavior was the NLL-trained baseline achieving better results than some Bayesian approximate methods, such as MultiSWAG and MC Dropout. It is also seen that models using Monte Carlo Dropout, SWAG and MultiSWAG got generally higher probabilistic metrics and their calibration curves (see Fig. 17) are not as well-suited as the NLL-trained models, especially the one using Deep Ensembles. This can be associated to the fact that MC Dropout, SWAG and MultiSWAG training processes aim to lower RMSE metric and Deep Ensembles and the NLL baseline aims to lower NLL metric. That seems to be more suited for the problem, since wind speed probability distributions if often modeled with a Weibull curve, which is quite similar to a Gaussian distribution (JENKINS, 2021).

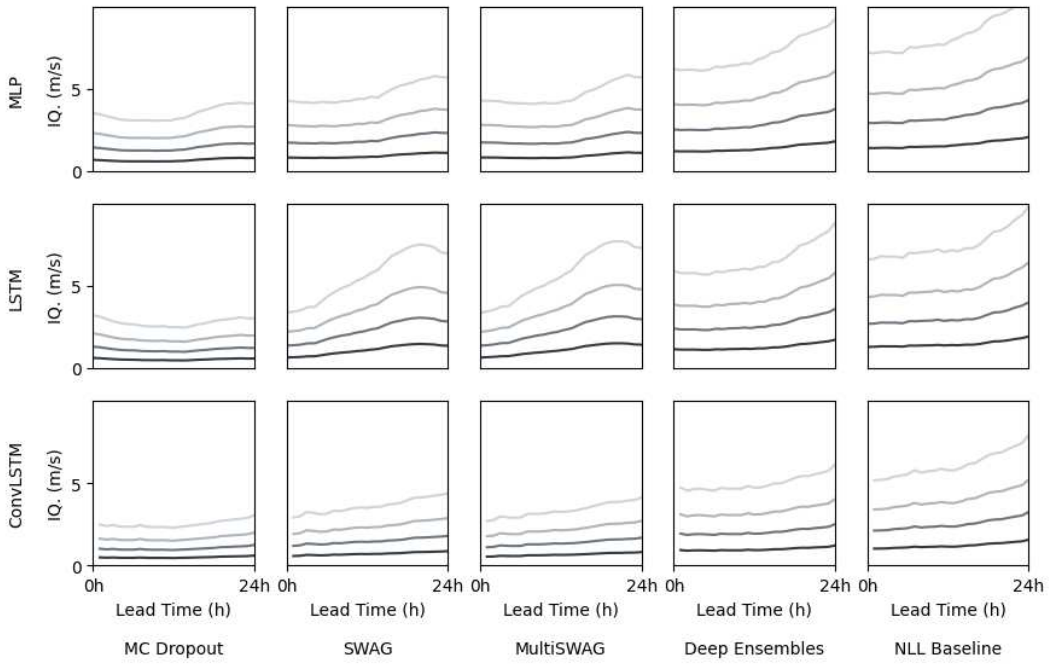
Finally, Fig. 17 also shows how sharpness curves vary over the lead time of the predictive distribution. NLL-trained models, which have performed better in terms of probabilistic metrics comparison, have wider distributions in the end of the lead time. This means the models indeed vary their uncertainty along the time period. Deep Ensembles also got wider predictive distributions (represented by higher inter-quantile intervals) than SWAG, MultiSWAG and MC Dropout, which is not a bad thing, since the probabilistic metrics are lower, demonstrating better fitting to the test data set.

Now, there will be presented some practical applications for the models, in the context of maintenance scheduling in a wind farm.

Figura 17 – The two graphs measure the posterior predictive distribution calibration with respect to the test data set.



(a) Cumulative sample rates, yield from predictive distribution, compared to theoretical sample rates in each quantile. ConvLSTM models trained with Deep Ensembles framework are the most well-calibrated models.



(b) Sharpness measurements for each lead time, with respect to P-60, P-70, P-80 and P-90 percentiles. Note that a sharper model do not necessarily is well-calibrated. It is also seen a trend to increase uncertainty over the lead time.

Fonte: Elaborated by the author.

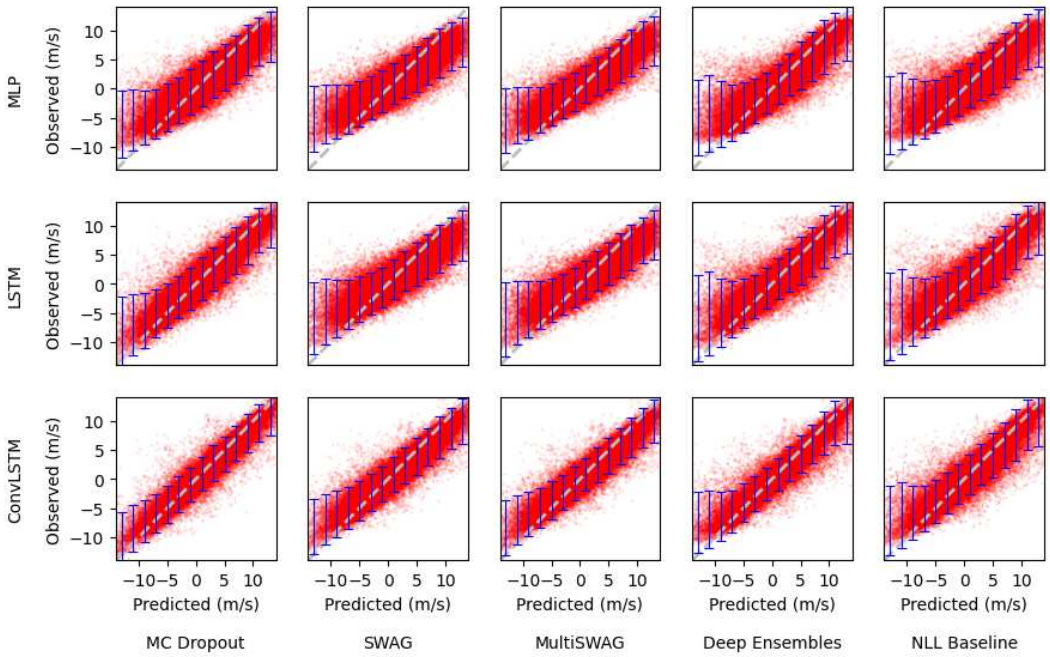
5.3 Practical Application in Maintenance Scheduling

In order to showcase the model into one of the main applications of wind speed forecasting, be a hypothetical situation of a maintenance scheduling application of the models. There are many other applications for wind forecasting, as mentioned in Chapter 1. In order to graphically see the difference in the models discussed in Sec. 5.2, forecasts from each Bayesian model will be shown. Models from ConvLSTM network architecture are taken for this matter.

During the windy season, one concerned manager needs to know if there will be any good window to schedule a quick maintenance to one or more turbines of the complex. In order to be sure about the decision to provide personnel and equipment for the job during the daily hours, that manager must be confident about that window. In other words, he/she waits for the wind to be around a specific secure level, with 90% of certainty that the wind will not be higher than that, because the work could not be done in worse circumstances.

His/her staff engineer decided to use this work to model and predict the wind, using data extracted from both SCADA system and NWP models. The model yielded a wind speed time series for the wind farm, predicted for the next day, along with an 80% confidence interval for uncertainty estimates, whose upper and lower boundaries are P-10 and P-90, respectively. The engineer then answered him/her, based on Fig. 19, that the wind may be lower by the

Figura 18 – The plot shows the dispersion graphs of real value versus predicted value, for each model. The bars in blue show 95% confidence intervals in each binned real value.



Fonte: Elaborated by the author.

afternoon, and 12pm should be a good time to start the intervention.

The manager found the model very useful. He/she asked the engineer if he could help him/her with another question: what would be the best turbine to stop for intervention? The engineer then showed him/her Fig. 20, a forecast map for each turbine in a specific time, arguing the most impacted turbine, in terms of frustrated generation, would be the further southwestern one, and the lowest impacted would be the further northeastern one, on the opposite side, by the time they chose to begin the intervention.

That was an example of how a manager could use the model to accurately plan a maintenance intervention in wind farm's operational context. Now, it is presented an experiment of the deployment using the same modeling process, but without adding any information from one of the data sources to the model. The aim is to see the effects of taking off either NWP or SCADA information from the model.

5.4 How Important Is It to Attach Each Source of Information to the Model?

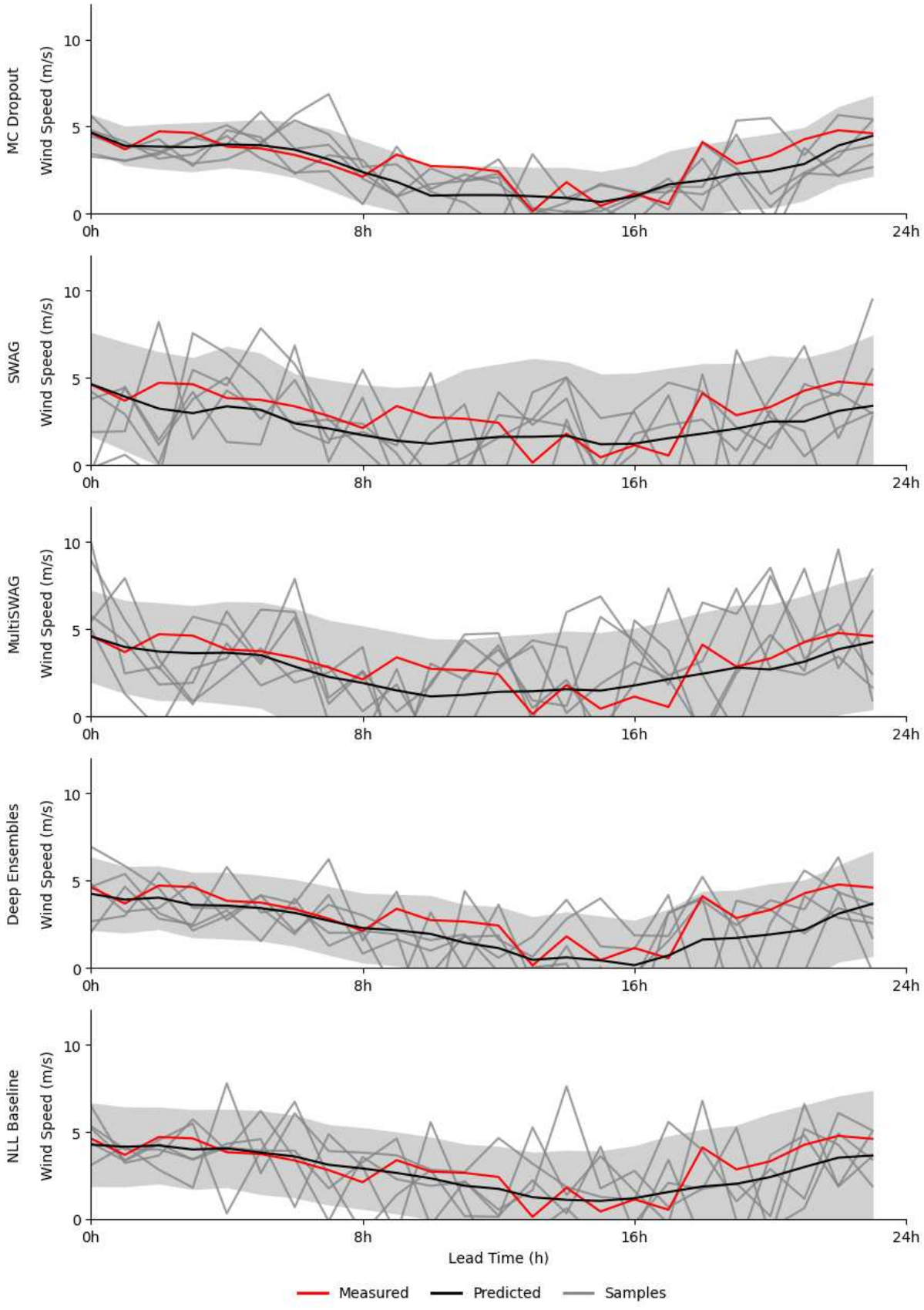
The results of the modeling process were shown, using both SCADA site-specific data and NWP global forecast data. Wind speed forecasting has been modelled with many network architectures, quantifying uncertainty taking advantage of Bayesian uncertainty quantification techniques.

But how important is it to add each data source to the model? Does the SCADA data, with wind measurements from the wind farm, bring enough information to predict the future patterns? Opposed to that, could the NWP global forecast data be able to yield a well-fitted predictive distribution by itself?

In order to answer those questions, an additional experiment is designed by simplifying the networks. An LSTM will be trained only with the turbine's SCADA data source. No information from global forecasts will be added to the model. For the NWP only model, a ConvLSTM will be trained with only global forecast information as input, as this network is able to attach spatial capabilities to this model. Table 3 shows the metric results for this experiment. The performance of the model is notably worse, compared to the models using both site-specific SCADA data and NWP global forecast data, presented in Section 5.2.

Fig. 21 shows examples of forecasts produced by this model. The Bayesian uncertainty quantification is extracted using Deep Ensembles. Observe that the predictive distributions obtained are much wider than the ones of the complete model.

Figura 19 – Wind speed forecasts for the turbine in south-east, in black, with 80% Confidence Interval, compared to real data in red. In light grey, samples yielded from the predictive distribution. Lower boundary is equivalent to P-90 wind speed. The examples were obtained from ConvLSTM networks for all Bayesian uncertainty quantification methods.



Fonte: Elaborated by the author.

Figura 20 – Wind speed forecasts for each turbine in red, compared to ground truth ones in black. In light grey, samples yielded from the predictive distribution. Example extracted from ConvLSTM neural network, along with Deep Ensembles Bayesian framework.

Fonte: Elaborated by the author.

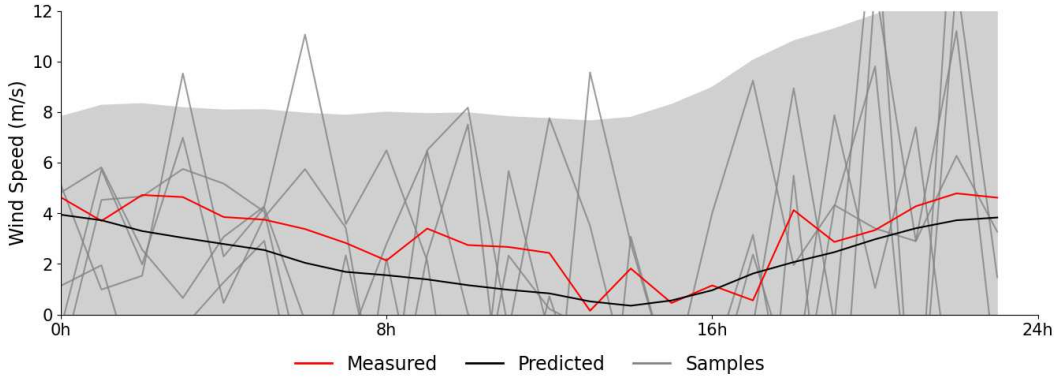
Tabela 3 – Performance metrics when using: (1) only WTG data to train the model. (2) only NWP data to train the model. ConvLSTM results are also shown for a quick reference. The performance of these models are significantly lower compared to the scenario where both sources of information are combined.

Model	**Bayesian Framework**	**CRPS**	**NLL**	**RMSE**
LSTM^1_{WTG}	MC Dropout	2.018 ± 0.034	3.041 ± 0.268	3.169 ± 0.046
	Deep Ensembles	1.807 ± 0.016	2.147 ± 0.020	3.543 ± 0.033
	MultiSWAG	1.787 ± 0.036	2.786 ± 0.016	3.306 ± 0.071
	NLL Baseline	2.164 ± 0.018	2.611 ± 0.038	4.102 ± 0.030
	SWAG	2.104 ± 0.057	3.090 ± 0.133	3.901 ± 0.086
ConvLSTM^2_{NWP}	MC Dropout	1.645 ± 0.077	2.475 ± 0.130	2.787 ± 0.087
	Deep Ensembles	1.447 ± 0.001	1.588 ± 0.001	2.823 ± 0.002
	MultiSWAG	1.903 ± 0.013	2.530 ± 0.032	2.243 ± 0.021
	NLL Baseline	1.610 ± 0.012	1.752 ± 0.022	3.058 ± 0.029
	SWAG	2.023 ± 0.016	3.523 ± 0.067	3.478 ± 0.033
ConvLSTM	MC Dropout	0.925 ± 0.004	1.447 ± 0.018	$\mathbf{1.728 \pm 0.005}$
	Deep Ensembles	$\mathbf{0.923 \pm 0.001}$	$\mathbf{0.979 \pm 0.001}$	1.981 ± 0.003
	MultiSWAG	1.116 ± 0.014	1.740 ± 0.039	2.065 ± 0.030
	NLL Baseline	1.160 ± 0.006	1.281 ± 0.004	2.339 ± 0.011
	SWAG	1.221 ± 0.031	1.847 ± 0.046	2.245 ± 0.064

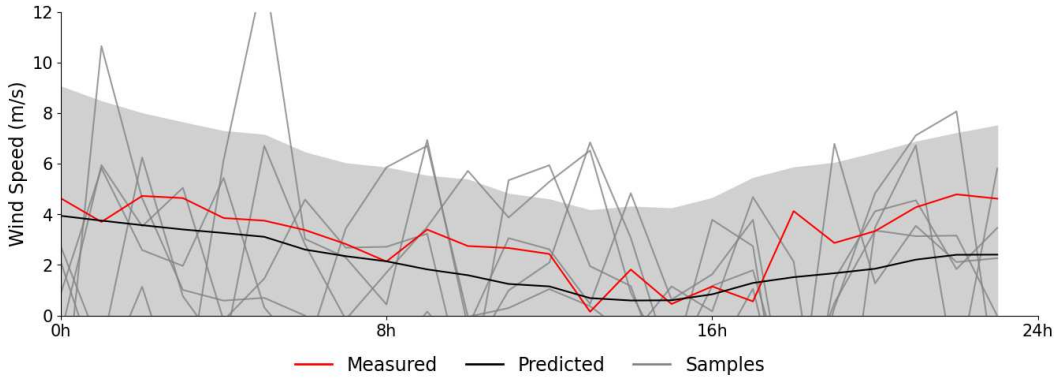
Fonte: Elaborated by the author.

This chapter presented the results achieved with the models. In Chapter 6 this work is concluded with some final comments and proposals for future work.

Figura 21 – The two graphs shows the same application example with the simplified models. The posterior predictive distribution is much wider and less accurate than the model trained with both data sources.



(a) Forecasts from The Model trained using only SCADA data.



(b) Forecasts from The Model trained using only NWP data.

Fonte: Elaborated by the author.

6 CONCLUSION AND FURTHER WORK

Weather forecasting has several different applications. In the context of wind energy, wind speed forecasts are particularly important, having many applications, such as maintenance scheduling, generation predictability and short-term energy trading markets. Enhancing these forecasts with uncertainty estimates is also very important, helping managers to reduce risks in any operational decisions. This chapter summarizes this work, highlighting the most important results, and presents some proposals for future work in this context.

6.1 Summary of Results

This work proposed a data-driven spatio-temporal approach to wind speed forecast prediction by combining general Numerical Weather Prediction global forecast data and local data collected from turbines' Supervisory Control And Data Acquisition (SCADA) systems. It has shown that training a Convolutional LSTM Network (ConvLSTM) neural network combined with the Deep Ensembles method achieved the best probabilistic performance metrics and calibration, compared to other network architectures and Bayesian frameworks.

The main motivation behind this work was to explore how spatio-temporal characteristics of the wind farm and global forecasts could be attached to the models. By combining these two data sources, one can model spatio-temporal behaviour and differentiate predictions between turbines.

Also, it was exemplified two practical applications of the model, using common situations in the daily operation of a wind energy farm. One could use this model to assist scheduling maintenance interventions and deciding which would be the best turbine to do the work, based on wind resource forecasts.

It is also shown that combining both global forecast NWP and site-specific SCADA data is of fundamental importance to enhance the model's performance metrics, since models with only one of these sources achieve worse performance than the complete ones.

It is worth pointing some limitations of this work. Since the pattern building process takes many temporal and spatial data, the models wound up having few patterns to train and test, forcing the deployments to gather many years of data to increase this number, especially when the model yields longer forecasts. Besides, since the model depends on SCADA data to be applied, low data quality in the wind farm's historian systems could restrict the model

application.

6.2 Future Work

Future work in this area could further explore the relation between the wind speed distribution and the Weibull distribution (JENKINS, 2021), for example training Deep Ensembles models using NLL with respect to a Weibull curve, or approximating the posterior distribution with a Weibull instead of a Gaussian distribution. Some concerns would be necessary when modeling wind speed latitude and longitude components through a Weibull distribution, however, as these variable can reach negative values too. It would also be interesting to compare Deep Ensembles results to Bayesian Neural Networks that use sampling techniques to yield a posterior distribution, such as Hamiltonian Monte Carlo (WILSON; IZMAILOV, 2020). Adding Laplace approximation methods to the methods being compared (DAXBERGER *et al.*, 2021) can be considered too. Finally, another possibility is comparing the training strategies to networks trained using probabilistic backpropagation (HERNANDEZ-LOBATO; ADAMS, 2015), which another Bayesian probabilistic method applied to neural networks by changing the training phase.

REFERENCES

A. SANANDAJI, B. M. G. Deep forecast: Deep learning-based spatio-temporal forecasting. *Journal of Artificial Intelligence*, JAIR, 2022.

ABDAR, M.; POURPANAH, F.; HUSSAIN, S.; REZAZADEGAN, D.; LIU, L.; GHAVAMZADEH, M.; FIEGUTH, P.; CAO, X.; KHOSRAVI, A.; ACHARYA, U. R.; MAKARENKOV, V.; NAHAVANDI, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. **Information Fusion**, Elsevier, v. 76, p. 243–297, 2021. ISSN 1566-2535. Disponível em: https://www.sciencedirect.com/science/article/pii/S1566253521001081.

ABE, T.; BUCHANAN, E. K.; PLEISS, G.; ZEMEL, R.; CUNNINGHAM, J. P. Deep ensembles work, but are they necessary? **Advances in Neural Information Processing Systems**, United States, v. 35, p. 33646–33660, 2022.

ACADEMY, D. S. **Deep Learning Book**. United States: [*S. n.*], 2022. Disponível em: https://www.deeplearningbook.com.br/.

ALZUBAIDI, L.; ZHANG, J.; HUMAIDI, A. J.; AL-DUJAILI, A.; DUAN, Y.; AL-SHAMMA, O.; SANTAMARíA, J.; FADHEL, M. A.; AL-AMIDIE, M.; FARHAN, L. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. **Journal of Big Data**, United States, v. 8, p. 53, Mar 2021. ISSN 2196-1115. Disponível em: https://doi.org/10.1186/s40537-021-00444-8.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006.

BOCHENEK, B.; USTRNUL, Z. Machine learning in weather prediction and climate analyses—applications and perspectives. **Atmosphere**, United States, v. 13, 01 2022.

BOURLARD, H.; WELLEKENS, C. Speech pattern discrimination and multilayer perceptrons. **Computer Speech Language**, United States, v. 3, n. 1, p. 1–19, 1989. ISSN 0885-2308. Disponível em: https://www.sciencedirect.com/science/article/pii/0885230889900119.

BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J. D.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. Language models are few-shot learners. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.; LIN, H. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2020. v. 33, p. 1877–1901. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

CASELLA, G.; BERGER, R. L. **Statistical Inference**. Belmont, CA: Duxbury Press, 2002.

CHEN, L.; YAO, P.; LIU, Y.; LI, G. Price composition and prediction of renewable energy in a free energy market. **Energy**, Elsevier, United States, v. 157, p. 9–17, 2018.

COEN, J. **Weather Forecasting: Wildfire Weather**. Uninted States: Academic Press, 2014. 323-331 p.

COUNCIL, G. W. E. **Global Wind Report**. 2021. Disponível em: https://gwec.net/global-wind-report-2022.

COUNCIL, N. R. **A Safer Future: Reducing the Impacts of Natural Disasters**. Washington, DC: The National Academies Press, 1991. ISBN 978-0-309-04546-9. Disponível em: https://nap.nationalacademies.org/catalog/1840/a-safer-future-reducing-the-impacts-of-natural-disasters.

DAMIANOU, A.; LAWRENCE, N. D. Deep gaussian processes. **Artificial intelligence and statistics**, PMLR, United States, p. 207–215, 2013.

DANEELS, A. What is scada? **International Conference on Accelerator and Large Experimental Physics Control Systems**, United States, 1999.

DAXBERGER, E.; KRISTIADI, A.; IMMER, A.; ESCHENHAGEN, R.; BAUER, M.; HENNIG, P. Laplace redux - effortless bayesian deep learning. In: BEYGELZIMER, A.; DAUPHIN, Y.; LIANG, P.; VAUGHAN, J. W. (Ed.). **Advances in Neural Information Processing Systems**. US: MIT Press, 2021.

DESAI, P.; SUJATHA, C.; CHAKRABORTY, S.; ANSUMAN, S.; BHANDARI, S.; KARDIGUDDI, S. Next frame prediction using convlstm. **Journal of Physics: Conference Series**, IOP Publishing, United States, v. 2161, n. 1, p. 012024, jan 2022. Disponível em: https://dx.doi.org/10.1088/1742-6596/2161/1/012024.

ECMWF. **ERA5 reanalysis dataset, European Centre for Medium-Range Weather Forecasts**. United Kingdom: [*S. n.*], 2021. [Online]. Disponível em: https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5.

ESPEHOLT L., A. S. S. C. e. a. **Deep learning for twelve hour precipitation forecasts**. Nature, 2022. Disponível em: https://doi.org/10.1038/s41467-022-32483-x.

ESTEVA, A.; ROBICQUET, A.; RAMSUNDAR, B.; KULESHOV, V.; DEPRISTO, M.; CHOU, K.; CUI, C.; CORRADO, G.; THRUN, S.; DEAN, J. A guide to deep learning in healthcare. **Nature medicine**, Nature Publishing Group, United States, v. 25, n. 1, p. 24–29, 2019.

GAL, Y.; GHAHRAMANI, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: **International Conference on Machine Learning**. New York, US: MIT Press, 2016. p. 1050–1059.

GAL, Y.; KOUMOUTSAKOS, P.; LANUSSE, F.; LOUPPE, G.; PAPADIMITRIOU, C. Bayesian uncertainty quantification for machine-learned models in physics. **Nature Reviews Physics**, United States, v. 4, n. 9, p. 573–577, 2022.

GAWLIKOWSKI, J.; TASSI, C. R. N.; ALI, M.; LEE, J.; HUMT, M.; FENG, J.; KRUSPE, A.; TRIEBEL, R.; JUNG, P.; ROSCHER, R.; SHAHZAD, M.; YANG, W.; BAMLER, R.; ZHU, X. X. **A Survey of Uncertainty in Deep Neural Networks**. 2021.

GELMAN, A.; CARLIN, J. B.; STERN, H. S.; RUBIN, D. B. **Bayesian Data Analysis**. London, UK: Chapman & Hall/CRC, 2003.

GERON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. California, US: O'Reilly Media, Inc., 2019.

GNEITING, T.; RAFTERY, A. E. Strictly proper scoring rules, prediction, and estimation. **Journal of the American statistical Association**, Taylor & Francis, United States, v. 102, n. 477, p. 359–378, 2007.

GOLDSTEIN, M. **The Complete Idiot's Guide to Weather**. Alpha Books, 2002. (Complete Idiot's Guide to). ISBN 9780028643410. Disponível em: https://books.google.com.br/books?id=mHbKk4xFzigC.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Massachussets, US: MIT Press, 2016. http://www.deeplearningbook.org.

GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. **Explaining and harnessing adversarial examples**, Cornell University Library, Cornell University, Ithaca, NY, 2015.

HASAGER, C. B.; NYGAARD, N. G.; VOLKER, P. J. H.; KARAGALI, I.; ANDERSEN, S. J.; BADGER, J. Wind farm wake: The 2016 horns rev photo case. **Energies**, United States, v. 10, n. 3, 2017. ISSN 1996-1073. Disponível em: https://www.mdpi.com/1996-1073/10/3/317.

HE, M.; YANG, L.; ZHANG, J.; VITTAL, V. A spatio-temporal analysis approach for short-term forecast of wind farm generation. **IEEE Transactions on Power Systems**, United States, v. 29, n. 4, p. 1611–1622, 2014.

HERNANDEZ-LOBATO, J. M.; ADAMS, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. PMLR, US, p. 1861–1869, 2015.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, United States, v. 9, p. 1735–80, 12 1997.

HOFFMANN, L.; ELSTER, C. arxiv. **Deep ensembles from a Bayesian perspective**, United States, 2021.

HONG, T.; PINSON, P.; WANG, Y.; WERON, R.; YANG, D.; ZAREIPOUR, H. Energy forecasting: A review and outlook. **IEEE Open Access Journal of Power and Energy**, United States, v. 7, 10 2020.

ISEH.A., J.; WOMA.T., Y. Weather forecasting models, methods and applications. **International journal of engineering research and technology**, United States, v. 2, 2013.

IZMAILOV, P.; PODOPRIKHIN, D.; GARIPOV, T.; VETROV, D.; WILSON, A. G. Averaging weights leads to wider optima and better generalization. **Conference on Uncertainty in Artificial Intelligence**, p. 876–885, 2018.

IZMAILOV, P.; VIKRAM, S.; HOFFMAN, M. D.; WILSON, A. G. G. What are bayesian neural network posteriors really like? **International conference on machine learning**, p. 4629–4640, 2021.

JAIN, G.; MALLICK, B. A review on weather forecasting techniques. **IJARCCE**, United States, v. 5, p. 177–180, 12 2016.

JENKINS, N. **Wind Energy Handbook 3e**. John Wiley Sons, Ltd, 2021. 637-663 p. ISBN 9781119451143. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119451143.ch9.

KULESHOV, V.; FENNER, N.; ERMON, S. Accurate uncertainties for deep learning using calibrated regression. In: PMLR. **International conference on machine learning**. [*S. l.*], 2018. p. 2796–2804.

LAKSHMINARAYANAN, B.; PRITZEL, A.; BLUNDELL, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: **Advances in neural information processing systems**. California, US: Neural Information Processing Systems (NIPS), 2017. p. 6402–6413.

LIU, M.; WANG, Y.; WANG, J.; WANG, J.; XIE, X. Speech enhancement method based on lstm neural network for speech recognition. **IEEE International Conference on Signal Processing (ICSP)**, p. 245–249, 2018.

LIU, Y.; QIN, H.; ZHANG, Z.; PEI, S.; JIANG, Z.; FENG, Z.; ZHOU, J. Probabilistic spatiotemporal wind speed forecasting based on a variational bayesian deep learning model. **Applied Energy**, United States, v. 260, p. 114259, 2020. ISSN 0306-2619. Disponível em: https://www.sciencedirect.com/science/article/pii/S0306261919319464.

MACKAY, D. J. C. A Practical Bayesian Framework for Backpropagation Networks. **Neural Computation**, United States, v. 4, n. 3, p. 448–472, 05 1992. ISSN 0899-7667. Disponível em: https://doi.org/10.1162/neco.1992.4.3.448.

MACKAY, D. J. C. **Bayesian methods for adaptive models**. United States: California Institute of Technology, 1992.

MACKAY, D. J. C. Probable networks and plausible predictions — a review of practical bayesian methods for supervised neural networks. **Network: Computation in Neural Systems**, Taylor Francis, United States, v. 6, n. 3, p. 469–505, 1995. Disponível em: https://doi.org/10.1088/0954-898X_6_3_011.

MADDOX, W. J.; IZMAILOV, P.; GARIPOV, T.; VETROV, D. P.; WILSON, A. G. A simple baseline for bayesian uncertainty in deep learning. In: **Advances in Neural Information Processing Systems**. Massachussets, US: MIT Press, 2019. p. 13153–13164.

MANWELL, J.; MCGOWAN, J.; ROGERS, A. **Wind energy explained: Theory, design and application**. 2nd ed.. ed. New Jersey, US: Wiley, 2010.

MOREIRA, N. D. A.; VASCONCELOS, R.; SILVA, Y. C. B.; MACIEL, T. F.; SIMOES, I.; MOTA, J. C. M.; HAMIDA, C.; PRADO, R. Z.; CAILLAULT, E. P.; KACOU, M. *et al.* Convolutional long-short-term memory networks (convlstm) for weather prediction using radar and satellite images. In: **XXIV Congresso Brasileiro de Automatica**. [*S. l.*]: UFC, 2022.

MOSHREFI-TORBATI, M.; LEDWICH, G.; TERZIJA, V. A review of wind speed and wind power forecasting techniques. **IEEE Transactions on Sustainable Energy**, IEEE, United States, v. 5, n. 1, p. 148–158, 2014.

MUKHOTI, J.; KIRSCH, A.; AMERSFOORT, J. van; TORR, P. H. S.; GAL, Y. **Deep Deterministic Uncertainty: A Simple Baseline**. United States: [*S. n.*], 2022.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Massachussets, US: MIT Press, 2012.

NEAL, R. M. **Bayesian Learning for Neural Networks**. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN 0387947248.

NEAL, R. M. Mcmc using hamiltonian dynamics. In: BROOKS, S.; GELMAN, A.; JONES, G.; MENG, X.-L. (Ed.). **Handbook of Markov Chain Monte Carlo**. Boca Raton, FL: CRC Press, 2011. p. 113–162. ISBN 9781439898211.

NIELSEN, M. A. **Neural networks and deep learning**. [*S. l.*]: Determination press San Francisco, CA, 2015. v. 25.

NIX, D.; WEIGEND, A. Estimating the mean and variance of the target probability distribution. **IEEE International Conference on Neural Networks (ICNN)**, United States, v. 1, p. 55–60 vol.1, 1994.

NOAA. **Global Forecast System, National Centers for Environmental Prediction, National Oceanic and Atmospheric Administration**. United States: NOAA, 2021. [Online]. Disponível em: https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast.

ONGOMA, V. **The science of weather forecasting: What it takes and why it's so hard to get right**. 2022. Disponível em: https://theconversation.com/the-science-of-weather-forecasting-what-it-takes-and-why-its-so-hard-to-get-right-175740.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In: **Advances in Neural Information Processing Systems 32**. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

PAVEZ, P.; MORALES, J. M.; HERNÁNDEZ, J. A. O.; RODRIGUEZ, L.; ARANEDA, E. Comparative analysis of wind resource characterization methods applied for a potential wind farm site in south chile. **Energies**, Multidisciplinary Digital Publishing Institute, United States, v. 14, n. 7, p. 2054, 2021.

PINSON, P.; KARINIOTAKIS, G.; NIELSEN, H. A.; NIELSEN, T. S.; MADSEN, H. Properties of quantile and interval forecasts of wind generation and their evaluation. **Proceedings of the European Wind Energy Conference & Exhibition**, United States, n. October 2015, p. 2–6, 2006.

PINSON, P.; MADSEN, H. Ensemble-based probabilistic forecasting at horns rev. **Wind Energy**, United States, v. 12, n. 2, p. 137–155, 2009. Disponível em: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.309.

QUASCHNING, V.; HANKE, T. Understanding renewable energy systems. **CRC Press**, United States, v. 2, 2019.

RODRIGO, J. S.; PAREDES, L. F.; GIRARD, R.; KARINIOTAKIS, G.; LAQUAINE, K.; STOFFELS, N.; BREMEN, L. V. 14 - the role of predictability in the investment phase of wind farms. In: KARINIOTAKIS, G. (Ed.). **Renewable Energy Forecasting**. Woodhead Publishing, 2017, (Woodhead Publishing Series in Energy). p. 341–357. ISBN 978-0-08-100504-0. Disponível em: https://www.sciencedirect.com/science/article/pii/B9780081005040000147.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, United States, p. 65–386, 1958.

SCIANNAMEO, V.; GOFFI, A.; MAFFEIS, G.; GIANFREDA, R.; PAGLIARI, D. J.; FILIPPINI, T.; MANCUSO, P.; GIORGI-ROSSI, P.; ZOVO, L. A. D.; CORBARI, A.; VINCETI, M.; BERCHIALLA, P. A deep learning approach for spatio-temporal forecasting of new cases and new hospital admissions of covid-19 spread in reggio emilia, northern italy. **Journal of Biomedical Informatics**, United States, v. 132, p. 104132, August 2022.

SHI, X.; CHEN, Z.; WANG, H.; YEUNG, D.-Y.; WONG, W.-K.; WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. **Advances in neural information processing systems**, MIT Press, v. 28, 2015.

SMITH, J.; JOHNSON, L. Bayesian inference with laplace approximations. **Journal of Statistical Computation and Simulation**, v. 50, n. 3, p. 327–345, 2020.

SONIYA; PAUL, S.; SINGH, L. A review on advances in deep learning. **IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)**, United States, p. 1–6, 2015.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, United States, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: http://jmlr.org/papers/v15/srivastava14a.html.

TASCIKARAOGLU, A. Evaluation of spatio-temporal forecasting methods in various smart city applications. **Renewable and Sustainable Energy Reviews**, United States, v. 82, p. 424–435, 2018. ISSN 1364-0321. Disponível em: https://www.sciencedirect.com/science/article/pii/S1364032117313308.

TRAN, K.; NEISWANGER, W.; YOON, J.; ZHANG, Q.; XING, E.; ULISSI, Z. W. Methods for comparing uncertainty quantifications for material property predictions. **Machine Learning: Science and Technology**, IOP Publishing, United States, v. 1, n. 2, p. 025006, 2020.

WANG, B.; LU, J.; YAN, Z.; LUO, H.; LI, T.; ZHENG, Y.; ZHANG, G. Deep uncertainty quantification: A machine learning approach for weather forecasting. In: **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**. New York, NY, USA: Association for Computing Machinery, 2019. (KDD '19), p. 2087–2095. ISBN 9781450362016. Disponível em: https://doi.org/10.1145/3292500.3330704.

WANG, M. Y.; PARK, T. A brief tour of bayesian sampling methods. In: TANG, N. (Ed.). **Bayesian Inference on Complicated Data**. Rijeka: IntechOpen, 2020. cap. 2. Disponível em: https://doi.org/10.5772/intechopen.91451.

WIKLE, C. K.; ZAMMIT-MANGION, A. Statistical deep learning for spatial and spatiotemporal data. **Annual Review of Statistics and Its Application**, Annual Reviews, United States, v. 10, p. 247–270, 2023.

WILLIAMS, C. K.; RASMUSSEN, C. E. **Gaussian processes for machine learning**. MA, US: MIT Press, 2006. v. 2.

WILSON, A.; IZMAILOV, P. Bayesian deep learning and a probabilistic perspective of generalization. **Advances in Neural Information Processing Systems**, MIT Press, United Kingdom, v. 2020-December, 2020. ISSN 1049-5258. Funding Information: This research is supported by an Amazon Research Award, Facebook Research, Amazon Machine Learning

Research Award, NSF I-DISRE 193471, NIH R01 DA048764-01A1, NSF IIS-1910266, and NSF 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science. Publisher Copyright: © 2020 Neural information processing systems foundation. All rights reserved.; 34th Conference on Neural Information Processing Systems, NeurIPS 2020 ; Conference date: 06-12-2020 Through 12-12-2020.

YANG, M.; FAN, S.; LEE, W.-J. Probabilistic short-term wind power forecast using componential sparse bayesian learning. **IEEE Industrial Commercial Power Systems Conference**, United States, p. 1–8, 2012.

YU, R.; LIU, Z.; LI, X.; LU, W.; MA, D.; YU, M.; WANG, J.; LI, B. Scene learning: Deep convolutional networks for wind power prediction by embedding turbines into grid space. **Applied Energy**, United States, v. 238, p. 249–257, 2019. ISSN 0306-2619. Disponível em: https://www.sciencedirect.com/science/article/pii/S030626191930011X.

ZHANG, Y.; WANG, J.; WANG, X. Review on probabilistic forecasting of wind power generation. **Renewable and Sustainable Energy Reviews**, United States, v. 32, p. 255–270, 2014. ISSN 1364-0321. Disponível em: https://www.sciencedirect.com/science/article/pii/S1364032114000446.

ZHU, Q.; CHEN, J.; ZHU, L.; DUAN, X.; LIU, Y. Wind speed prediction with spatio–temporal correlation: A deep learning approach. **Energies**, United States, v. 11, n. 4, 2018. ISSN 1996-1073. Disponível em: https://www.mdpi.com/1996-1073/11/4/705.