Author:
Sheehan, Helen M

Title:
Machine learning for wind flow modelling
using grid-based neural networks to capture wind flow changes over terrain

Machine Learning for Wind Flow Modelling

Using Grid-Based Neural Networks to Capture Wind Flow Changes Over Terrain

By

HELEN SHEEHAN

Cabot Institute
&
School of Civil, Aerospace and Mechanical Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of MASTER OF SCIENCE BY RESEARCH in the Faculty of Engineering.

APRIL 2022

Word count: 27114

Abstract

Modelling the wind flow over terrain is a key element of wind resource assessments within the wind energy industry. Existing flow modelling methods range from fast, low fidelity analytical models to time-consuming and computationally expensive high-fidelity Computational Fluid Dynamics software. Machine learning offers the potential for high-fidelity yet fast-running surrogate flow models. This project created surrogate wind flow models using machine learning which aim to achieve the accuracy of the industry-standard WAsP software for wind over terrain. WAsP is split into three components: orographic (elevation-induced) speedup; orographic turn; and roughness speedup. Hence, surrogate models were developed for each. While initial tests with Convolutional Neural Networks were unsuccessful, a *Grid Neural Network* approach was developed as part of this work, which takes in sub-grids of each input (terrain) map and output (speed or direction change) map, and uses data points from these sub-grids as the inputs and outputs of a Deep Neural Network. Models using this novel architecture proved to be trainable with a relatively small set of data, with the optimal sub-grid sizes discovered providing an understandable measure of the radius of influence of each input variable on the corresponding output. Using Grid Neural Networks, surrogate models were created to predict the orographic speedup and turn, and roughness speedup, at heights of 10m and 100m above ground level. The predictions for orographic speedup and turn for multiple sites and wind directions correlated well to the WAsP data at both heights. The surrogate model predictions for the roughness speedup at 10m above ground level were also a close match to the WAsP values, but at 100m above ground level the roughness speedup predictions were inaccurate for some sites. Future work could combine the surrogate models at separate heights, and incorporate the separate sub-model predictions into full wind resource maps.

Dedication and acknowledgements

I would like to sincerely thank everyone who has helped me with this project, including my supervisors Dr. Paul Harper and Professor Ervin Bossanyi, Dr. Lars Landberg and Dr. Daniel Poole for their support and guidance in all aspects of this research. Enormous thanks also to Dr. Liz Traiger of DNV, who has been an invaluable source of knowledge, technical insights and endless positivity! I am grateful also to James Bleeg and Renzo Ruisi of DNV, for their help with gathering data and learning to use the software needed for this work, and to Tom Bewley for his advice on machine learning techniques. And finally, I am thankful for the unconditional support of my family and my partner Pete, without whom I would never have been able to spend this year pursuing a research degree, particularly during such strange times.

DNV provided the site data and equipment to carry out this research, as well as the time of several of their researchers as mentioned above.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

Table of Contents

Page

List of Tables

Table **Page**

LIST OF FIGURES

ACRONYMS

ABL Atmospheric Boundary Layer. 2, 17

Adam Adaptive Momentum. 27, 56, 69, 86, 104

AEP Annual Energy Production. 2, 9

AGL Above Ground Level. xi–xiii, 16, 17, 21, 37, 38, 46, 47, 54, 55, 58–60, 62–65, 67–78, 80–103, 105–123, 125–128, 147–150, 156, 157

AI Artificial Intelligence. 5, 25

CFD Computational Fluid Dynamics. 4–6, 9, 10, 13, 15, 125, 128, 129

CNN Convolutional Neural Network. xi–xiii, 9–14, 29–31, 34, 47–51, 53, 55, 77–79, 99–104, 111, 112, 114, 116, 123, 125–127, 145, 147, 153–155, 157

CO_2 carbon dioxide. 1

DNN Deep Neural Network. xii, xiii, xv, 7–10, 29, 31, 32, 34, 51, 52, 54, 56, 59, 62, 63, 68, 79, 86, 104, 107, 112, 126, 128, 146, 148, 151, 152, 156, 158

DNS Direct Numerical Simulation. 5

FE Finite Element. 10, 13

GAN Generative Adversarial Network. 10, 11

GNN Graph Neural Network. 8, 10, 13

GNS Graph Network-based Solver. 10

Grid NN Grid Neural Network. xi–xiii, 31, 46, 51–59, 61–66, 68–96, 98, 104–123, 126–129, 146, 148, 150–152, 156–158

HAWT Horizontal Axis Wind Turbine. 2, 3

IBL Internal Boundary Layer. 19, 97

LES Large Eddy Simulation. 4, 5, 7, 10, 15

LSTM Long Short Term Memory. 7, 8, 29

MAE Mean Absolute Error. xii, 26, 35, 36, 39, 54, 57, 59, 60, 64, 66, 76, 80, 83, 88, 92, 105, 107, 119, 120, 126, 127, 153

met. meteorological. 15, 17, 20

MSE Mean Squared Error. xi, 26, 33, 35, 36, 56, 69, 86, 104, 113, 147

NN Neural Network. 10, 11, 13, 22, 25–28, 31–36, 41, 46, 47, 51, 53, 54, 70, 100, 101, 108, 125

POD Proper Orthogonal Decomposition. 12

RANS Reynolds-Averaged Navier-Stokes. 5, 8, 10

ReLU Rectified Linear Unit. 27, 46, 50, 54, 56, 68, 79, 86, 112, 126

RGB Red Green Blue. 30, 32, 53

SiLU Sigmoid Linear Unit. 48, 50

SOWFA Simulator fOr Wind Farm Applications. 5

Std. Dev. Standard Deviation. 39

w.r.t. with respect to. 25, 26, 45

WAsP Wind Atlas and Analysis Program. 4, 6, 14–22, 33, 36–38, 41, 43, 44, 47–50, 56, 58, 59, 62, 64, 65, 68, 70–79, 81–85, 87–89, 91–99, 101, 102, 104, 108, 111, 112, 114–123, 125–128

WRG Wind Resource Grid. 16, 21, 129

CHAPTER 1

Introduction and Literature Review

1.1 Introduction

1.1.1 Climate Change and the Need for Renewable Energy

It is well known that the production of carbon dioxide (CO_2) and other greenhouse gases by human civilisation is leading to an increase in global temperatures, aptly known as global warming, which will have devastating consequences on the climate (including weather patterns and the environment) if it continues [1]. To attempt to curb climate change, the United Nations have formed an international alliance known as the Paris Agreement, one of the key aims of which is: *"holding the increase in the global average temperature to well below 2°C above pre-industrial levels and pursuing efforts to limit the temperature increase to 1.5°C"* [2]. In a similar vein, the European Commission has created a Climate Target Plan, which aims to *"cut net greenhouse gas emissions by at least 55% by 2030"* and to *"make Europe a climate-neutral continent by 2050"* [3]. Achieving these ambitious targets will necessitate changes in many industries across Europe, including transport, manufacturing and energy production. One of the key steps that can be taken by countries to achieve these aims is to significantly diversify their energy mixes to decrease their reliance on fossil fuels.

The energy production industry accounted for ~25% of the UK's total CO_2 production in 2020, according to the currently available (provisional) statistics [4]. While the nuclear and renewable energy industries are not exempt from CO_2 production over their life cycle, they are considered to be "low carbon" or "carbon neutral" energy sources as they do not directly produce CO_2 when generating electricity. Wind energy is currently the second largest renewable energy technology in terms of GWh of electricity generated globally, accounting for ~19% (1.1 million GWh) of the world's electricity generation in 2018[1]. There is also a clear trend of increasing installed wind energy capacity, with the onshore and offshore wind energy installed generation capacity predicted to increase by 10 times and 40 times respectively by the end of 2050 [5]. It is evident, then, that wind power is here to stay, and that maximising the amount of power it produces is crucial in the fight to slow global warming.

1.1.2 Wind Energy

For onshore locations, to convert energy from the wind into electricity, wind turbines are installed at chosen sites, usually in arrays of tens or hundreds of individual turbines, e.g. the Whitelee wind farm in Scotland[2], which has 215 turbines with a total generation capacity of 539MW. There are a number of criteria for selecting the particular sites of land on which to locate wind turbines, such as site access, potential disruption to neighbouring settlements, and the suitability of the terrain for installing such large pieces of machinery. Paramount among these considerations is the amount of energy that can be harvested from the winds at that location, known as the wind resource. At present the wind industry has converged on the upwind three bladed Horizontal Axis Wind Turbine (HAWT) design (as in Figure 1.1), as typically seen on any British landscape; in this design the incoming wind creates a lift and drag force on the turbine blades, causing them to rotate in the vertical plane about the turbine hub. The rotating turbine blades are connected (either in direct drive or via a gearbox) to a generator, which generates electricity. Therefore, when deciding where to situate a wind farm or any individual turbines it is vitally important to know what the typical wind speeds and directions in that location are, in order to calculate the expected Annual Energy Production (AEP) from the proposed wind farm, optimise the layout, and determine whether it is financially viable.

The wind flow velocity is an almost chaotic variable to predict at a given instant, as it can change drastically in the space of minutes or even seconds (see e.g. Figure 1.2, which shows the onshore wind speed at a mast taken at 10 minute intervals); on top of this the wind has diurnal variations due to the changes in temperatures throughout the day and night, and daily changes in the weather. The seasonal changes in environmental conditions also affect the expected wind behaviour. In addition, the scale of wind turbines (a typical onshore wind turbine[3] may have a height of 100m at the hub, and blade lengths of around 70m) causes them to interact with the Atmospheric Boundary Layer (ABL). The ABL is a boundary layer that exists over the surface of the Earth, characterised by anisotropic, frictional, slow-moving flow (compared to the free stream flow above it). Fortunately, for the purposes of calculating the wind resource, a wind farm developer will require the steady state wind speeds over the area of terrain being

[1] `https://www.irena.org/Statistics/View-Data-by-Topic/Capacity-and-Generation/Technologies`, accessed on 30/08/2021

[2] `https://www.scottishpowerrenewables.com/pages/whitelee.aspx`, accessed on 10/09/2021

[3] `https://www.siemensgamesa.com/products-and-services/onshore` accessed on 12/09/2021

FIGURE 1.1. An example of a Horizontal Axis Wind Turbine. By © Hans Hillewaert, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=6361901

considered, usually calculated with a known set of atmospheric conditions (air temperature, pressure, density).

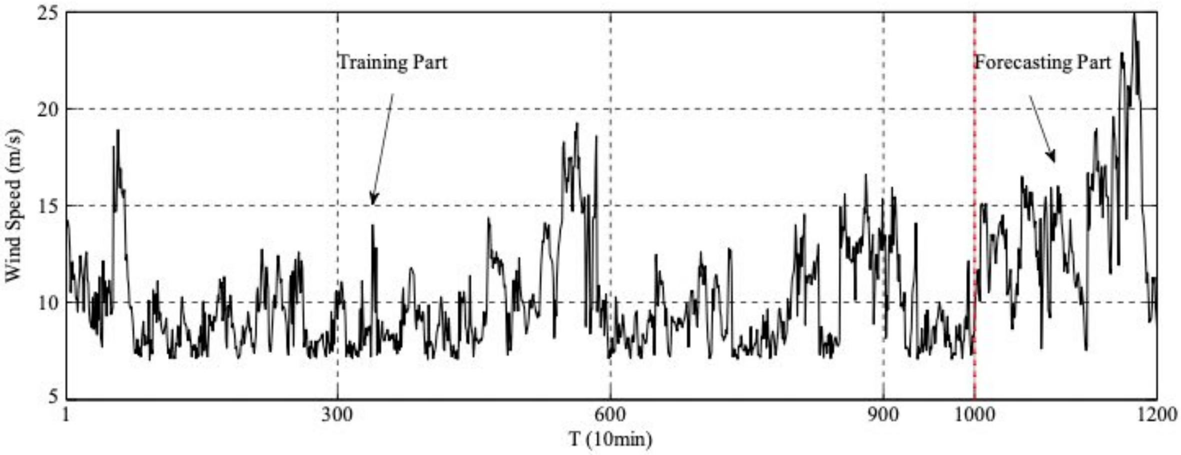


FIGURE 1.2. An example of a wind speed time series, reproduced from Figure 2 of [6].

1.1.3 Wind Flow Models

There are a number of commercially available software packages for calculating the wind resource or instantaneous flow, with a range of fidelity levels each with a corresponding computational cost.

Higher fidelity Computational Fluid Dynamics (CFD) models such as Large Eddy Simulation (LES) require significant computational power and time (on the order of days), while lower fidelity analytical models can take minutes to run. Ideally, all wind flow simulations would be performed with the highest fidelity models available, however this is not possible in reality due to time and computing resource constraints. This is of particular relevance when multiple wind directions, grid resolutions or atmospheric conditions are to be analysed, or for example when optimising the layout of a wind farm by testing multiple options.

1.1.3.1 Navier-Stokes Equations

The flow field over terrain or through wind farms can be modelled by the incompressible Navier-Stokes equations for mass conservation:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1.1}$$

and for momentum conservation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} \tag{1.2}$$

where $\boldsymbol{u}$ is the velocity vector of the flow; ρ is the flow density; ν is the kinematic viscosity; p is the pressure; and assuming that there are no external body forces on the fluid. The non-linear convective term ($(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$) in the Navier-Stokes equations is not easy to resolve, and so there are very few complete solutions to these equations. Turbulent flows contain eddies of varying length and time scales which, while incorporated into the Navier-Stokes equations, adds to the complexity of their solutions. In order to solve the Navier-Stokes equations with CFD, a common method is to discretise the derivatives in space and time, and apply initial conditions and boundary conditions to set up the appropriate scenario.

1.1.3.2 Analytical Models

Analytical models are the fastest, but lowest fidelity, class of flow model. Designed through consideration of the governing physical equations for a flow scenario (mass, momentum, energy balancing), analytical models contain a set of equations which estimate the general flow outline or speed. Examples include the Jensen wake model [7], which uses equations with a single solution to predict the time-averaged shape of the wake behind a wind turbine. A more complex analytical model is the Wind Atlas and Analysis Program (WAsP)[4], a software developed by DTU Wind Energy. WAsP calculates the wind climate of a specified geographical region by extrapolating measured wind data and incorporating the physical characteristics of the terrain. The WAsP calculations are based on the European Wind Atlas methodology [8], and are described more fully in Chapter 2. The WAsP software is much faster to use than the more complex CFD methods

[4] `https://www.wasp.dk/wasp`, accessed on 30/08/2021

outlined below, and is an industry standard method of calculating the wind resource over complex and large terrains and multiple wind directions.

1.1.3.3 Computational Fluid Dynamics

Computational Fluid Dynamics solvers exist for a wide range of fidelity scales, with the highest fidelity tools requiring the most computational time to solve. The Reynolds-Averaged Navier-Stokes (RANS) methodology is a form of CFD that treats flow properties (velocity and pressure) as a combination of average and fluctuating components, and is used to calculate steady-state flow regimes; it is one of the lower fidelity CFD solvers. RANS is commonly used in the wind energy industry due to its reasonable computational times, with examples such as [9] and [10]. However, RANS does not model individual turbulent eddies at any scale, so cannot be used to model precise turbulent motion.

Large Eddy Simulation explicitly models turbulent eddies in flow down to a specified grid scale, and hence it can calculate turbulent fluid motion. Eddies below the grid scale are accounted for via a stress term. LES solvers such as Simulator fOr Wind Farm Applications (SOWFA)[5] are used in scenarios where the turbulent flow is of specific interest, such as wind turbine wakes. Collecting real wind data at suitable quality and frequency for analysis or validation can be challenging, which leads to LES models being used in lieu of e.g. SCADA data; examples of LES in the wind energy industry are [11] and [12].

Finally, the highest fidelity form of CFD used for wind energy is Direct Numerical Simulation (DNS), which aims to solve the Navier-Stokes equations for a flow, without simplifications, between specified length scales. This is a useful method where knowing the exact patterns and motion of turbulent flow over time (such as a transitional region between laminar and turbulent flow) is important. For time-averaged applications of large-scale engineering tasks such as modelling the wind resource or the flow over a wind farm, DNS is impractical given its high time and computational requirements.

1.1.4 Motivation for this Project

Flow models are key tools for many aspects of the wind energy industry, and projects must often make a choice on how to compromise fidelity of the results with available time and computational resources. As such, any routes to reducing the cost of generating high fidelity wind flow predictions are of great interest to both the industrial and academic communities. One potential solution to this is the use of machine learning "surrogate models" trained to emulate existing physics-based methods. When appropriately designed and trained, such models can learn complex relationships between input and output variables, and once trained they typically run in a matter of seconds. The renewed interest in Artificial Intelligence (AI) in recent times has led to data-driven modelling

[5] `https://github.com/NREL/SOWFA`, accessed on 16/03/2022

being trialled for a wide variety of industrial applications. For the wind energy industry, relevant areas where such surrogate flow models would be of use include calculating the wind resource over terrain for proposed wind farms, or predicting the real-time wind flow through an operational wind farm (for the purposes of monitoring and control).

This work aims to develop a machine learning framework to calculate the changes in the wind speed and direction over a site that are induced by the elevation and roughness of the terrain. The WAsP model has been chosen as the target model for these surrogate data-driven models to emulate, rather than a CFD software; this is due to its understandable physics basis and simplified flow assumptions. The outcomes of this work can be seen as a first step towards creating fully data-driven CFD.

1.2 Literature Review

A review into the topics of machine learning for physics (in particular, fluid dynamics) applications, machine learning within the wind industry, and wind resource calculation methods, is now presented here.

1.2.1 Wind Resource Over Complex Terrain

The problem of calculating accurate wind fields over terrain has been investigated for many decades; even describing the flow over a low hill, which may sound simple, is a difficult task when considering the texture of the ground, slope of the terrain, thermal effects, viscous boundary layer effects, etc.. In 1975, Jackson & Hunt [13] produced a seminal piece of work in this field, presenting an analytical model for boundary layer flow over a low hill with uniform roughness. This model is the basis (with improvements and adjustments over the years) of the modern WAsP software, which is still industry-standard for flow over terrain. Despite real terrain being "complex", in that it consists of combinations of terrain types (roughnesses), steep slopes, and undulating and uneven ground, the complexity of the models needed for corresponding flow calculations have meant that most validation campaigns (i.e. measurements of real wind flow to evaluate the performance of computational models) were undertaken at sites with relatively "simple" terrain. Some of the most famous examples (which are still used today) include the Askervein [14], Bolund hill [15] and Perdigão [16] experimental campaigns; the features of these sites are, respectively, a 2-D uniform roughness ridge, an isolated hill, and a pair of parallel ridges (Figure 1.3). However, in recent times winds over more complex sites have been investigated and modelled, such as the mountainous Alaiz site [17], and islands in the northern Norwegian archipelago [18]. The review paper by Finnigan et al. [19] gives a comprehensive summary of advances in the field of wind resource calculation, outlining recent research in this field that focusses on the incorporation of detailed surface uses (in particular plant canopies) into flow models, as well as ongoing work into combining the different environmental effects such as

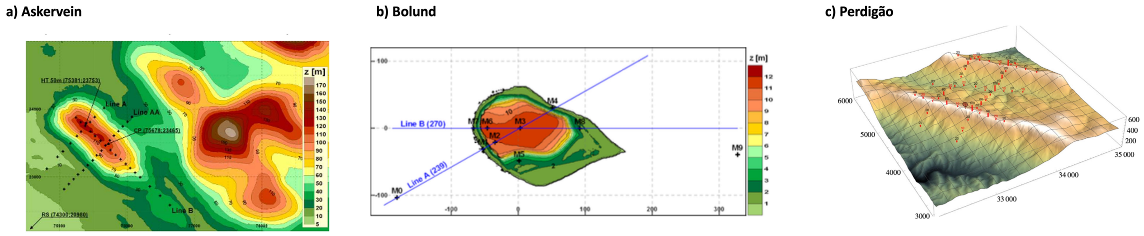


FIGURE 1.3. Terrain heights and meteorological mast layouts for three experimental campaigns to measure wind flow over terrain (a-c): Askervein (reproduced from [17]); Bolund (reproduced from [17]); Perdigão (reproduced from [16]).

temperature variation, gravity and boundary layer physics into universal flow models. The latest wind flow over terrain models look to integrate high fidelity LES methods within lower fidelity models (e.g. [20]) to achieve finer mesh resolution, particularly at the ground surface. Statistical methods [21] and even fully machine learning models [22] have been the subject of recent work on calculating the wind flow over complex terrains.

1.2.2 Machine Learning in Wind Energy

There have been a number of projects into the application of machine learning techniques to different areas of wind energy research. For example, many papers use machine learning techniques to either improve or replace wind turbine wake models ([23], [24], [25]). Within the maintenance sector, data-driven methods have been applied in order to link factors that contribute to wind turbine failures [26], for predictive maintenance through calculating turbine loads [27], and to detect damage such as cracks on turbine blades [28] (which takes inspiration from image processing research). Machine learning models are also being created for various aspects of the operation of wind farms, particularly to aid wind farm control via fast models of the flow through wind farms [29], and to predict the effects of changing operating conditions on a farm's power output [30].

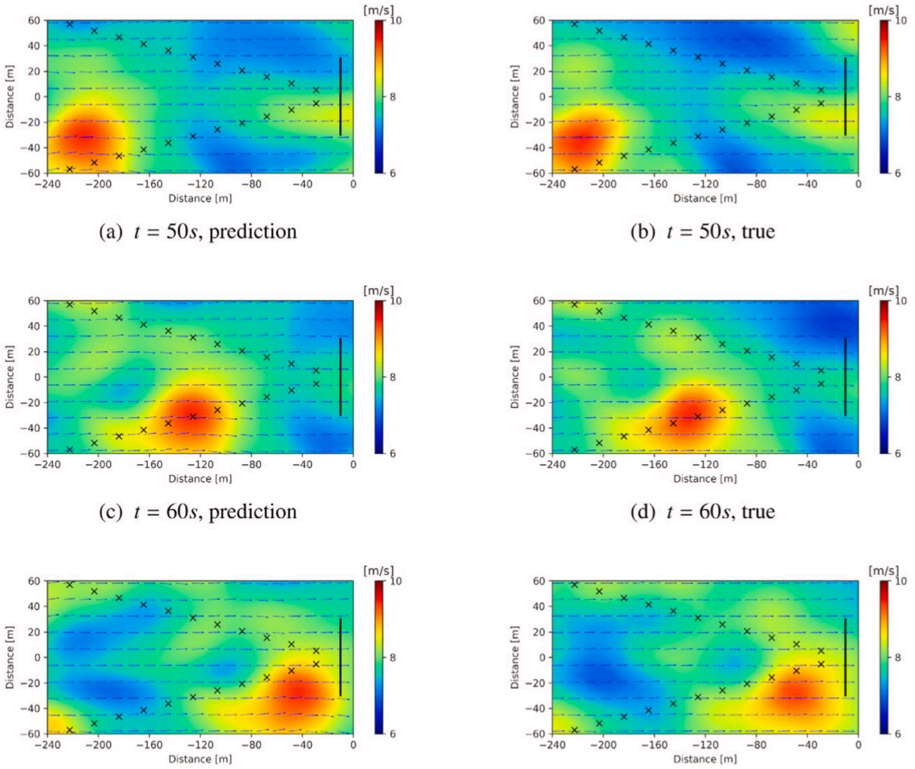
Another active area of interest in the wind energy community is the creation of data-driven models to predict time series of short-term wind speeds, which is of particular interest to both wind farm operators and electricity traders. The methods used to capture different time scales range from multiple Deep Neural Network (DNN) models [31] (one for each scale), to Long Short Term Memory (LSTM) nets [32] with "in-built" temporal memory capabilities; a review of data-driven methods for short-term wind speed time series prediction [33] suggested that the most promising techniques combine convolutional and LSTM networks. Building from this, there are a number of papers that combine both temporal and spatial dimensions, predicting the wind speed time series at locations corresponding to different wind farms. Liu et al. [34] used a network (the

"STNN-VB") formed of LSTM cells and convolutional layers in an encoder-forecaster architecture to produce predictions of the point wind speed values, with probability distributions of the model parameters calculated through variational Bayesian inference. Their STNN-VB model predictions of time series data are very promising, with a low level of uncertainty in the results; the maps of predicted wind speeds are close to the ground truth, but do diverge with increasing time steps. In a similar vein, Wilms et al. [35] combined spatial and temporal features of wind speed time series at neighbouring wind farms to predict the time series at a particular location, using a convolutional long short term recurrent neural network (convLSTM). Interestingly, the authors of this paper did not know the exact locations of the wind farms they received the measured wind speed data from, so instead experimented with different configurations of the relative farm locations on a grid; they found that constructing grids that most closely matched the real relative farm positions gave the best results.

A slightly different technique was investigated in Zhang & Zhao [36], using a DNN to predict grids of wind velocity and pressure from LIDAR measurements. Since DNNs do not have spatial or temporal feature extraction properties by design, the network designed here operated on point data, taking in the time and position of points and calculating the stream function, pressure function and therefore the wind velocity components and pressure for each point (see Figure 1.4). This method is dependent on the LIDAR data used to train the model, but it works well to predict the time-variant velocity and pressure fields given sparse LIDAR measurements. A key feature of this model was the incorporation of the Navier-Stokes equations for fluid dynamics into the loss functions of the network during training; this was inspired by Raissi et al. [37], discussed further on, and both increases the interpretability of the model and allows training of the DNN with relatively few input data points.

Bleeg [38] tackled the problem of modelling flow through a wind farm in a different manner, by constructing a Graph Neural Network (GNN) [39] to predict the power output at each wind turbine in a farm, setting the nodes of the GNN to represent the turbines themselves and learning the form of the connections between them given certain environmental variables including the free stream speed. This model gave accurate predictions of the wind speed and hence power output at each turbine when compared to RANS simulation results. Similarly, Park & Park [40] used a physics-induced GNN to predict the power output from wind turbines, by treating the turbines as nodes and the interactions between them as the edges of the GNN. There is potential for this model (and other data-driven surrogate models) to be used alongside physics-based models such as RANS when analysing numerous similar scenarios, such as different wind inflow directions to a wind farm, where the physics-based model could be used for a subset of the directions and a surrogate model for the remainder. This hybrid analysis could remove some of the uncertainty of "black box" machine learning models, by allowing for validation of each set of analyses with trusted simulation methods.

For the class of problem being investigated in this work, i.e. the use of machine learning



(a) $t = 50s$, prediction

(b) $t = 50s$, true

(c) $t = 60s$, prediction

(d) $t = 60s$, true

(e) $t = 70s$, prediction

(f) $t = 70s$, true

FIGURE 1.4. Examples of wind velocity fields predicted by the DNN of [36], at three successive times (a), (c) and (e). The corresponding true values are also shown for comparison (b, d, f) (reproduced from Figure 3 of [36]).

methods to predict time-invariant grids of wind speeds, recent research in this specific field points towards either a Convolutional Neural Network (CNN), or a combination of Deep Neural Networks applied to coordinate grids. Most research predicting the wind resource has focussed on the prediction of short-term time series, and few appear to have concentrated on the direct effect of terrain on the wind speeds. As wind farms become larger, the interactions of the flow between wind turbines within the farm will become more complex, which has led a new wave of interest into the application of wind farm control to both maximise AEP and minimise fatigue damage to turbines. Hence, there is a growing need for fast and accurate models to predict short-term wind flow variations through a wind farm, which could be a reason that less effort has been dedicated to finding surrogate models for wind flow over terrain.

1.2.3 Machine Learning for Fluid Dynamics

In general, machine learning has been trialled for two categories of problems within fluid dynamics: the creation of full surrogate models for CFD, and the amalgamation of machine learning methods with physical models to create hybrid tools. The papers described here focus more on the former of these since it is of the most relevance for the problem being investigated in this project. Examples of the latter application include the use of the Neural Network (NN) technique to calculate turbulence properties based on higher fidelity CFD, and incorporating this into lower fidelity CFD (e.g. from LES into RANS [23]); or using machine learning methods to learn additional turbulence terms for RANS as in [41] and [42]. As for creating machine learning surrogate flow models, it is evident from the literature available that this is a very active research field, and that almost every known machine learning technique from regression forests [43] to the Generative Adversarial Network (GAN) [44] has been applied to this end. The review paper by Brunton et al. [45] provides a comprehensive summary of the current state of the art in machine learning applications within various contexts of fluid dynamics.

Google's Deepmind research team have had a number of successes in applying Graph Neural Networks in a novel way to simulate a range of physical domains, including fluid flow, developing both the Graph Network-based Solver (GNS) in [46] and MeshGraphNets in [47]. GNS is analogous to molecular simulations, solving the motion of particles in the system by treating them as nodes in a GNN, and passing information on the dynamics along edges generated between nodes within a certain connectivity radius. By contrast, the MeshGraphNets model more closely mimics a mesh- or grid-based CFD (or Finite Element (FE)) solver, with GNN nodes at the grid points, and the edges connecting these points; example applications of MeshGraphNets are shown in Figure 1.5. The MeshGraphNets method creates a mesh (graph) over the object(s) to be modelled, which is adaptively re-meshed at each timestep to ensure that the connections between the GNN nodes represent the proximity of the nodes in terms of both the model architecture and physical distances. Similarly to GNS, information on the dynamics of the system are passed via the GNN edges at each timestep in MeshGraphNets, while the nodes hold information such as displacement and velocity at that position. Both GNS and MeshGraphNets produced very believable predictions of the motion of a range of materials and conditions with time, and they outperformed other machine learning surrogate solvers such as ConvNet [48] and Theurey et al.'s modified U-Net [49] architecture [50], which use CNNs and CNN-autoencoders respectively. Although a number of metrics such as rollout mean squared error were calculated to compare model performance, it is more informative to review the results of the different simulations by eye to distinguish their strengths and weaker areas. Both of Deepmind's models ([47], [46]) are compelling advances towards full machine learning flow solvers.

CNNs are a popular choice of NN for surrogate CFD models, as they are designed to extract spatial dependencies from data, and can be easily constructed to work on two- or three-dimensional "grid" data. Like the Deepmind projects described above ([47], [46]), Ummenhofer et al. [48] worked towards creating a machine learning based flow solver. The network proposed in [48] contained both convolutional and fully connected layers (i.e. a CNN-DNN hybrid), based on particle solver methods (similar to the GNS described above), designed to calculate the change in

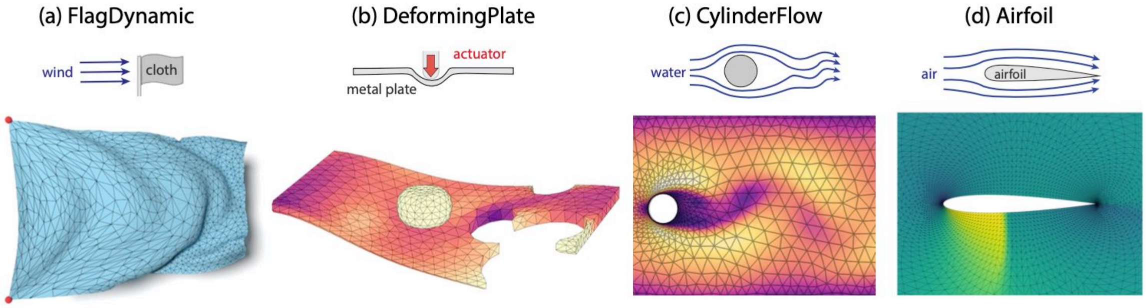


FIGURE 1.5. Examples of the applications of MeshGraphNets to predict the dynamics of physical scenarios, showing (a-d) a waving flag; deformation of a plate; water flowing around a cylinder; air flowing over an aerofoil. Reproduced from Figure 2 of [47].

positions and velocity of fluid particles over a single timestep. At each timestep, the positions and velocities of the particles were approximated from their current positions and external forces such as gravity, and the NN (termed ConvNet) calculated the "correction" that should be applied to each particle to account for interactions and collisions. Simulations performed with this ConvNet method closely match ground truth, even with large numbers of individual particles (6,000).

Lee & You [51] implemented a deep learning model for the prediction of unsteady flow over a cylinder, aiming to capture both the spatial flow features and the fluid flow evolution over time. The authors of this paper recognised connections between video processing and unsteady flow prediction, as video processing has both spatial (image processing on individual frames) and temporal (video is a series of images through time) characteristics. The aim of the deep learning model in this work was to predict flow fields for one or multiple future timesteps, given a time history of previous snapshots of the flow. To do this, a GAN was proposed, where both the generator and discriminator were CNNs, which was compared to the generator model on its own (i.e. a CNN). A key element of this work was the comparison of the performance of both the CNN and the GAN models with and without the addition of physical loss terms (representing mass and momentum conservation) to the loss functions. The ability of GAN models to learn from unsupervised data could be exploited to extract underlying flow physics from a large number of flow scenarios, as noted in this paper, and seen in the believable flow fields the GANs created at Reynolds numbers outside of those they were trained on. As an example, when predicting the flow over a cylinder after a single, large time step with known 3-D free stream speeds, the mean errors in the wind speeds in each of the three directions were 7%, 5% and 16% (from Table 3 of [51]).

Several papers use the classic example of flow over an aerofoil to train and test their NNs for flow prediction. Two such papers are [50] and [52], both of which aimed to predict the steady state flow over an aerofoil; these are perhaps the most relevant papers out of those described here for the problem at hand, as they predict time-invariant flows and are both based on structured grid input and output data. Bhatnagar et al. [52] created a CNN with an autoencoder (encoder-decoder) structure, consisting of progressively smaller convolutional filters and feature maps through the encoder section, followed by transposed convolution layers with increasing filter size in the decoder section. The purpose of this autoencoder structure was to extract important flow features in the encoder, and use those as a basis for building up the full map of fluid flow through the decoder - this is similar to the application of Proper Orthogonal Decomposition (POD). With this network Bhatnagar et al. were able to calculate the pressure and velocity fields around a range of aerofoil shapes, and with varying angles of attack, using a map of the shape of the aerofoil and the value of the angle of attack as the only inputs. This is a striking amount of relations between inputs and outputs for a CNN to learn, and it achieved a reasonable amount of success, with the predicted flow fields being very similar to the ground truth. Using their CNN to predict the 2-D flow over a previously unseen aerofoil shape gave errors of 6.41% and 8.59% in each of the directions over the entire flow (from Table 8 of [52]). Thuerey et al. [50] undertook a very similar investigation, again using convolutional layers in an autoencoder structure, but instead their network architecture was based on the U-Net model [49], and contained (feed-forward) connections between corresponding encoder and decoder layers. As with [52], this network was designed to accept grids of input data, in this case grids of the free-stream wind speeds in the x and y directions (with zero values inside the aerofoil), and a mask of the aerofoil shape. There is a lot of redundant information in these inputs, as the free stream speeds are effectively constants, and the aerofoil shape itself is repeated in all three grids. This method also produced maps of the flow speeds around the aerofoil with good accuracy (see example results in Figure 1.6), stating an error of 2.32% in their final model. From these works it is evident that CNNs can work well on grid-based flow prediction problems, which was a promising starting point for investigations into predicting terrain-induced flow changes.

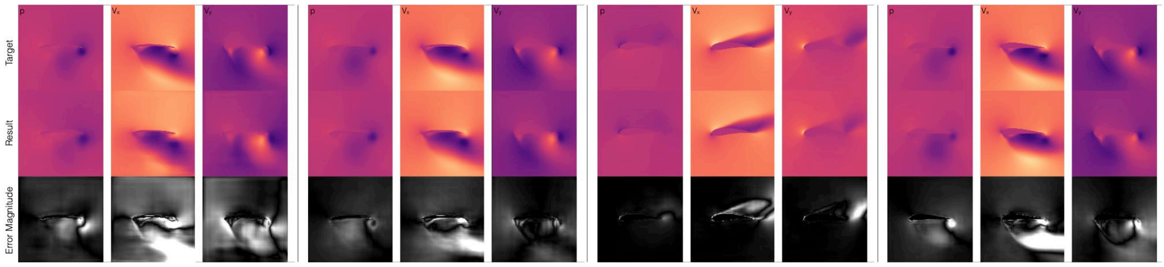


FIGURE 1.6. Examples of results from the U-Net style CNN of Thuerey et al. [50] for predicting flow around aerofoils in four cases (left-right). Reproduced from Figure 8 of [50].

The increasingly widespread applications of machine learning techniques to real-world scenarios (such as surrogate CFD solvers) has started to raise questions on how to ensure or even quantify the "trustworthiness" of such "black box" models, where the inner workings are unknown. This has piqued the interest of researchers in a range of fields, and a seminal piece of work relevant for those looking to create surrogate models to predict fluid dynamics (and any other physics-based scenarios) is that of Raissi et al. [37]. This paper demonstrated that inclusion of the physical equations into machine learning CFD or FE solvers could increase prediction accuracy, needed only a small number of training data points and crucially, was designed to obey the physical equations governing the system. Marrying the physical basis with the computational abilities of NNs is potentially an important route to ensuring the understanding and reliability of machine learning models in the future, and these considerations have been made in several of the papers described above, through loss functions ([51], [36]) or model update loop design [40], allowing for easier interpretation of the workings of the machine learning models. Gagne et al. [53] took an alternative approach to creating an interpretable CNN for the purpose of classifying storms by their likelihood of producing severe hailstorms. Once they had trained their CNN, the authors of this work undertook to determine how much recognisable climate physics their model had learned, through examining the filters (i.e. the inner workings) of the CNN, and experimenting with the types of storms which would "activate" different parts of their model. While this model may not have been designed with physical equations "built in", this is a very interesting approach to increasing the understandability of machine learning models, and connected "black box" elements to real weather modelling parameters.

There have been a number of research works on the creation of Neural Networks as surrogate flow models, several of these taking in "grid" data as input and producing velocity and pressure field "grids" as outputs. The most successful of these approaches tend to use Convolutional Neural Networks, which are known for their ability to recognise spatial dependencies within data, or Graph Neural Networks, which can be constructed to work on unstructured grids. In recent times, the incorporation of flow physics equations into the loss functions when training such networks has been seen to improve the accuracy of the predicted flows, as well as guiding the network towards correct solutions with fewer training data needed, and making the relations learned by the networks more understandable. From the literature, it would seem prudent to start with CNNs to create a data-driven flow solver, and to consider the form of its governing equations in case they can be incorporated into such a model.

Work from this project was presented at the Wind Energy Science Conference[6] in May 2021, titled "*Predicting Linearised Wind Resource Grids using Neural Networks*", alongside work by Dr. E. Traiger on a similar topic. A corresponding paper is currently in progress.

[6] https://www.wesc2021.org/fileadmin/wesc2021/themes/ParallelSessions20210528_1530.pdf, accessed on 27/09/2021

The aim of this project is to create a data-driven model which can achieve the accuracy of the WAsP software for flow over complex terrain. To do this, different machine learning techniques were applied, starting with CNNs as concluded above, and considering the physical basis of the target model (WAsP). After this chapter, the methodology of this project is outlined in Chapters 2 to 4, starting with the physics of the target flow model WAsP (Chapter 2), followed by descriptions of the supervised machine learning methods used in this project (Chapter 3) and of the ways that the accuracy and errors are measured and compared between the test models in Chapter 4. Once the methods are established this thesis presents the results of testing these machine learning techniques to create data-driven surrogate models for three sub-models of WAsP: orographic speedup (Chapter 5), orographic turn (Chapter 6) and roughness speedup (Chapter 7). Finally, conclusions and suggestions for further work are made in Chapter 8. Maps of the site terrains used as training data for the machine learning models are presented in Appendix A, and tables of the performance metrics for all the machine learning model tests discussed are given in Appendix B.

Chapter 2

WAsP

As a first step in the challenge of building a machine learning model to predict CFD results, this project investigated the creation of a surrogate model for WAsP. WAsP was developed from the European Wind Atlas [8], with the aim of calculating wind resource maps over large areas of terrain (such as the continent of Europe). The WAsP software is commonly used within the industry to carry out wind resource assessments, for example on prospective wind farm sites, as it is fast and well-validated, with known and documented limitations. This chapter describes the workings of the WAsP flow model, to establish the physical background of the target for the machine learning models tested in this work.

2.1 The WAsP Model

The WAsP model is a linear potential flow model, which gives it the advantage of fast calculation times compared to CFD methods such as LES, but conversely does mean that this model has some limitations in its accuracy, particularly over complex terrain. The WAsP flow model is made up of three constituent sub-models, which calculate effects on the flow due to different features of the terrain:

1. Shelter sub-model;

2. Roughness sub-model;

3. Orographic sub-model.

The effects of each of these features on the terrain are removed in the order listed above to calculate the generalised wind climate over the terrain assuming geostrophic wind conditions, using the measured wind data from meteorological (met.) masts on the site, and maps of the terrain. From this generalised wind climate, the terrain effects are applied in the reverse order to calculate the wind resource at five different heights Above Ground Level (AGL)[1]. From there a Wind Resource Grid (WRG) can be calculated at heights specified by the user (for example, turbine heights).

Figure 2.1 shows the terrain maps for the Waspdale example site and a single wind direction, and Figure 2.2 shows the orographic speedup and turn, and the roughness speedup calculated from the WAsP model for this site and direction.

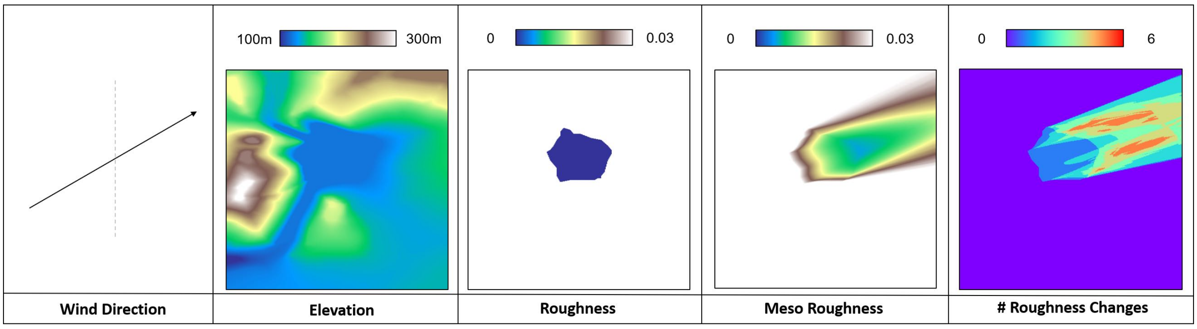


FIGURE 2.1. Terrain maps (as calculated by WAsP) for the Waspdale example site (left-right): elevation; roughness; meso-roughness; roughness changes.

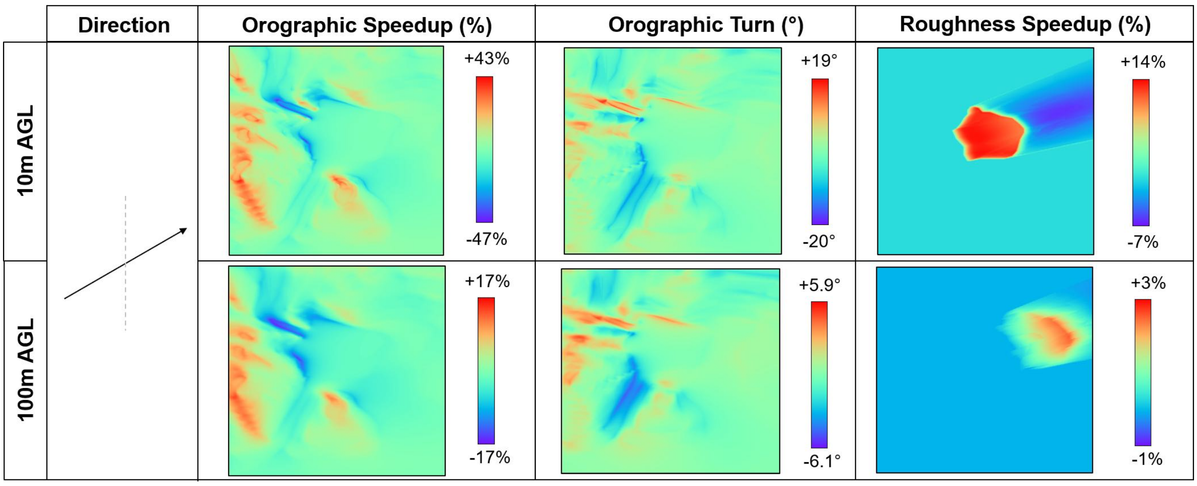


FIGURE 2.2. Resource maps (as calculated by WAsP) for the Waspdale example site at 10m AGL (top) and 100m AGL (bottom) (left-right): orographic speedup; orographic turn; roughness speedup.

[1] Note that this height is measured vertically upwards from the terrain elevation at each coordinate, so it follows the terrain elevation.

This work focussed on creating surrogate machine learning models for the orographic and roughness sub-models of WAsP. It does not include the shelter sub-model, hence the shelter effects are not described in any detail here. Full descriptions of the sub-models can be found in the WAsP manual [54] and the European Wind Atlas [55], and are outlined below for the orographic and roughness sub-models.

2.1.1 Geostrophic Wind

Within the ABL, large scale winds arise from the interactions of high and low pressure systems. For neutral stability, an approximate balance between the pressure forces and the effect of surface friction at the planet's surface can be found, giving the Geostrophic Drag Law equations:

$$G = \frac{u_*}{\kappa}\sqrt{\left(\ln\left(\frac{u_*}{f\zeta_0}\right) - A\right)^2 + B^2} \tag{2.1}$$

$$\sin\alpha = -\frac{Bu_*}{\kappa G} \tag{2.2}$$

where G is the geostrophic wind, u_* is the surface friction velocity, ζ_0 is the surface roughness length, κ is the von Kármán constant (0.40), α is the angle between the wind near the land surface and the geostrophic wind, f is the Coriolis value, $A = 1.8$ and $B = 4.5$. The WAsP model removes the effects of the orographic, roughness and shelter features of the terrain from the flow, then extrapolates this wind flow to the geostrophic height. At this height, the winds can be extrapolated horizontally, before the terrain effects are re-applied to give the wind resource at particular heights AGL and coordinates.

2.1.2 The Orographic Sub-Model

The WAsP orographic sub-model accounts for the effects of elevation changes across the terrain on the flow. It is a linear potential flow model, based originally on Jackson & Hunt's 1975 paper [13], in which they analysed the flow over low hills, assuming that the flow would remain attached at all points. WAsP has adapted this approach (the validity of which is detailed in Section 2.1.4) and applies a zooming, polar, cylindrical grid over the terrain, centred on the point of interest, e.g. a met. mast. The potential flow solution for a given orography is assumed to be a summation of Bessel functions, in the form of Equation 2.3:

$$\chi_j = K_{nj} J_n\left(c_j^n \frac{r}{R}\right) e^{in\phi} e^{-c_j^n \frac{z}{R}} \tag{2.3}$$

where K_{nj} are arbitrary coefficients, J_n the n$^{\text{th}}$ order Bessel function, r the radius, R the outer radius of the area considered, ϕ the azimuth, z the height above ground, and c_j^n are the j^{th} zero of J_n. An example of a 3-D Bessel function plot is given in Figure 2.3.

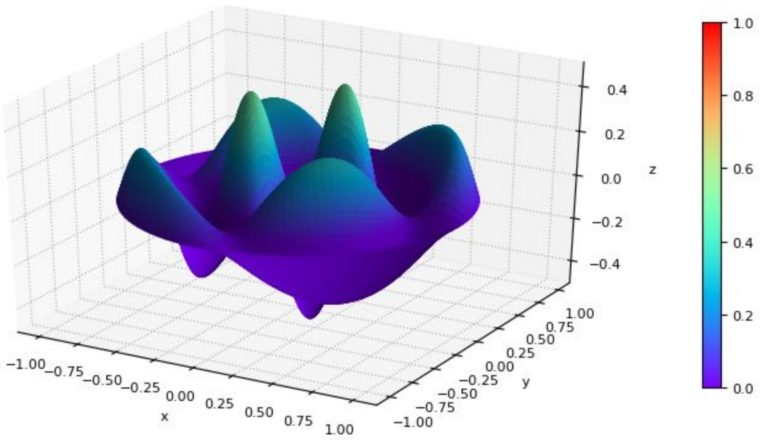


FIGURE 2.3. An example three-dimensional Bessel function.

To calculate the wind velocity field the flow potential derivative is calculated as:

$$\boldsymbol{u} = \nabla\chi \tag{2.4}$$

The coefficients of Equation 2.3 are found by considering the boundary conditions of the terrain, which is the kinematic condition at the ground (Equation 2.5):

$$w_0 = \frac{\partial}{\partial z}\chi\Big|_{z=0} = \boldsymbol{u}_0 \cdot \nabla h(r,\phi) \tag{2.5}$$

where w_0 is the terrain-induced vertical velocity, $\boldsymbol{u}_0$ is the velocity vector at the ground, and h is the terrain height. There is assumed to be zero flow velocity at the outer boundary of the terrain [55], i.e. $\nabla\chi = \mathbf{0}$ at $r = R$.

The effects of surface friction are approximated in the orographic sub-model by applying a factor to the wind velocities derived from the potential flow, resulting in a logarithmic wind velocity profile between the ground surface and a characteristic height L_j of the flow. The factor applied is dependent on the roughness of the terrain, the velocities at the top of the perturbation depth, the terrain surface and the height at which the flow is being calculated.

The orographic speedup calculated by WAsP is the change (in %) in the wind speed due to the elevation of the terrain as compared to the wind speed with no elevation changes. This could be e.g. the change in wind flow speed over a low hill.

The orographic turn calculated by WAsP is the change (in °) in the wind flow direction due to the elevation of the terrain as compared to the incoming wind direction. This could be e.g. wind turning to flow around a hill, or following a valley.

2.1.3 The Roughness Sub-Model

The WAsP roughness sub-model calculates the effects of changes in the terrain type on the flow; this model uses three types of terrain inputs, which are:

- **roughness**: roughness values are categories assigned to each coordinate of the terrain, depending on the height of the covering on the terrain;
- **meso-roughness**: meso-roughness values are the roughness values that would occur at each point of the terrain, if the wind over the terrain obeyed the geostrophic drag law;
- **roughness changes**: the number of transitions between different roughness values that occur over the terrain in a given wind direction.

A change in terrain roughness is assumed to result in the development of an Internal Boundary Layer (IBL), extending from the surface of the terrain to a characteristic height, which is dependent on the downwind distance from the roughness change and the roughness heights on either side of the change (Figure 2.4).

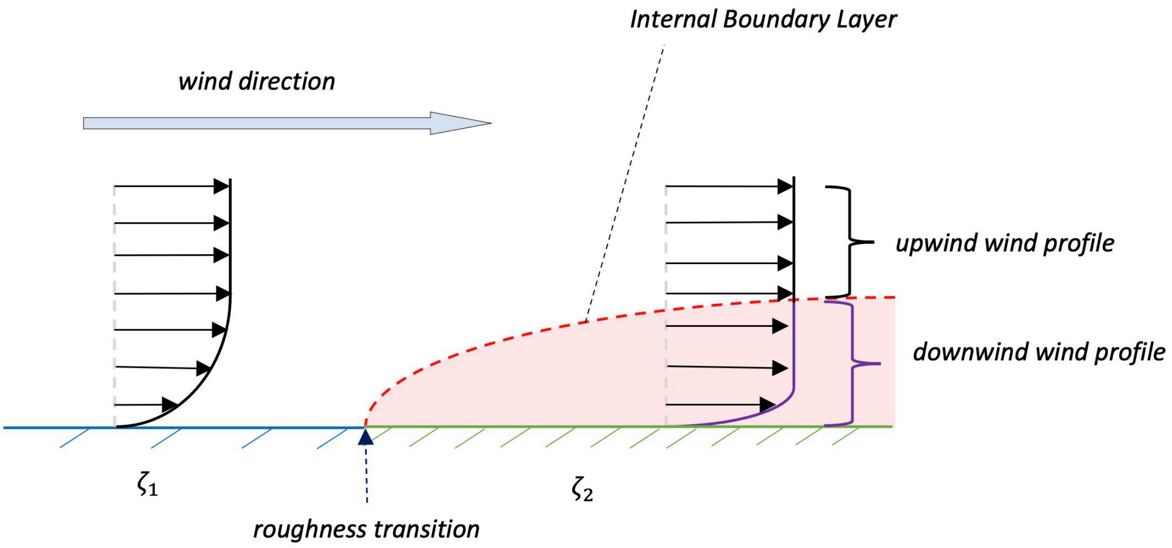


FIGURE 2.4. Schematic of the roughness submodel, depicting a transition between roughnesses ζ_1 and ζ_2 in the streamwise direction (where $\zeta_2 > \zeta_1$); the IBL is shown in red, and the change in the velocity profiles are shown in black and purple.

This IBL causes a change in the velocity profile of the flow which propagates vertically upwards with downstream distance [56], and at a downstream distance x from a roughness transition, the height h of the IBL follows the equation:

$$\frac{h}{\zeta_0'}\left(\ln\frac{h}{\zeta_0'}-1\right)=0.9\cdot\frac{x}{\zeta_0'} \tag{2.6}$$

where

$$\zeta_0' = \max(\zeta_{01}, \zeta_{02}) \tag{2.7}$$

and ζ_{01} and ζ_{02} are the roughness values on either side of the transition. The flow speeds across the transition can be related using the following equation:

$$\frac{u_{*2}}{u_{*1}} = \frac{\ln(h/\zeta_{01})}{\ln(h/\zeta_{02})} \tag{2.8}$$

where u_{*1} is the surface friction velocity upwind of the roughness transition, and u_{*2} the surface friction velocity at the point where the wind speed is to be calculated. The perturbed velocity profile can then be described as three sections of logarithmic velocity profiles:

$$u(z) = \begin{cases} u' \frac{\ln(z/\zeta_{01})}{\ln(c_1 h/\zeta_{01})} & \text{for } z \geq c_1 h \\ u'' + (u' - u'') \frac{\ln(z/c_2 h)}{\ln(c_1/c_2)} & \text{for } c_2 h \leq z \leq c_1 h \\ u'' \frac{\ln(z/\zeta_{02})}{\ln(c_2 h/\zeta_{02})} & \text{for } z \leq c_2 h \end{cases} \tag{2.9}$$

where

$$u' = \frac{u_{*1}}{\kappa} \ln\left(\frac{c_1 h}{\zeta_{01}}\right) \quad \text{and} \quad u'' = \frac{u_{*2}}{\kappa} \ln\left(\frac{c_2 h}{\zeta_{02}}\right) \tag{2.10}$$

with $c_1 = 0.3$ and $c_2 = 0.09$. The effects of multiple roughness transitions can be combined in sequence (in the streamwise direction), but WAsP limits the number of roughness changes that can occur per wind direction sector to 10. The roughness transitions are weighted such that their influence decreases exponentially with upstream distance from the current point being considered.

The roughness speedup calculated by WAsP is the change (in %) in wind speed at each point over the terrain compared to the wind speed at those points if the terrain had a single meso-roughness value.

2.1.4 Limitations

WAsP uses a linearised potential flow solver, which treats the air flow as incompressible and irrotational, and remains attached over surfaces. In order for the flow to remain attached over curved surfaces such as hills, it assumes that the length scale of the hill is much greater than its height (i.e. shallow slopes); this does not hold true over topography such as bluffs and cliffs, which is a known limitation of the WAsP model. WAsP calculations also assume that the atmospheric conditions are neutral (i.e. no temperature differential between the ground and the air), although it is able to handle small perturbations from the neutral state. WAsP permits users to make small corrections to the wind speed and direction measurements that it produces. These might be due to wind tunnel measurements of the shelter effects or known effects on the met. mast data due to shadowing from other obstacles in certain wind direction sectors [54].

The WAsP model's orography, roughness and shelter sub-models are distinct from each other, and each of these models provides factors to be applied to the input wind speed (speedup) and/or direction (turn). A linear combination of these factors is applied to achieve the actual wind resource. Hence, the speedup and turn values from the orography and roughness (and shelter, though it is not used in this work) sub-models can be used to train separate surrogate models. Additionally, in the case of the orography sub-model, given that the turn and speedup values are calculated in the same process (as detailed in Section 2.1.2 above), treating these as separate values and using them to train individual surrogate models does not invalidate the values.

2.2 Site Data

DNV provided WAsP workspaces from 15 real terrain sites, including roughness and elevation maps. Using these real sites and the data from the Waspdale computer-generated site provided with the WAsP installation, the orographic speedup and turn, and roughness speedup were calculated with WAsP 11 (as part of the WRG calculation) at 10m and 100m AGL (and additionally for Site 1, 25m AGL). In addition, the meso-roughness and roughness changes per site and direction were obtained as part of each WRG. The minimum size of the grids was 400x400 grid points, with a resolution of 50m in both the x and y axes giving a minimum grid size of 20km x 20km. All sites were assumed to have neutral, steady boundary layer flow conditions. Some sites were large enough to provide multiple grids of data, giving 19 sets of site data in total, the elevation and roughness maps of which are presented in Appendix A.

Each of the sites was split into 12 direction sectors, and the speedups, turn, meso-roughness and roughness change maps are calculated on a per-sector basis. Sector 1 encompasses data between ±15° clockwise of due North, where the wind is assumed to come from due North (Figure 2.5). In all the grids provided, the y axis was assumed to coincide with the North-South axis.

In all the models tested, nine out of twelve sectors per site were used for training, two for validation, and one was held back for testing on the final model. Different combinations of validation and blind test sectors were chosen for each terrain. As explained in the relevant sections further on, some models tested used a subset of sites for training.

2.2.1 Data Quality

All the site data (terrain data and wind resource values) used here were calculated with consistent environmental conditions as part of this work, using elevation and roughness maps provided by DNV; hence there should be no difference in the accuracy of the wind resource values (orographic speedup and turn, roughness speedup) between sites. The roughness maps which were input to machine learning models were the only form of input data which was not produced as part of the WAsP wind resource calculation; instead the "WAsP Grid Maker" tool (part of the WAsP 11 package) was used to produce roughness grids from contour (i.e. roughness transition) maps.

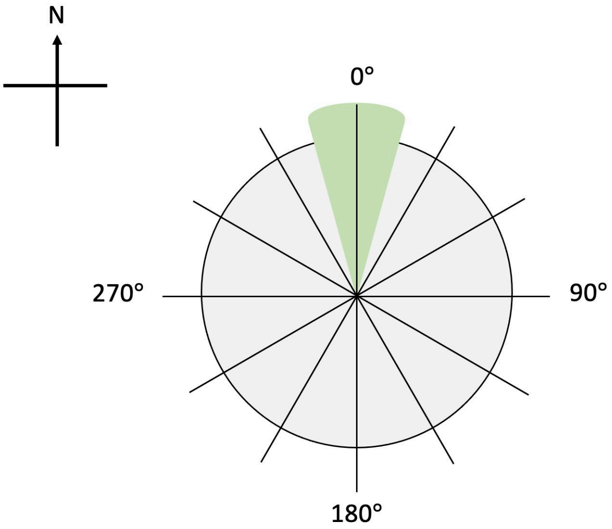


FIGURE 2.5. Wind directions for 12 sectors, with Sector 1 highlighted in green; note that the wind direction is the direction that the wind is coming from, i.e. for Sector 1 wind comes from 0°.

Calculating Cartesian grids of the roughness from contours sometimes produced small areas of high roughness in the direction of the x and y axes, which appear to be artefacts of the calculation rather than real features (see e.g. Figure A.3). These were not taken out of the roughness grids before using them as inputs to the machine learning models, which could give rise to some small errors. Apart from an area of unusually sharp slopes in Waspdale (Figure 2.1), the elevation of all sites appeared to be smooth. Within image processing, it is standard practice to augment the training data set through rotating, zooming or flipping the images; this was not possible here as any change in the conditions (wind direction, grid resolution or terrain) would require re-calculation of the data. WAsP 11 is (at the time of writing) the second newest version of WAsP available, and was chosen over the latest version (WAsP 12) as it did not alter the air density with the global location of the site.

While ideally each machine learning model tested would take in inputs from all the available sites, limits on the computing power available meant this was not viable. Sub-sets of sites used for training were chosen to include a range of different terrain features, such as flat areas, valleys, isolated hills and mountains (large elevation changes). As discussed later in the text, it is possible that flat, uniform roughness sites, with very small changes in the wind speed or direction, provide less "information" to Neural Networks during training. Conversely, it could be that using more mountainous terrain or terrain with many roughness changes would provide a greater variety of input-output relations for an NN to learn. However, as it is unlikely that wind farms would be installed on extremely mountainous terrain or within a forest, it was still important to produce data-driven flow models which performed well over plains and uniform roughness areas.

Having explained the theory of the physics-based model which the surrogate models will attempt to emulate, the various types of machine learning model elements, algorithms and techniques will be explained in the next chapter.

CHAPTER 3

MACHINE LEARNING METHODS

The term machine learning refers to a branch of AI that involves training machines (i.e. computers) to recognise patterns in data through the application of statistics. This work uses supervised machine learning techniques, wherein the models are trained to learn the relationship between input and known output data; in this case the input data are terrain variable maps, and the output data are the changes in wind speed and direction over this terrain. This chapter provides an overview of the range of machine learning techniques and Neural Network architectures that were used in this work to create and test surrogate flow models.

3.1 General Form of Supervised Machine Learning Models

The machine learning models created in this work followed a consistent format, and were constructed and trained via the following process:

1. Specify model architecture, initialising the neuron parameters (weights and biases) with a distribution of random numbers;

2. For a specified number of epochs:

    a) Pass a batch of input training data through the model;

    b) Calculate the loss (a form of error) over this batch by comparing predictions with target output data;

    c) Calculate the gradients of the loss with respect to (w.r.t.) the weights and biases of the model neurons;

    d) Update the weights and biases of the model using the calculated gradients, such that the losses are minimised.

The schematic in Figure 3.1 (a) shows the process of training a supervised NN in more detail. There are several important elements which are defined here (the capital letters refer to items highlighted in Figure 3.1 (a)):

(A) **Training, Validation and Test Data**: The set of pairs of input-output data is split into training, validation and (optionally) test data. Training data is used to update the NN parameters; validation data is used after training data each epoch to examine the model's ability to predict outputs from previously unseen inputs; test data can be used as blind test data on the final model, to evaluate its performance. Test data is not meant to influence the final model architecture, so it should not be used before then.

(B) **Batch**: Data can be split into batches such that the gradients are back-propagated and the NN parameters updated after each batch, rather than after all training data has been passed through the model. This reduces the computational load and allows the network to be updated after smaller, more representative groups of training data.

(C) **Epoch**: All training and validation data having passed through the network once is termed an epoch.

(D) **Loss**: The difference between the predicted and ground truth outputs is the loss, and can be calculated with different methods, such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). The loss is minimised through updating the network's weights and biases.

(E) **Back-Propagation**: After the overall loss of a network has been calculated, the gradients of the loss w.r.t. each of the network neuron's weights and biases (i.e. all the trainable parameters) must be calculated, in order for them to be updated to minimise the loss. This is achieved by using the chain rule to propagate the gradients backwards from the output layer, towards the first layer of the network. In a network with N layers, the gradient w.r.t. weights in layer i is calculated via the chain rule from layers N through i as:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial h_{N-1}}\cdots\frac{\partial h_{i+1}}{\partial h_i}\frac{\partial h_i}{\partial w_i} \tag{3.1}$$

where y is the model output, L is the model loss, h_i is the hidden layer output at layer i and w_i are the weights at layer i.

(F) **Optimiser**: Once the gradients of the loss w.r.t. the weights and biases have been back-propagated through the model, the optimiser calculates the actual changes needed in the values of the weights and biases.

(i) **Goal**: The overall aim of the machine learning model's training is to reach the minimum possible loss, by updating the model parameters to follow the gradients which will decrease the loss.

(ii) **Learning Rate**: Most optimisers use a learning rate, which is a factor applied to the gradients to determine how quickly the NN parameters change. A high learning rate has the risk of over-shooting optimum model parameter points, while a low learning rate may result in the network parameters not changing fast enough to reach an optimum network configuration. A **Learning Rate Scheduler** can be used during model training to alter the optimiser's learning rate, e.g. decreasing it if the model's validation loss does not improve for a given number of successive epochs, to fine-tune the NN's learning.

(iii) **Momentum**: Some optimisers include the effect of previous parameter updates, in order to remember the trend of the change in model parameters. This is done by applying momentum factors to the first and/or second order moments of the gradients. Using momentum in an optimiser helps the model to avoid getting stuck in localised gradient minima.

(iv) **Adam**: The Adaptive Momentum (Adam) optimiser [57] is used in this work, and uses both a learning rate and two types of momentum, applied to the first and second moments (mean and variance respectively) of the gradients. Adam updates NN parameters via:

$$\hat{m}_t = \frac{\beta_1 m_{t-1} + (1-\beta_1) g_t}{\left(1-\beta_1\right)} \tag{3.2}$$

$$\hat{v}_t = \frac{\beta_2 v_{t-1} + (1-\beta_2) g_t^2}{\left(1-\beta_2\right)} \tag{3.3}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} - \epsilon} \hat{m}_t \tag{3.4}$$

where η is the learning rate, β_1 and β_2 are the momentum values for the first (m_t) and second (v_t) moments of the gradients g_t at the current timestep, ϵ is a small constant value (1E-8), and θ_t and θ_{t+1} are the current and updated NN parameters.

(G) **Activation Function**: The output from a neuron can be passed through an activation function (Figure 3.1 (b)), e.g. to introduce non-linearity to the network or force the output from the neuron to be within a given range. Examples include the Rectified Linear Unit (ReLU) (for the former) and Tanh (for the latter) [58].

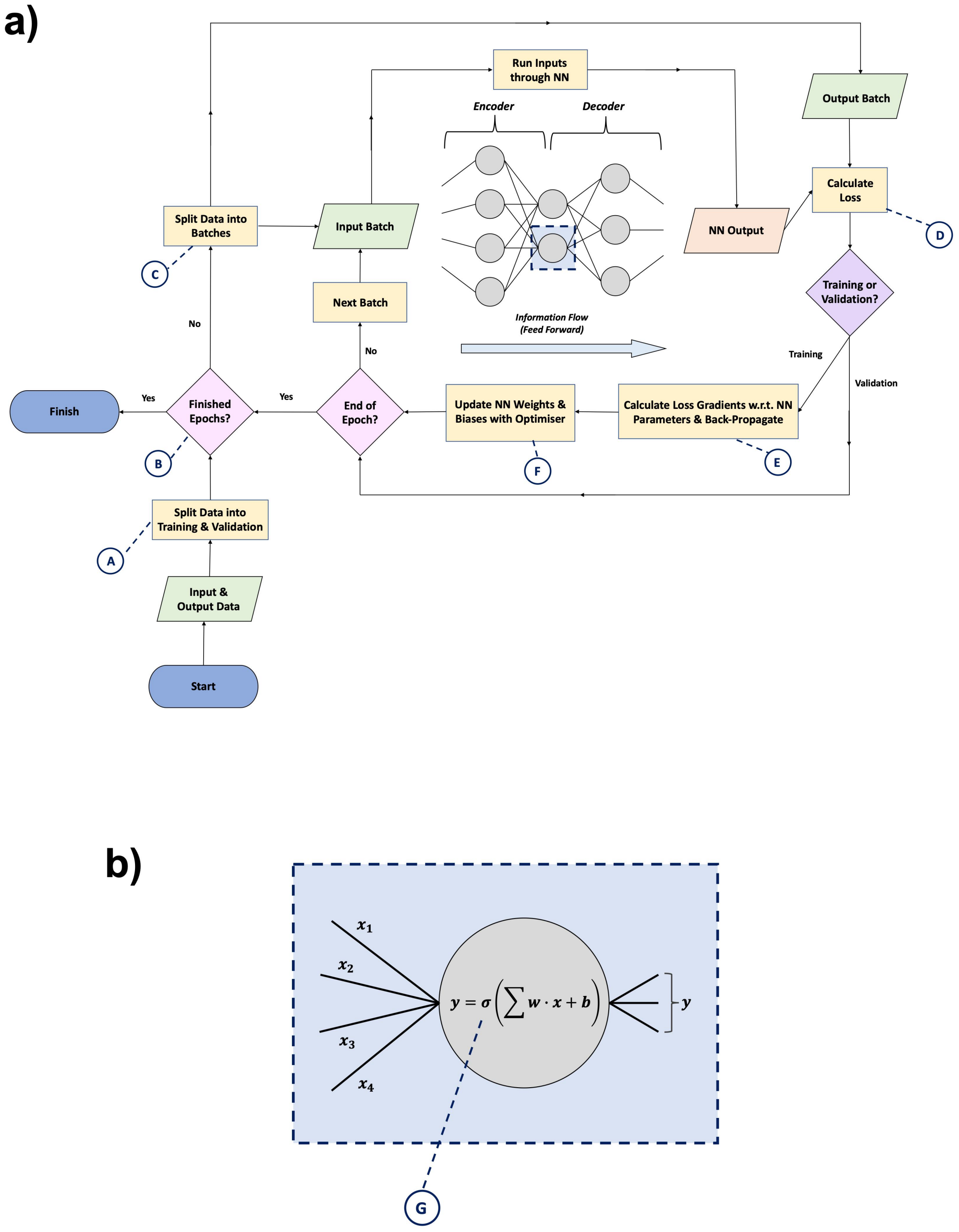


FIGURE 3.1. a) Flow chart of the training process of a Neural Network; b) detail of a single neuron, with inputs x_n, weight matrix w, bias b, activation function σ and output y.

3.2 Deep Neural Networks

A Deep Neural Network is formed of multiple layers of neurons, usually as a fully-connected, feed-forward network. Feed-forward networks have connections between successive layers only in the direction from the input towards the output; no reverse connections from later layers to earlier layers are present. In fully-connected networks, each neuron in a layer takes inputs from all neurons in the previous layer, and provides output to all neurons in the next layer (assuming a feed-forward network). Aside from fully-connected, feed-forward nets, DNNs can be used in architectures such as LSTM networks, or CNNs as described below.

3.3 Convolutional Neural Networks

A Convolutional Neural Network contains convolutional layers; Figure 3.2 shows an illustration of the workings of a convolutional layer in a CNN. A convolutional layer in a CNN contains a number of different filters, which are passed over the input data to extract relevant features from this data, producing "feature maps" which are the outputs of this layer. This can also be termed a cross-correlation operation, as the filters are representations of the features they recognise. The values of these filters are learned during model training so that the most relevant and useful features of the data are extracted. CNNs are a particular form of DNN architecture, with the neurons of each layer applied to input data so as to form filters in the required number of dimensions. Arranging the neurons in this way gives the network the capability to learn spatial dependencies in the data, but they are trained in the same way as DNNs.

Convolutional layers can be used on data with any number of dimensions, by using filters that have a corresponding number of dimensions. For a single convolutional layer, if the input data has c_i input channels and the number of output channels required is c_o, then the number of filters needed is:

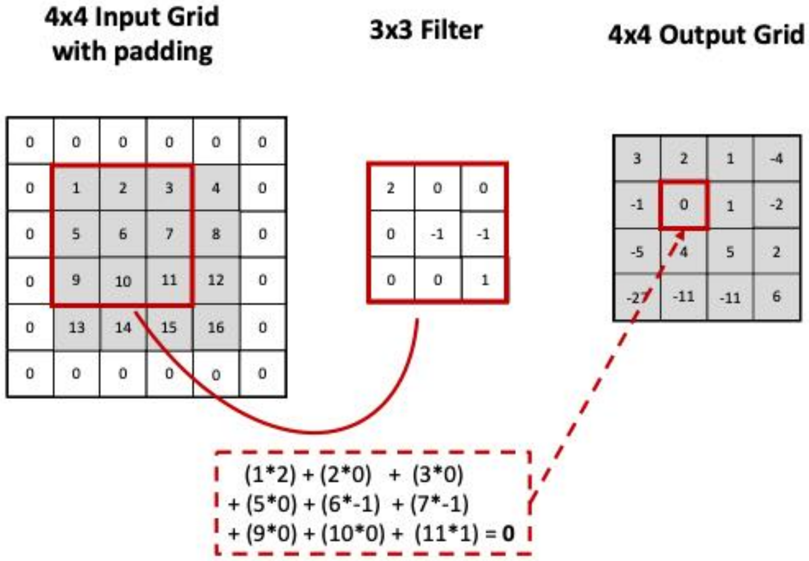
$$N_f = c_i * c_o \tag{3.5}$$

and assuming that the filters have side lengths of f_w and f_h, the number of weights to be learned in this layer is:

$$N_w = c_i * c_o * f_w * f_h \tag{3.6}$$

This can be a large number of filter weights to learn. A single feature map output from a convolutional layer is the sum of the convolution of each of the input maps with the filters corresponding to this output for each input.

The convolution operation on a two-dimensional grid results in an output grid smaller than the input grid; to keep the grid size constant through a CNN, the grids at each layer can be padded with zero values (as shown in Figure 3.2) or edge values.



4x4 Input Grid with padding

0	0	0	0	0	0
0	1	2	3	4	0
0	5	6	7	8	0
0	9	10	11	12	0
0	13	14	15	16	0
0	0	0	0	0	0

3x3 Filter

2	0	0
0	-1	-1
0	0	1

4x4 Output Grid

3	2	1	-4
-1	0	1	-2
-5	4	5	2
-27	-11	-11	6

(1*2) + (2*0) + (3*0)
+ (5*0) + (6*-1) + (7*-1)
+ (9*0) + (10*0) + (11*1) = **0**

FIGURE 3.2. Process of convolution for a layer in a CNN; an input grid of 4x4 is padded with zeros and convolved with a 3x3 filter; the resulting output grid retains the same size as the input grid. The outlined sections show the convolution inputs that result in a single output value.

Transposed convolutional layers are used to enlarge feature maps, e.g. in the decoder section of an autoencoder CNN. They are a version of convolutional layers which apply *transposed* convolutional filters to feature maps; the increase in feature map size is achieved through padding the input maps. CNNs often contain pooling layers after convolutional layers to reduce the model's translational variance [58]; these pooling layers cut down the size of the feature maps by passing pre-determined filters over the outputs from a convolutional layer that extract the maximum or average values from each patch seen by the filter. Pooling the output from layers means the model is less dependent on the relative spatial positions of important features, making it an important step for classification CNNs in particular.

3.3.1 Uses of CNNs

CNNs are most commonly used in image processing, in particular for classification and image transformation; they are the basis of well-known image recognition networks such as ImageNet [59] and VGG-16 [60]. While CNNs can be used to process one-dimensional data such as time series (by using one-dimensional filters), their strength is usually recognised as their ability to process multi-dimensional inputs with spatial dependencies, such as Red Green Blue (RGB) images or maps.

3.4 Autoencoders

Autoencoders consist of two stages: an encoder, and a decoder. Their aim is to break down input data into its most important components (i.e. extract the most relevant features) in the encoder stage, and then to regenerate the data (or a form of the data), using these components in the decoder stage. A straightforward example of this is as shown in Figure 3.1 (a), using a feed-forward NN with decreasing numbers of neurons in each successive layer as the encoder, with the decoder having layers of increasing numbers of neurons. Autoencoders are often used to de-noise data in their first stages by reducing the dimensionality of the inputs, and then to predict the de-noised version as output.

3.4.1 U-Net

The U-Net architecture [49] is a particular format of CNN, which was developed to automate segmentation of biological images, i.e. recognising different cells within a microscopic image of biological tissue. U-Net uses an autoencoder architecture, but with skip connections from encoder layers to their corresponding decoder layers. This passing of information between the encoder and decoder sections of the network in U-Net was designed to ensure that the network could remember the input features clearly, and it was very successful in performing biological segmentation compared to its competitors at the time. Variations on the U-Net architecture have been used in many applications of machine learning, including [61], [62] and [50].

3.5 Grid Neural Network

In this work, the concept of a Grid Neural Network (Grid NN) is introduced, and is described more fully in Chapter 5. In brief, the Grid NN applies kernels of a given size to two-dimensional input and output maps, and uses the resulting vectors of data as inputs and target outputs to a DNN. This network architecture is intended to give a clear relationship between the area of input information (the kernel sizes) used to produce the output predictions. The layers of a CNN can be difficult to interpret, even with methods such as occlusion (see [63]), hence the Grid NN uses kernels (analogous to convolutional filters) only over the raw input and target output data. Unlike CNNs, which must use whole images (for two-dimensional data) leading to spatial correlations, the two-dimensional data can be split into as many pairs of vectors of input and target output data as are possible, and then shuffled, for use in the Grid NN. The Grid NN also exploits more of the data than CNNs employed for the same task, as they are optimised after a certain number of input-output pairs, rather than after a certain number of images (or terrain maps) have been passed through the model. The Grid NN model is a novel architecture developed as part of this work, intended to incorporate the capabilities of CNNs for learning spatial dependencies, but with an easily understandable relationship between the kernel sizes applied to the input and output data.

3.6 Prevention of Overfitting

An NN can become *overfit* to the data used to train it, meaning that the network fails to learn generalised relations between the input and output data, and instead learns to produce specific training data outputs given particular inputs. This can occur if the training data set is not sufficiently large or varied; if the network is over-trained; or if the network is too large (e.g. too many neurons), as this means the network has the capacity to simply "memorise" the training data inputs and outputs rather than having to learn more general relationships. There are several techniques that can be used during training to prevent this from happening, with some of the most commonly used being batch normalisation, dropout, weight regularisation, and early stopping. The aim of most of these techniques is to keep network weights small, as overfitting will cause large weights on neurons corresponding to particular training data input-output pairs; this makes the network unstable as small changes in the inputs (e.g. noise) can then cause large, unexpected changes in the network outputs.

3.6.1 Batch Normalisation

Batch normalisation [64] is a widely used technique which ensures that the output from each layer in a network is reasonably small and centred around zero, which prevents subsets of model weights from becoming comparatively large and dominating the network predictions. Batch normalisation normalises all layer outputs within a batch such that the mean becomes 0.0, and the standard deviation becomes 1.0, as in Equation 3.7. The model then estimates, updates and keeps track of two factors (β and γ) per layer for scaling the normalised data back to its initial values if ever needed. ϵ is a small factor (1E-5) for denominator stability (in case of zero variance).

$$y = \frac{x - \mathbf{E}[x]}{\sqrt{\mathbf{Var}[x] + \epsilon}} * \gamma + \beta \tag{3.7}$$

If this layer is included in a Neural Network, it is typically placed between the layer of neurons (e.g. a linear or convolutional layer) and the activation function for that layer, so that the activation function does not distort the normalisation, and the following layer has a consistent set of input values. For one-dimensional data, such as a 4x1 vector of inputs passed through a DNN, each of the input channels would be normalised, i.e. every row in the vector would be normalised across each batch of data, and compute its own β and γ values. For two-dimensional data, such as images, passed through e.g. a two-dimensional convolutional layer, the normalisation is applied per channel over all images in each batch. If RGB images with three channels (red, green and blue) per image were batch normalised, each of the colour channels would be normalised separately, and have its own β and γ values.

3.6.2 Early Stopping

If a model trains for too long on a given set of training data, it will inevitably specialise towards learning the specifics of that particular data set, and will not generalise. To prevent this, the training of a machine learning network can be terminated early when a condition is met, e.g. if the validation loss has not improved over five successive epochs. Applied correctly, it can automatically stop the model at its optimum point of training and prevent it from overfitting.

3.6.3 Dropout

Dropout refers to the "dropping out" of neurons within an NN with a given probability [65]. For a given layer in an NN, dropout can be used to specify that the neurons in that layer must not be trained for certain batches of data, e.g. each neuron in a layer with a dropout probability of 10% will only be trained by 90% of the input data. This is to stop specific neurons from building up large weights and dominating the network.

3.6.4 Weight Regularisation

In order to keep NN weights small, the magnitude of the weights can be penalised during the training phase [66]; this is done by adding a term for the magnitude of the weights into the optimiser loss function, and minimising this combined term. In this work the sum of the squares of the weights (L2 regularisation) is used for certain sensitivity studies, and is included within the loss equation as in Equation 3.8 where it is added to MSE loss.

$$\text{loss} = \frac{1}{N}\left[\sum_{i=1}^{N}\left(o_{i,predict} - o_{i,real}\right)^2\right] + \lambda \sum_{j=1}^{P} w_j^2 \tag{3.8}$$

where $o_{i,predict}$ and $o_{i,real}$ are the predicted and real output values, w_j is the j$^{\text{th}}$ neuron's weight value, N is the number of samples in the batch of training data, P is the number of parameters in the network, and λ is the decay factor applied to control the effect of the network weights on the loss.

3.7 Coding

In this work, custom Python 3[1] code was written to read in the WAsP data, create and train the machine learning models, and evaluate and display the outputs from these models in comparison to the WAsP target values. The PyTorch[2] machine learning Python library was used in this code for the machine learning models themselves, as it has features such as templates for linear and convolutional layers, implementations of loss functions, optimisers, activation functions and other

[1] `https://docs.python.org/3/`, accessed on 16/03/2022

[2] `https://pytorch.org/`, accessed on 16/03/2022

relevant hyperparameters. The PyTorch framework is specifically designed to perform gradient calculation and back-propagation in machine learning models, making them easy to construct, train and validate. Other commonly used Python modules such as NumPy[3] and Matplotlib[4] were used in the custom code for data processing and plotting. As this code was generated on DNV systems it has not been made available here.

This chapter outlined the process of setting up and training a Neural Network, as well as describing the workings of Deep Neural Networks and Convolutional Neural Networks (which are the two architectures used in this project). The problem of overfitting has also been introduced, along with common techniques for preventing it. Once an NN has been designed and built, its performance must be evaluated, including how well it has learned to generalise the relations between input and output data, and whether any overfitting has occurred. The next chapter describes how performance metrics of the machine learning models tested in this work were calculated, and the difficulties of measuring the accuracy of the two-dimensional output maps.

[3] https://numpy.org/, accessed on 16/03/2022

[4] https://matplotlib.org/, access on 16/03/2022

Chapter 4

Evaluation Metrics

When evaluating the performance of machine learning models, it can be difficult to choose methods that accurately describe how well a model has achieved its goal. For classification networks such as image recognition models that assign labels to photographs, performance can be quite easily measured as the accuracy of the model's discrete predictions. However, in this work, the aim of the network is to predict a variable over a given two dimensional grid of points (e.g. a speedup map). The error of each grid point prediction can be measured and the overall average given as an indicator of the performance of the model, but this metric alone does not give sufficient information to make judgements on the accuracy of the model. Therefore, a number of different methods were used to assess model performance as discussed in this chapter.

4.1 Loss

The loss of a Neural Network is the difference between the output from the NN, and the target (real) output, in this case the WAsP speedup or turn values. Different methods can be used to calculate loss, with the most commonly used being Mean Absolute Error and Mean Squared Error, where:

$$\text{MAE} = \frac{1}{N}\left[\sum_{i=1}^{N} |o_{i,predict} - o_{i,real}|\right] \tag{4.1}$$

and

$$\text{MSE} = \frac{1}{N}\left[\sum_{i=1}^{N} (o_{i,predict} - o_{i,real})^2\right] \tag{4.2}$$

where N is the number of data points, $o_{i,predict}$ is a single predicted output value, and $o_{i,real}$ is the corresponding real output value. MSE penalises large errors more heavily than MAE, while MAE is preferred for fine-tuning models with small errors.

4.1.1 Loss Curves

When training a machine learning model, it is standard practice to consider the loss per epoch for both training and validation data. After the training part of each epoch, the validation data (which is different to the training data, but should be representative of the same learning aim) is input to the model, and the validation loss is calculated but not used to update the model. Plotting the loss from the training and validation per training epoch gives the user a look at how fast or slow the model is learning (if at all), and whether it is able to generalise and work well on the validation data. Figure 4.1 shows three different sets of loss curves:

(a) a good set of loss curves, where the training loss improves and saturates with more epochs, and the validation loss, while higher than the training loss (to be expected as the network is using new inputs), still decreases with more training;

(b) an overfit set of loss curves, where the model has not learned to generalise, but instead has learned how to produce specific outputs given the inputs from the training data - this can be seen from the poor performance of the validation data;

(c) an underfit set of loss curves, where the network does not have the the capacity to learn the relation of the inputs to the outputs, or the training data does not give a sufficient range of input and output pairs to learn from.

These loss curves can be used as a first impression of the model's expected performance, and helps to diagnose issues such as under- and overfitting.

4.2 Human Evaluation (By Eye)

In the early stages of model development, when testing many different NN architectures with greatly varying levels of success, it is possible to distinguish obviously failed or under-performing runs by eye. These include collapsed or unsuccessful runs compared to successful runs as seen in Figure 4.2. In such cases, it is perhaps easiest and fastest to compare the tests using human judgement (assuming these humans are experienced in this problem field) to decide which avenues are worth pursuing. Isola et al. [62] used a "mechanical Turk" approach to evaluating certain image transformation tasks by asking humans to distinguish between real and model-created street maps, and Gatys et al. [67] frequently discuss and evaluate their image transformation model's performance based on how the model outputs look to humans. However the work in this project, unlike [67] and [62], does require quantitative measures of accuracy for the WAsP surrogate models once promising machine learning methods are found.

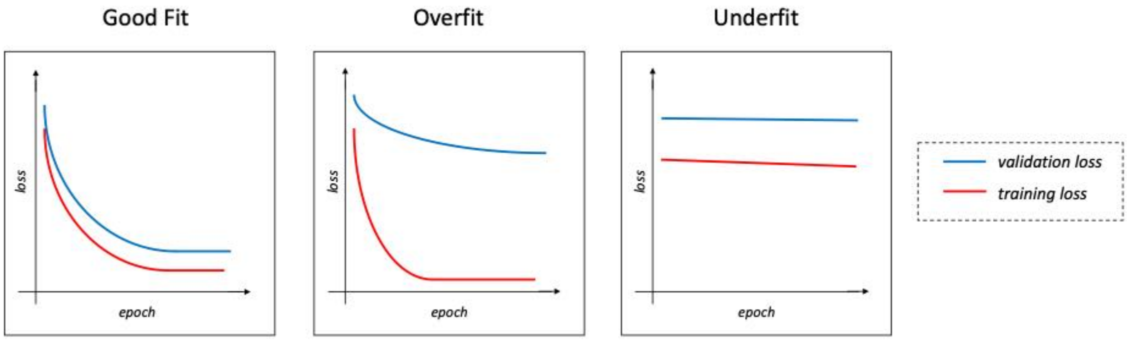


FIGURE 4.1. Examples of loss curves for models which show (left-right): good fit to data; overfit to data; underfit to data.

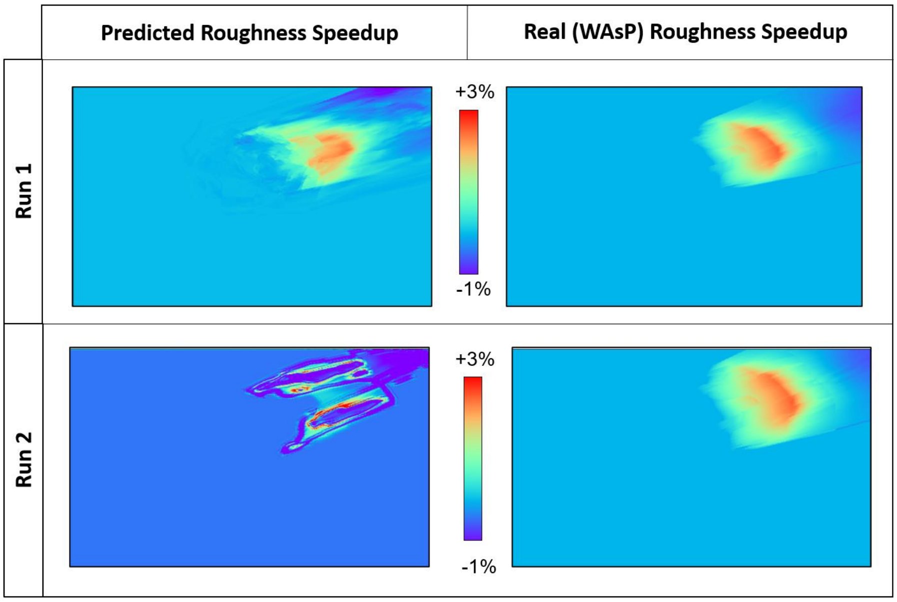


FIGURE 4.2. Plots of roughness speedups for Site 1 at 100m AGL from (left) model predictions and (right) WAsP: (top) a model with good performance; (bottom) a model with poor performance.

4.3 Error Metrics

In this section the *error* is defined as the difference between the ground truth (i.e. WAsP) values and the surrogate model predicted values, and the distribution of these errors has been presented in the following ways for each model tested.

4.3.1 Error Maps

The output for each validation site and sector from each test network is plotted as a two-dimensional map of the speedups or turn angles (as relevant) over the terrain. The difference between the model output and the ground truth values (WAsP) for this variable is also plotted, giving a map of the errors with colours representing the sign and magnitude of the error (i.e. strong red for large model over-prediction, white for a perfect match between WAsP and the machine learning model, and strong blue for large model under-prediction); an example is seen in Figure 4.3. This allows the user to see where errors are happening, which can help with diagnosing problem areas of the model; for example, in Figure 4.3 the shape of the roughness is clearly visible in the error map and in the orographic speedup plot.

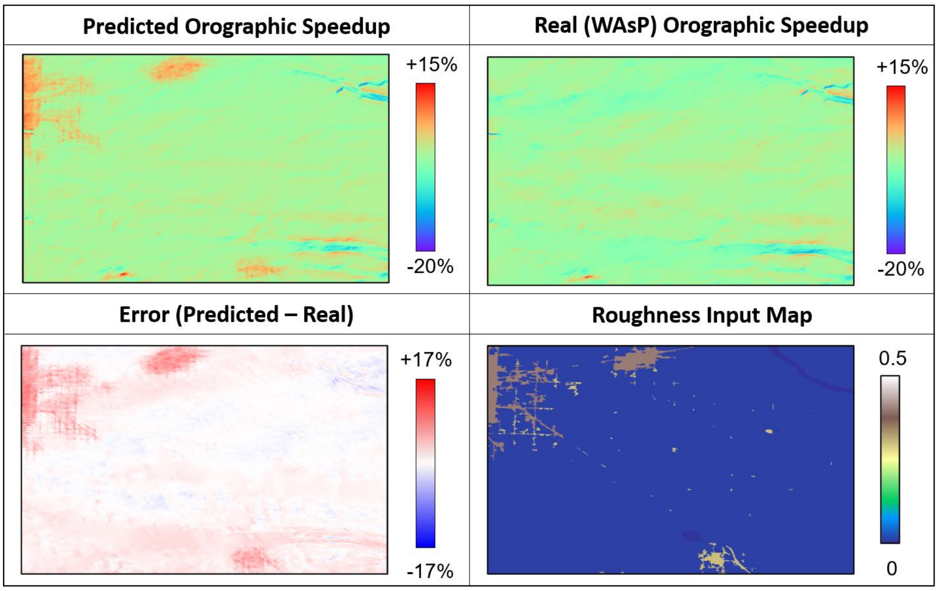


FIGURE 4.3. Plots of orographic speedup for Site 4 at 10m AGL from (left) model prediction and (right) WAsP, showing the input of roughness visible in the orographic speedup map.

The error values themselves were calculated for each point i.e. x,y coordinate as (where o is the output at a point):

$$\text{error} = o_{predict} - o_{real} \tag{4.3}$$

Speedup values were provided to the models in the form of speedup factors, such that the target speedup outputs were between 0.0 and 4.0. A speedup value of 1.1 corresponds to a 10% increase in wind speed at that point, and a speedup of 0.75 corresponds to a 25% decrease in wind speed, compared to flow over a flat terrain. To make the plots of the outputs and the errors more "human-readable", the speedups are plotted as % speedups rather than as factors - this was purely a post-processing step and did not affect the training or running of the models. Figure 4.4

shows an example of the error map plotted for the output from a machine learning model, where a general slight underprediction of the speedups can be seen.

4.3.2 Scatter Graphs and Bar Charts

Along with the maps of error values, for every site and sector in each run presented here the errors were also plotted as scatter graphs and bar charts, an example of which is shown in Figure 4.4. In the scatter graphs, the real and predicted values for each coordinate point are plotted, with colours corresponding to the error (as described in the previous section). The scatter plots also include a linear trend line for the data points, and the "perfect match" trend line for reference. These plots show the overall correlation between the real and predicted data points. This style of plot is also useful for determining limitations of the model, as seen in Figure 4.5 where the top model predicts a range of different speedup values for a single real value, compared to the good correlation shown in the bottom model. Note that the scales of these scatter plots are different as they are from two unrelated runs, and are intended only to illustrate a range of correlations between predictions and ground truth values.

The bar charts of errors (Figure 4.4) divide the errors into bins with equal ranges. Again, the colours of the bars correspond to the errors as per the scatter graph and error map for this site and sector. These bar charts are useful for viewing the modal error range, and for models and site sectors that work well, it confirms that most errors are very small. The bar charts of errors are plotted for each validation or blind testing site sector individually, and also for the aggregated data from all validation/inference site sectors.

4.3.3 Error Statistics

In addition to plotting the differences between predicted and target output values for every validation site sector per test run, the prediction error statistics are calculated. These are:

- mean error;
- MAE;
- Standard Deviation (Std. Dev.) of errors;
- minimum error;
- maximum error.

The mean and standard deviation are the parameters of a normal distribution that is plotted for each validation or inference site sector, and all sectors together, per test run along with the error bar charts. The MAE gives the overall error, and the standard deviation gives an indication of the spread of the errors, both of which are useful when comparing different models. The maximum and minimum errors are usually outlier values, but are interesting to note.

With the range of evaluation methods and error statistics defined above, the surrogate flow models tested in this work could be measured in a consistent manner. In the following chapters, the machine learning models created and run are detailed, and their results presented, showing the process of testing and updating the model architectures and configurations to achieve the best predictions of each of the WAsP output variables, starting with orographic speedup in the next chapter.

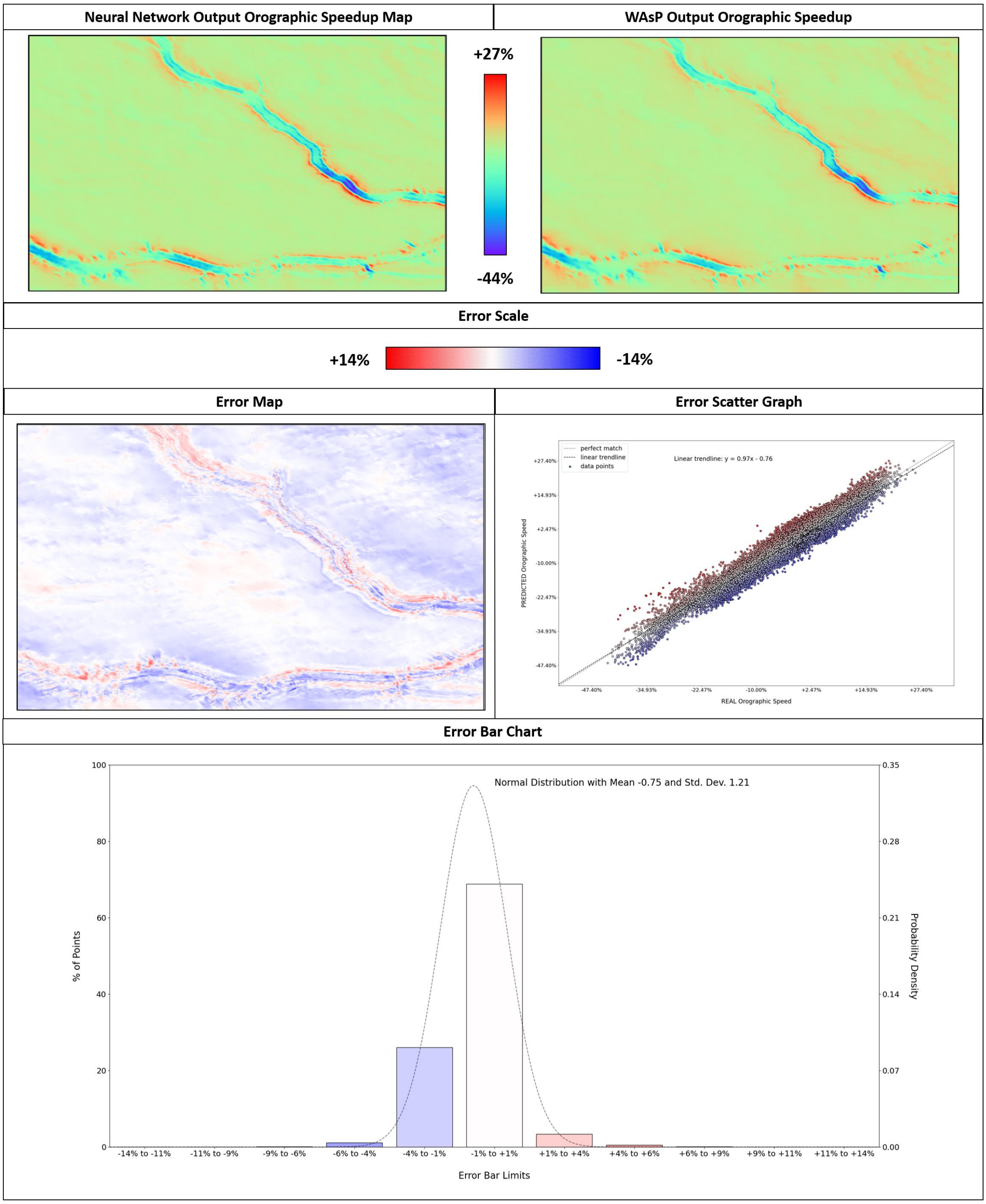


FIGURE 4.4. Top to bottom: orographic speedup map for a single site and wind direction produced by WAsP (left) and a Neural Network (right); error map between the predicted and WAsP maps (left) and scatter graph of the errors per coordinate (right); bar chart of errors.

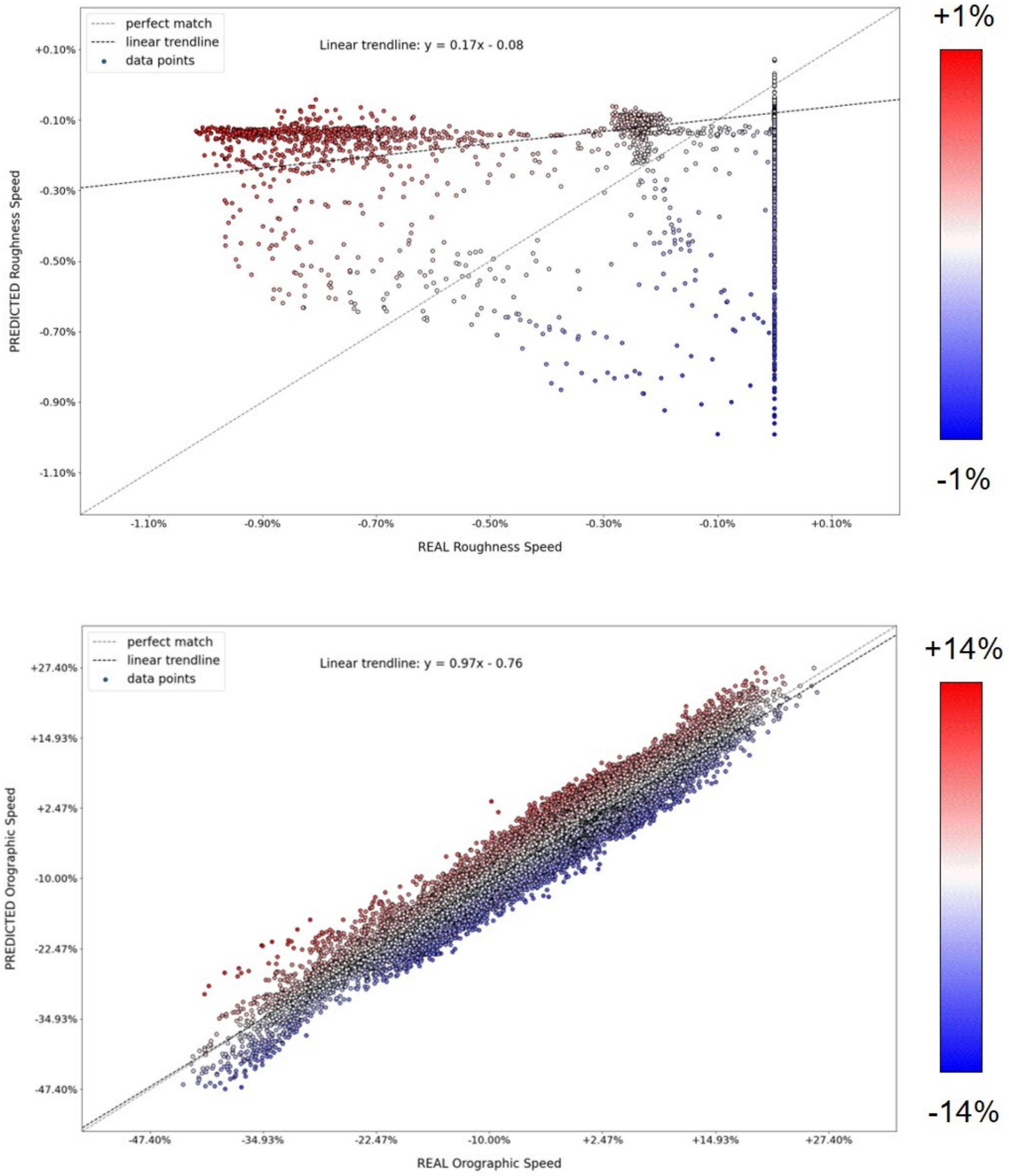


FIGURE 4.5. Error scatter plots for (top) a run with weak correlation between surrogate model predictions and true values; (bottom) a run where the true and predicted data points have strong correlation.

Chapter 5

Orographic Speedup Surrogate Models

The first sub-model of WAsP to be investigated was the orography sub-model, which calculates the effect of terrain elevation on wind velocity. The orography-induced changes to the wind speed and direction (turn) are reported separately in the WAsP software, and so are also treated as two separate variables in this work, with surrogate models created for each. This chapter investigates machine learning models to predict the orographic speedup, while Chapter 6 focusses on orographic turn.

Note that all figures in this chapter and Chapter 7 (Roughness Speedup Surrogate Models) that contain results from WAsP or machine learning models have been post-processed to present speedups as %, but the raw data is in the form of speedup factors (e.g. a speedup of 1.25 from WAsP is presented as +25% here). For all plots of speedups or turn in this chapter, Chapters 6 and 7:

- the plots are not to scale;
- all model output plots presented are validation data from the data set used in that run, unless otherwise specified;
- the colour scales of the plots of speedups are determined by the limits of the real (WAsP) speedups, hence model predicted speedups outside these limits take the extreme colours.

5.1 Inputs

Terrain elevation is obviously a key factor in the orographic speedup calculations, and Equation 2.5 of the WAsP orographic sub-model uses the elevation gradients to predict the speedup. As described in Section 2.1.2, the effects of terrain roughness are included in the orographic speedup model. Figure 5.1 shows the difference in orographic speedup when the roughness of the terrain is set to zero everywhere, and while the speedups are not significantly altered, there is a difference when roughness is not accounted for. For the sites available in this work the elevation varies between around -10m and 1,000m; using a first order central difference method, the range of elevation gradients is between approximately ±1. WAsP uses roughness category values, rather than roughness heights, which range between 0 and 1.5. Standard good practice in machine learning is to non-dimensionalise inputs and to ensure that the scales of all inputs are similar.

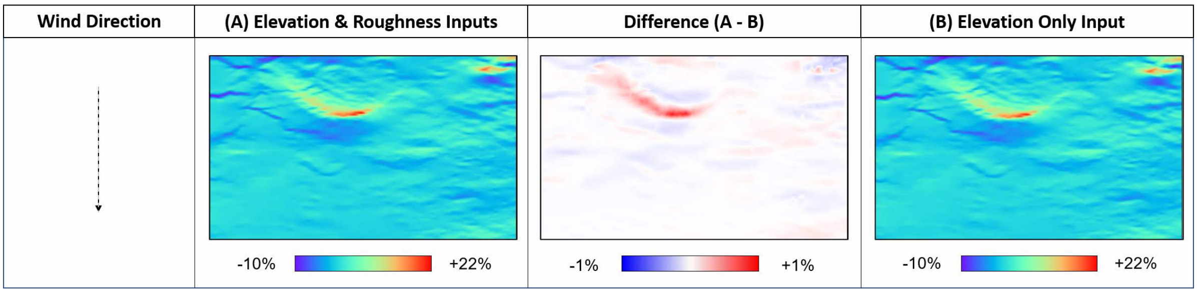


FIGURE 5.1. Effect on orographic speedups of excluding terrain roughness for Site 2, calculated by WAsP (left-right): orographic speedup with elevation and roughness inputs; difference in orographic speedup with and without roughness; orographic speedup with elevation only input. Values in % speedup.

5.1.1 Elevation Normalisation

To non-dimensionalise and re-scale the elevation data, two different normalisation techniques were tested, as described below. As not all sites were used for training, each machine learning model tested would normalise using its specific input sites, and saved the parameters for normalising any new sites in the inference stage.

5.1.1.1 Min-Max Normalisation

For min-max normalisation, elevation data (z) is normalised using the minimum and maximum values of the data set, by:

$$z_{mm} = \frac{z - z_{min}}{z_{max} - z_{min}} \tag{5.1}$$

where z_{min} and z_{max} are the minimum and maximum elevation values, and z_{mm} is the min-max normalised data. The normalised data is in the range of 0 to 1, and provides a simple way of bringing the scale of the elevation data in line with that of the roughness data.

5.1.1.2 Z-Scale Normalisation

Z-scale normalisation uses the mean and standard deviation of a set of data to re-scale the data, meaning that most of the normalised data lies between around ±2. Z-scale normalisation uses the following formula:

$$z_{zs} = \frac{z-\mu}{\sqrt{\sigma^2+\epsilon}} \quad \text{where} \quad \mu = \frac{\sum z_i}{N} \quad \text{and} \quad \sigma = \sqrt{\frac{\sum (z_i-\mu)^2}{N}} \tag{5.2}$$

where z or z_i are elevation data points, N is the total number of data points, and z_{zs} is the z-scale normalised data. The value of ϵ is set to 1E-5, and is used to prevent errors arising from any data sets with zero standard deviation.

A comparison of the real, min-max and z-score normalised elevations for Site 2, normalised with data from 11 sites total, is shown in Figure 5.2.

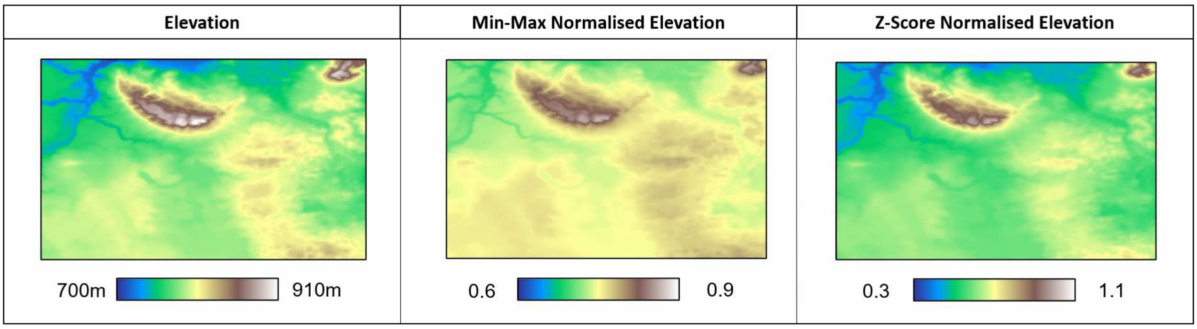


FIGURE 5.2. Comparison of elevation normalisation for Site 2.

5.1.2 Elevation Gradients

The first derivative of the elevation w.r.t. distance is already dimensionless. A first order central difference method was used in this work to calculate the first derivatives over the Cartesian grid of data points:

$$\dot{z}_x = \frac{z_{i+1,j} - z_{i-1,j}}{2\Delta x} \quad \text{and} \quad \dot{z}_y = \frac{z_{i,j+1} - z_{i,j-1}}{2\Delta y} \tag{5.3}$$

where z is the elevation, i and j are coordinates in the x and y directions, $\dot{z}$ is the spatial derivative of the elevation, and Δx and Δy are equal to one grid spacing in the x and y directions, respectively. At the edges of the elevation grid, first order forward or backward differences are used as appropriate to calculate the gradients. Each of these derivatives were resolved into directions parallel[1] and perpendicular to the wind. This was done using the following equations:

$$\dot{z_\alpha} = \dot{z_x} * \sin(\alpha) + \dot{z_y} * \cos(\alpha) \tag{5.4}$$

[1] This is actually anti-parallel to the wind, due to the notation of the wind being e.g. "from 0°" i.e. at 180°. This is consistent through all the tests done, and is the equivalent of multiplying the derivative parallel to the wind by -1.

with α being the angle clockwise from the y-axis (North-South). An example of the components of the elevation gradients resolved parallel and perpendicular to the wind direction (which is shown by the arrow) are in Figure 5.3.

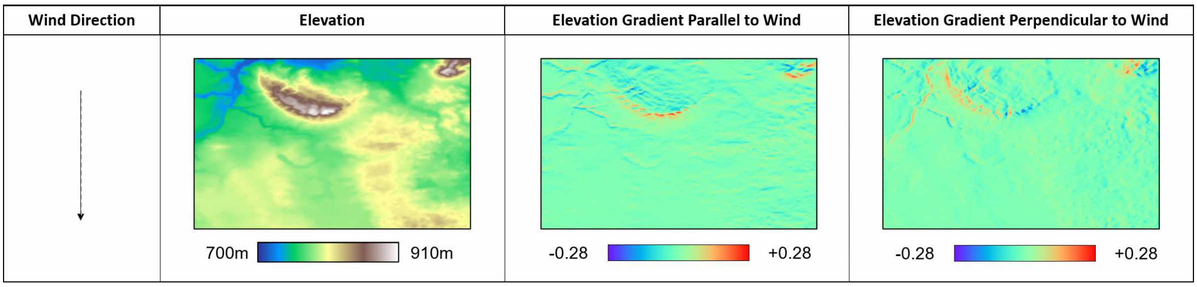

Wind Direction
Elevation
Elevation Gradient Parallel to Wind
Elevation Gradient Perpendicular to Wind
700m
910m
-0.28
+0.28
-0.28
+0.28

FIGURE 5.3. Site 2 (left-right): elevation; component of the elevation gradients resolved parallel to the wind direction; component of the elevation gradients resolved perpendicular to the wind direction.

5.1.3 Direction

The single point NN and Grid NN models include the incoming wind direction as an input, unless otherwise specified. Since the direction is given as a number between 0°and 360°, when it is used an an input it is normalised to ensure that this input is also between 0 and 1, using:

$$d_{norm} = \frac{d}{360.0} \tag{5.5}$$

where d and d_{norm} are the raw and normalised incoming wind directions. In this case there is a step change at 360° as it is normalised to 0 rather than 1, and a direction of e.g. 330° would be normalised to 0.92.

5.2 Single Point Neural Network

The first Neural Network tested had a deliberately simple design, taking in data from a single point and the wind direction as inputs, and producing orographic speedup point values as output; the network itself consisted of two hidden layers with five fully-connected neurons in a feed-forward configuration, each using ReLU activation. Each test used input data from one site and one type of variable to predict the orographic speedup at 100m AGL. Figure 5.4 shows the results for Site 1 for tests using min-max normalised elevation (top) and elevation gradient (bottom) as inputs, and Table B.2 gives the performance metrics.

The error maps for both of these tests are nearly identical to the orographic speedup target map, indicating that the NN predicted speedups have almost no correlation to the target speedups. Although it is difficult to see due to the scales being very small, the NN-produced orographic

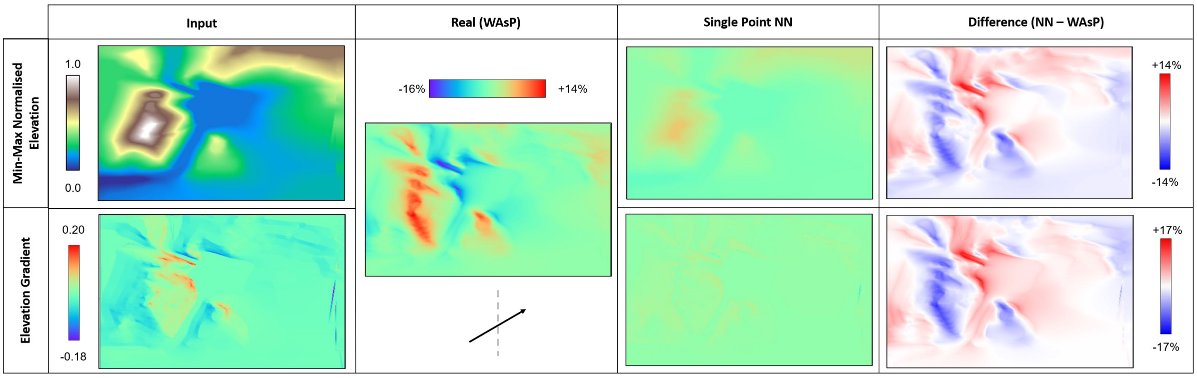


FIGURE 5.4. Single point NN results for Site 1, using (top) min-max normalised elevation and (bottom) elevation gradient as inputs to predict the orographic speedup at 100m AGL. Shown are (left-right): input maps; WAsP-calculated orographic speedup; single point NN predicted orographic speedup; difference between the NN predicted and WAsP calculated speedups.

speedup maps bear a strong resemblance to the input maps; for example the hill can be seen in the high speedup area predicted by the NN using elevation as an input. This simplistic NN does not contain enough spatial information to produce accurate predictions of the orographic speedup, and the inclusion of the direction as an input did not seem to change the results significantly for different sectors. Moving on from this, the next network architectures needed to incorporate spatial information from the terrain maps.

5.3 Convolutional Neural Networks

A number of successful recent studies have used CNN-based models to predict flow field maps given grid-based inputs, as outlined in the earlier literature review (Section 1.2). Bhatnagar et al. [52] and Thuerey et al. [50] employed two different CNN architectures to predict the static pressure and two-dimensional velocity fields around aerofoils, given certain information such as the aerofoil shape and flow conditions. Parallels can be drawn between these papers and this project, which also has grid-based inputs (terrain maps) and outputs (speedup or turn maps), so CNNs would seem a logical architecture for a WAsP surrogate model. The code written for this part of the research was able to create CNNs with multiple convolutional and transposed convolutional layers, different numbers of channels, activation functions, pooling and upsampling. This gave a large range of potential network configurations to test. Note that the input and output grids were not rotated and the wind direction was not provided as an input to these CNNs, and hence the directional information was assumed to come from the direction-resolved elevation gradients. All CNNs were tested using orographic speedups at 100m AGL.

The work of Bhatnagar et al. [52] on flow prediction over a grid used an autoencoder-based

CNN structure, with small filters in each convolutional layer (the input data grids were 150x150, the filters were either 5x5 or 3x3). The size of successive filters decreased in the encoder section, and then increased in the decoder section. The network in this paper was used as a starting point for creating and testing multiple CNN architectures for this problem - small filters and an autoencoder structure. Unfortunately, this style of model was not successful, with an example of the results shown in Figure 5.5, for a run using the elevation gradients parallel and anti-parallel to the wind direction as inputs. This model used input grids of 150x150 grid points, with an autoencoder structure of 5x5, 3x3 and 3x3 filters in the encoder, a mirrored set of transposed convolution filters in the decoder, 10 channels, Sigmoid Linear Unit (SiLU) activation [68] and 2-D batch normalisation. While the speedups predicted by this model have reasonable output scales (owing to the inclusion of batch normalisation), they are still very similar to the inputs (i.e. the elevation gradients) in shape. It is possible that this model did not train for a sufficient number of epochs, or use enough training data (it was trained with data from 11 sites), when compared to models such as [52] and [50].

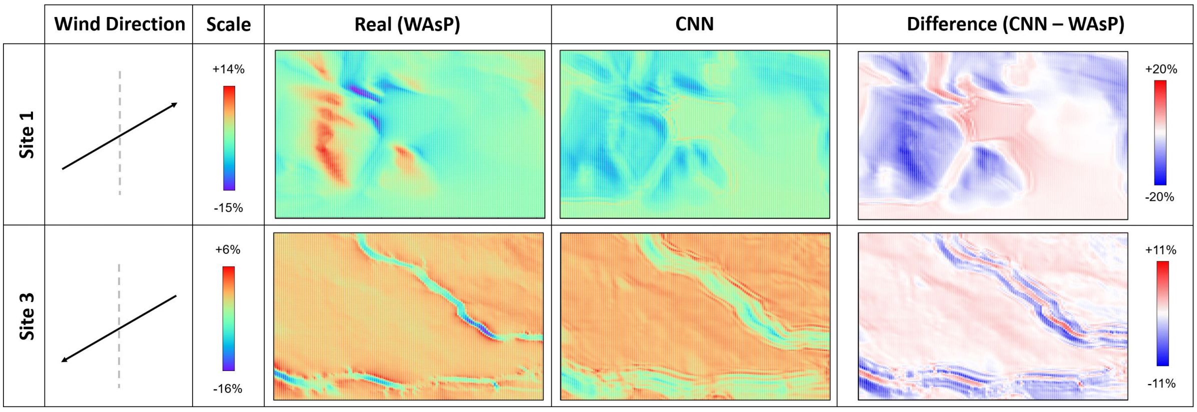


FIGURE 5.5. Results of an autoencoder CNN for predicting orographic speedup, for Site 1 (top) and Site 3 (bottom), showing (left-right) WAsP calculated speedups, CNN predicted speedups, and the difference between them.

5.3.1 Best CNN Model

Despite using different combinations of input variables, numbers of channels, and filter sizes, many of the CNNs tested collapsed, giving unrecognisable outputs. From the more successful tests, the architecture that most reliably produced reasonable results consisted of an encoder-type set of 2-D convolutional layers with small filters that decreased in size with successive layers, followed by a transposed convolutional final layer. The results from the most promising CNN are shown in Figure 5.6; as with the previous tests described there is still a clear correlation between the predicted speedups and the input elevation gradients (especially in Site 1), but for Sites 2-4, the scale and overall shape of the speedups are reasonable. This model used the elevation gradients parallel and perpendicular to the wind direction, with architecture described in Table B.1. It is possible that the "right" combination of convolutional/normalisation/pooling layers, activation functions, loss, training epochs, and other hyperparameters was not found in these tests, or that more training data would improve the results.

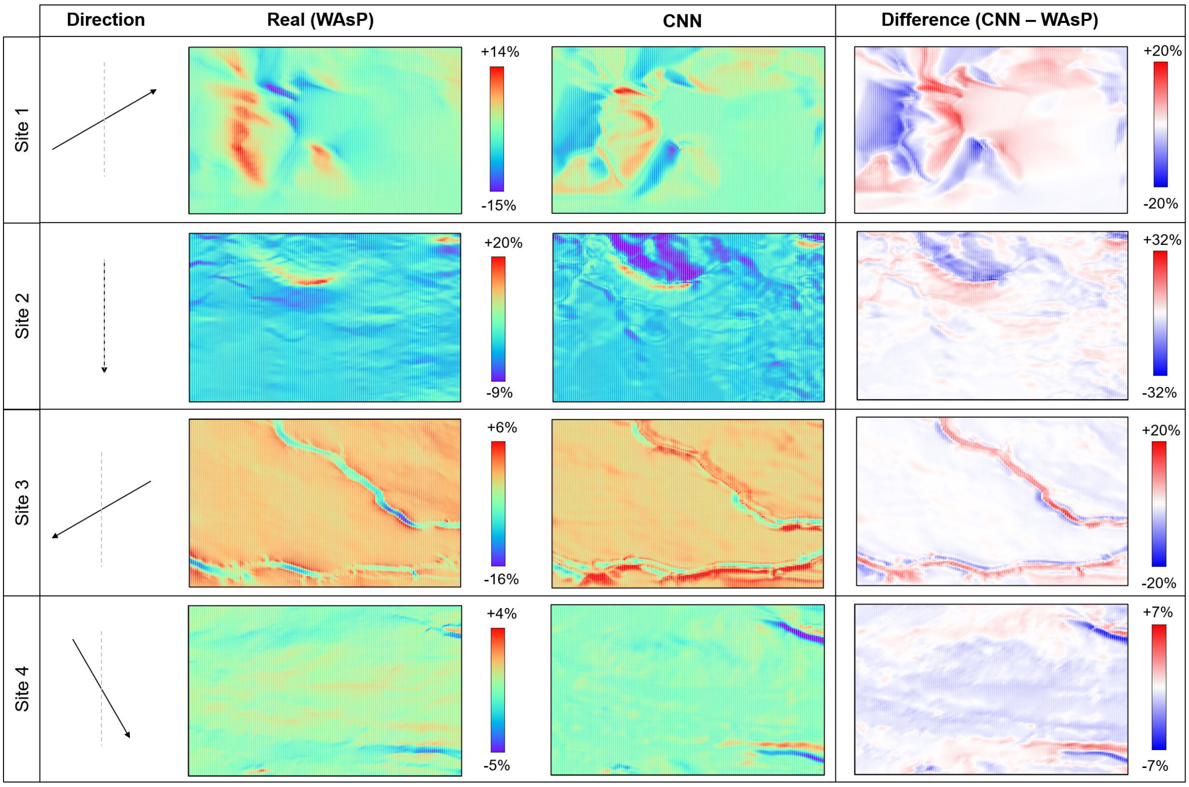


FIGURE 5.6. Results of the most successful CNN for predicting orographic speedup, for (top-bottom) Sites 1-4, showing (left-right) WAsP calculated speedups; CNN predicted speedups; the difference between them. Note that the Site 1 sector data shown here was part of the training data rather than the validation data, but is included here for consistency.

Investigation into how to train CNNs with relatively small data sets and comparison of this machine learning problem to image transformation problems led to the U-Net architecture, which was trialled next.

5.3.2 U-Net Style Networks

The U-Net model [49] consists of blocks of convolutional/transposed convolutional, batch normalisation and pooling/upsampling layers in an autoencoder-style structure, but with skip connections between corresponding encoder and decoder layers. These skip connections allow the model to carry through data from earlier layers deeper into the network, which helps it to "remember" input structures. Given there is a degree of correlation between the elevation, elevation gradients and orographic speedups the U-Net structure was thought to be an appropriate architecture for this problem. U-Net style networks which were less complex than those in [49] and [50] in terms of number of channels and layers were tested here. Using "blocks" consisting of:

- **encoder**: two layers of 2-D convolutional layers with a 3x3 kernel, no padding and stride 1, followed by ReLU activation; the encoder doubles the number of channels out compared to the number in; after the encoder is a max pooling layer with kernel size 2x2 and stride 2.
- **decoder**: a 2-D transposed convolutional layer with a 2x2 kernel, no padding and stride 2; concatenation with the corresponding encoder layer's output; a 2-D convolutional layer as per those in the encoder; the decoder halves the number of channels out compared to those in.
- a final 2-D convolutional layer with kernel size 1x1 and stride 1, which outputs a single channel.

Again, none of the U-Net style models tested showed great promise, and frequently collapsed into "blocky" sections or predicted a single output value per site. The speedups predicted by of one of the more successful models are given in Figure 5.7. This model also used the elevation gradients parallel and anti-parallel to the wind direction as inputs in 350x350 grids, had three encoder and three decoder blocks (using SiLU instead of ReLU activation), included batch normalisation in the encoder layers, and was trained using data from six sites. While there are some small areas of the predicted speedup that have the correct shape, overall this method still has not properly learned the transformation between the inputs and outputs, and is too similar to the input maps.

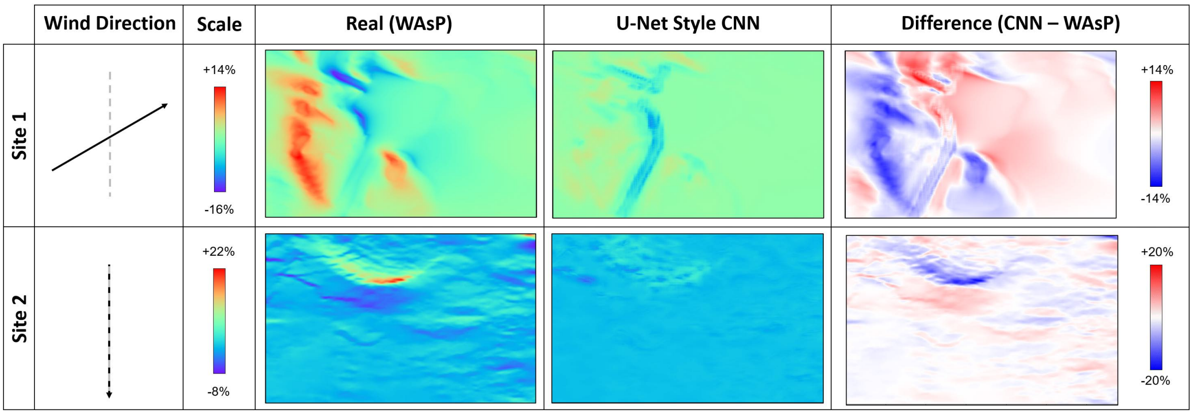


FIGURE 5.7. Results of a U-Net style CNN for predicting orographic speedup, for Site 1 (top) and Site 2 (bottom), showing (left-right) WAsP calculated speedups; CNN predicted speedups; the difference between them.

The performance metrics for the CNNs presented here are given in Table B.1. While this U-Net style model has better performance metrics than the previous CNNs described, the plots of the outputs from the models show that the CNN described in Section 5.3.1 produced outputs that are closer to the ground truth. The U-Net style model predicted almost uniform speedup values across the terrain, and produced smaller output grids than the previous CNNs (an artefact of the U-Net approach), which could be the cause of these lower errors without a corresponding improvement in performance.

After a lack of success with the U-Net architecture, a more systematic approach to CNN design was suggested, namely an investigation into the appropriate filter sizes needed. This led to revisiting and expanding on the grid approach started with the *Single Point Neural Network* in Section 5.2.

5.4 Grid Neural Networks

5.4.1 Motivation and Method

The Grid Neural Network method was created to investigate the spatial extent of terrain orography information required to calculate orographic speedup, which could then be used to inform filter sizes in CNNs for predicting orographic speedup. The Grid NN method applies sub-grids to the input and target output data grids, and passes the points from these sub-grids into a Deep Neural Network. The steps of this method (as shown in Figure 5.8) are:

1. take in a grid (or grids) of input data (e.g. elevation, elevation gradient and roughness);

2. split each input data grid into as many (square) sub-grids of a specified size (e.g. 15x15 points) as possible;

3. take a given pattern of points from each sub-grid (and the wind direction), and use these as inputs to a fully-connected, feed-forward DNN;

4. take another pattern of points (or a single point) from corresponding sub-grids of the output data, and use these as target outputs for the DNN.

This Grid NN technique incorporates spatial information into a Deep Neural Network approach, and is designed to facilitate testing of a range of input sub-grid sizes (which are analogous to CNN filters) to determine the "radius of influence" of the input variables on the output. An optimal sub-grid size determined using this method could be used to inform the filter size for future CNN tests, as well as being a generally useful finding about the wind flow characteristics in the target model.

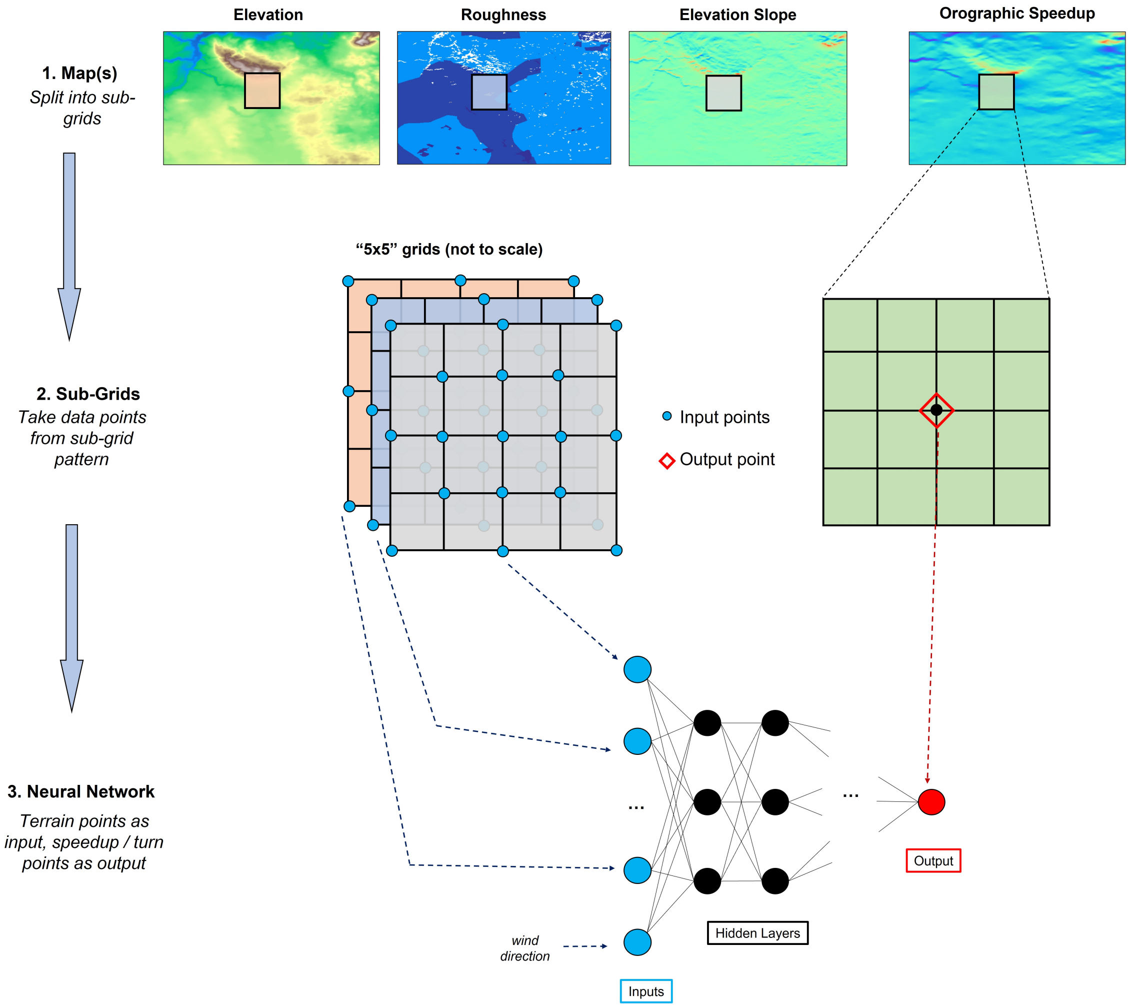


FIGURE 5.8. Grid Neural Network schematic, showing the stages: 1) splitting the input and output maps into sub-grids; 2) taking a pattern of points from each sub-grid; 3) passing the sub-grid points as inputs and outputs to a Deep Neural Network.

While sub-grids of different sizes can be applied to each input variable, investigating all combinations of sub-grid sizes and input variables would take too much time to be achievable in this work. Hence, all input variables used in a Grid NN run have the same input sub-grid size, and use the same pattern of input points from each sub-grid. Unless specified in the description, all Grid NN tests here calculated a single output point, and used a consistent pattern of input sub-grid points. Figure 5.9 shows examples of input and output sub-grid patterns.

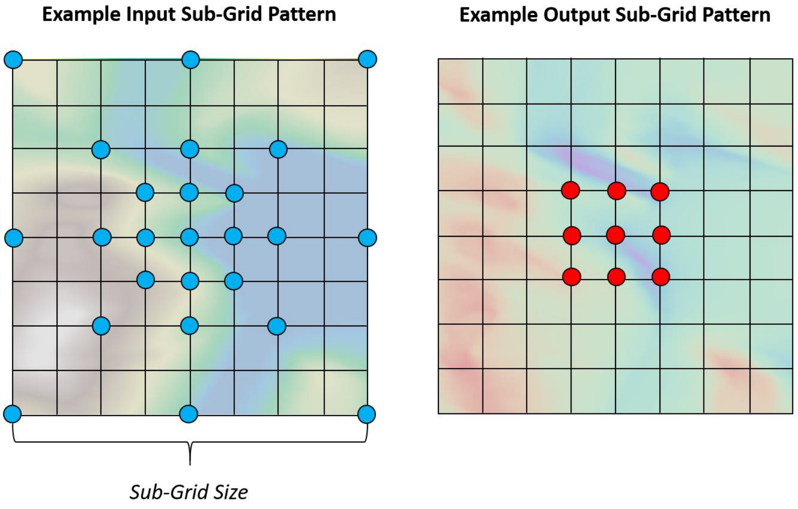

Example Input Sub-Grid Pattern
Example Output Sub-Grid Pattern
Sub-Grid Size

FIGURE 5.9. Example Grid NN sub-grid patterns for input (left) and output (right).

It is difficult to class one type of Neural Network as more "complicated" than another, but the Grid NN approach could have advantages over CNNs in terms of understandability, as the sub-grid size is a clear measure of the important spatial dimensions that relate the input data to the output data. The Grid NN method takes in a 1-D vector of input data points by splitting the input grids, resulting in around 150,000 pairs of input-output data for training and validation per site and sector, and hence millions of input-output data pairs over 11 sites and all sectors. By contrast, the CNNs described in Section 5.3 above use 2-D grids of inputs (with one input variable per channel, similar to RGB images), giving around 250 pairs of input-output data[2] for training and validation. Having more data pairs in the Grid NN model means that this model will have substantially more iterations of updating the model parameters (i.e. learning) than a CNN using the same input data.

5.4.2 Sub-Grid Size

To determine the optimal sub-grid size for a Grid NN model predicting orographic speedup, a range of sub-grid sizes were tested, using a single input variable type for each run (the Grid

[2] Assuming two random grids of the specified input size are taken from each input grid.

NN model could not take in multiple input variables at this stage), and data from a single site. The DNNs of these Grid NNs comprised of two hidden layers with five neurons each, both using ReLU activation.

A selection of the most relevant sub-grid sizes are given in Figure 5.10 for Site 1, using a input sub-grid pattern of a 3x3 points, a single point output, and input variables of min-max normalised elevation (top) and elevation gradient parallel to the wind direction (bottom). Results from a corresponding set of tests using data from Site 2 instead, and using a denser pattern of points per sub-grid (17 points rather than the nine for the Site 1 tests) are given in Figure 5.11. The error statistics for both sets of tests are given in Table B.2. Note that the sub-grid size refers to the sub-grid side length, as shown in Figure 5.9. In both of these figures, the output from the smallest sub-grid size models is strongly correlated to the input data, scaled to match the orographic speedup values, as seen with the single point NN in Figure 5.4. Using the largest sub-grid size (2.7km square) the elevation gradient input tests for both sites start to collapse, with the predicted speedups lacking detail. The 1.2km sub-grid tests with elevation gradient input give the best speedup maps for both sites. Taking the results from Site 1 as an example (Table 5.1), using a sub-grid size of 1.2km gives the lowest MAE and error standard deviation with the elevation gradients as input. For Site 1, the speedups produced using elevation gradient as input are superior to those from the elevation as input (despite having larger errors overall), but for Site 2 there is a stronger influence of the elevation itself on the orographic speedup (see the valley shapes in the top left of the Site 2 speedup map in Figure 5.11). At this point, the Grid NN model was only able to take in one input variable per run, so the optimum sub-grid size was taken to be 1.2km square, with elevation gradient as the input.

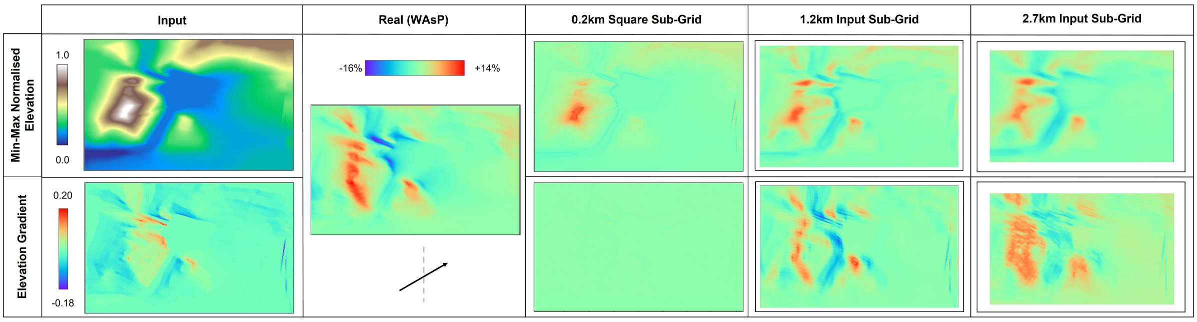


FIGURE 5.10. Grid NN sub-grid size tests using Site 1 data at 100m AGL, with inputs of (top) min-max normalised elevation and (bottom) elevation gradient parallel to the wind direction, and sub-grid sizes of (left-right): 0.2km; 1.2km; 2.7km.

These results provide some insight into how the Grid NN approach works - the input maps are "re-printed and re-scaled", with increasing sub-grid sizes able to "re-print" the input data at larger distances from the output point. This leads to very small sub-grids effectively repeating the input data as output, with the scales adjusted to match the outputs; sub-grids that are too

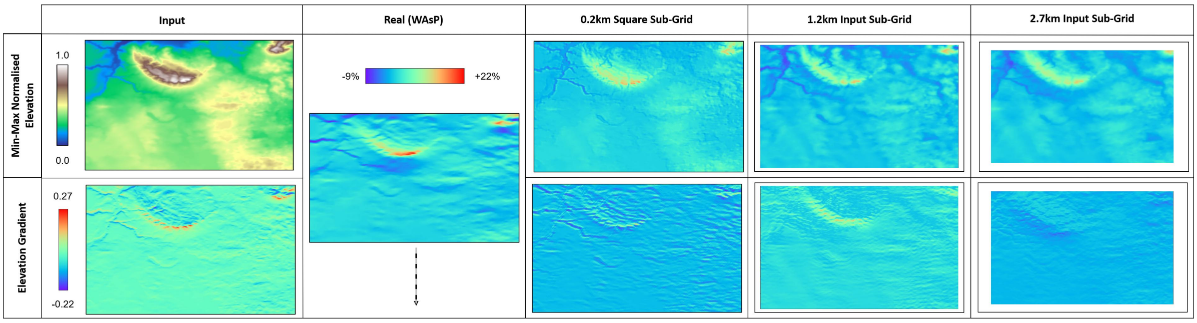


FIGURE 5.11. Grid NN sub-grid size tests using Site 2 data at 100m AGL, with inputs of (top) min-max normalised elevation and (bottom) elevation gradient parallel to the wind direction, and sub-grid sizes of (left-right): 0.2km; 1.2km; 2.7km. These tests use a denser grid pattern than for Figure 5.10.

Run ID	Input	Sub-Grid Size (km)	MAE	Error Std. Dev.
OS-S-1	Min-Max Elevation	Single Point	1.43	1.98
OS-G-1	Min-Max Elevation	0.2	1.40	1.95
OS-G-2	Min-Max Elevation	1.2	**1.31**	**1.84**
OS-G-3	Min-Max Elevation	2.7	1.36	1.89
OS-S-2	Elevation Gradient	Single Point	1.67	2.52
OS-G-4	Elevation Gradient	0.2	1.68	2.54
OS-G-5	Elevation Gradient	1.2	**1.56**	**2.29**
OS-G-6	Elevation Gradient	2.7	2.24	2.67

Table 5.1: Performance metrics for orographic speedup Grid NN sub-grid size investigations in % speedup for Site 1; minimum values for each type of input are highlighted.

large produce "blurred" speedup maps as the input points are too far from the output point, which is seen in the 2.7km square grid using elevation gradient in Figure 5.10.

CNN models were built which used this optimal filter size, but they did not produce any successful results. As such, work continued on the Grid NN models rather than returning to CNNs for orographic speedup.

5.4.3 Improvements to Baseline Model

Figure 5.12 shows the evolution of the Grid NN models for orographic speedup at 100m AGL, starting from models which have:

- input sub-grid size of 1.2km square, with elevation gradient parallel to the wind direction as input (as determined in the previous section);

- output of a single point value (at the centre of each sub-grid) of orographic speedup;

- a DNN with 50 neurons arranged as five hidden layers of 10 neurons each, with batch normalisation and ReLU activation after each layer;
- training with data from a single site, for 50 epochs;
- Adam optimiser with MSE loss for training.

The full performance metrics of each model tested in Figure 5.12 are given in Table B.3, and are summarised in Table 5.2. Starting from the model configuration known as *Baseline A* (the second from top set of maps in Figure 5.12, where three separate models have been trained with data from Sites 1, 2 and 3), the first change was to include the min-max normalised elevation and the roughness data for each site as inputs - the Grid NN code had been updated at this point to take input maps from multiple variables. This made a significant difference for all sites shown, in particular for Sites 1 and 3 where the shapes of the predicted speedups are visibly closer to the real (WAsP) outputs. Following this, the next change was to train each Grid NN model with data from four different sites, rather than individual sites - this was a necessary change if the Grid NN approach was to be considered for any real-world applications, and should prevent overfitting to data from a single site. The orographic speedups predicted from this model were an improvement over the Baseline A models.

Run ID	**Training Sites**	**Inputs**	**Neurons**	**MAE**	**Error Std. Dev.**
OS-G-13	1	Elevation Gradient	50	1.36	1.86
OS-G-16	1	Elevation Gradient, Min-Max Elevation, Roughness	50	1.09	1.51
OS-G-19	1-4	Elevation Gradient, Min-Max Elevation, Roughness	50	0.790	1.09
OS-G-23	1-4	Elevation Gradient, Min-Max Elevation, Roughness	500	0.692	**0.928**
OS-G-24	1-11	Elevation Gradient, Min-Max Elevation, Roughness	500	**0.691**	1.11

Table 5.2: Performance metrics for orographic speedup Grid NN improvements in % speedup; minimum values are highlighted.

Expanding the DNN section of the Grid NN model to include 500 neurons rather than 50 (arranged in 10 hidden layers of 50 neurons, with batch normalisation and ReLU activation after each layer as per Baseline A) gave superior performance, which can be seen from the metrics in Table 5.2. For Sites 1 and 2, the speedup shapes and scales appear closer to the WAsP outputs, and are smoother over the terrain compared to the 50 neuron model. However, there are areas of overprediction along the valley edges in Site 3. Finally, additional training data was used so that the Grid NN models trained on data from 11 sites (Sites 1-11 in Appendix A); this model is marked as *Baseline B* in Figure 5.12. This improved the detailed shape of the speedups in Sites 2 and 3 compared to the previous model, and addressed the overprediction at the valley edges in Site 3. While the MAE decreased with more training sites, the error standard deviation increased, which could be due to the corresponding larger range of validation sites over which the error was measured. Comparing the results between Baseline A and Baseline B, there is significant improvement in the scales, shapes and detail of the orographic speedups with the changes applied in Baseline B.

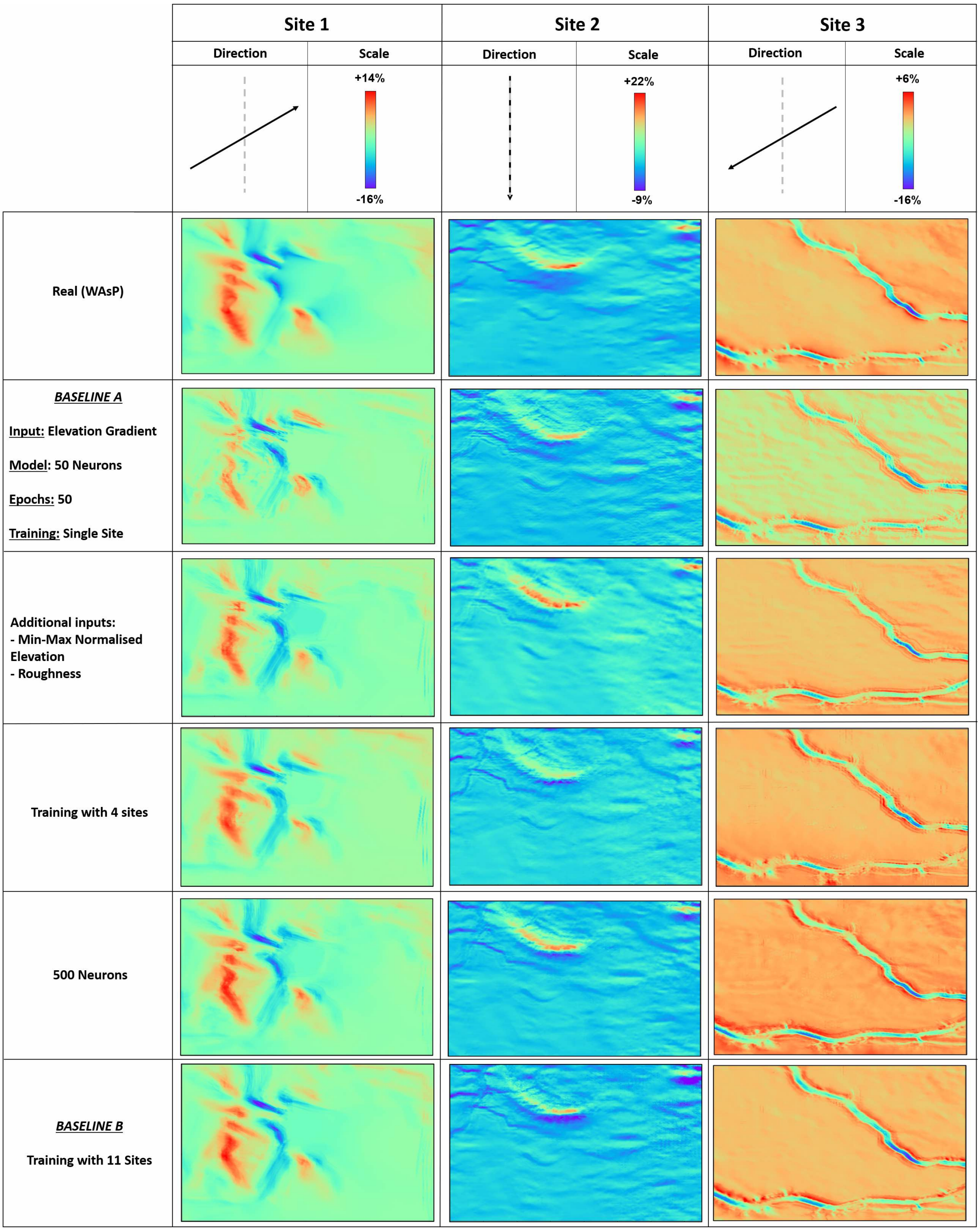


FIGURE 5.12. Evolution of the Grid NN models for predicting orographic speedup at 100m AGL, for (left-right) Sites 1, 2 and 3. These tests show (top-bottom): WAsP output; baseline Grid NN model with 50 neurons, using single site data, elevation gradient parallel to the wind direction as input; adding roughness and min-max normalised elevation; training each model with data from four sites; enlarging the model to have 500 neurons; training each model with data from 11 sites.

There were additional investigations which were not successful, i.e. changes trialled that were not kept in for the Baseline B models, the metrics of which are reported in Table B.3. These were: training for 100 epochs; using z-scale elevation normalisation; using an autoencoder-style DNN architecture.

5.4.4 Grid NN Investigations

Using the same Grid NN parameters as defined for Baseline B above, but training on data at 10m AGL, gave the Baseline B model for orographic speedup at 10m AGL, which performed well (see metrics in Table B.5). The WAsP orography model extrapolates a potential flow solution over a given terrain to different heights AGL, so it is not too surprising that the Grid NN configuration that works well for 100m AGL is applicable at 10m AGL as well. A series of sensitivity studies were undertaken on the Baseline B models with the results detailed below, with the MAE and error standard deviation collated in Tables 5.3 and 5.4 for the tests at 100m AGL and 10m AGL respectively.

Run ID	**Neurons**	**Rotate?**	**Balancing**	**Batch Normalisation**	**MAE**	**Error Std. Dev.**
OS-G-24	500	No	-	Yes	**0.691**	**1.11**
OS-G-25	Autoencoder	No	-	Yes	0.721	1.17
OS-G-26	50	No	-	Yes	0.926	1.43
OS-G-27	500	Yes	-	Yes	0.793	1.43
OS-G-28	500	No	-	No	1.17	2.17
OS-G-29	500	No	Equal	Yes	0.904	1.41
OS-G-30	500	No	Normal Distribution	Yes	0.905	1.34

Table 5.3: Performance metrics for orographic speedup Grid NN investigations in % speedup at 100m AGL; minimum values are highlighted.

Run ID	**Neurons**	**Rotate?**	**Balancing**	**Batch Normalisation**	**MAE**	**Error Std. Dev.**
OS-G-31	500	No	-	Yes	**1.89**	**3.09**
OS-G-32	Autoencoder	No	-	Yes	1.92	3.16
OS-G-33	50	No	-	Yes	2.41	3.27
OS-G-34	500	Yes	-	Yes	1.95	3.25
OS-G-35	500	No	-	No	2.83	6.05

Table 5.4: Performance metrics for orographic speedup Grid NN investigations in % speedup at 10m AGL; minimum values are highlighted.

5.4.4.1 Network Size

The first sensitivity study was a repeat of the network size testing detailed in Section 5.4.3, and was carried out to ensure that the Baseline B network size of 500 neurons in uniformly-sized hidden layers did not have excess capacity, which could lead to overfitting. The three network sizes tested are as shown in Figure 5.13, and the results for Site 1 at 100m AGL and 10m AGL are in Figures 5.14 and 5.15 respectively. At both heights, 50 neurons is an insufficient network size to capture the orographic speedup calculations, and gives poor results and a significant increase in the MAE from the 500 neuron model, e.g. from 0.69% to 0.93% speedup at 100m AGL. When comparing the autoencoder structure to the 500 neuron architecture, it is difficult to decide on which gives better results by eye, either from the speedups or the error maps, but the 500 neuron models do have lower error values in Tables 5.3 and 5.4. Additionally, looking at the loss curves in Figure 5.16 (this is for 10m AGL; the curves are similar for 100m AGL), the training losses are smaller for the 500 neuron model, and the output plots show no sign of overfitting.

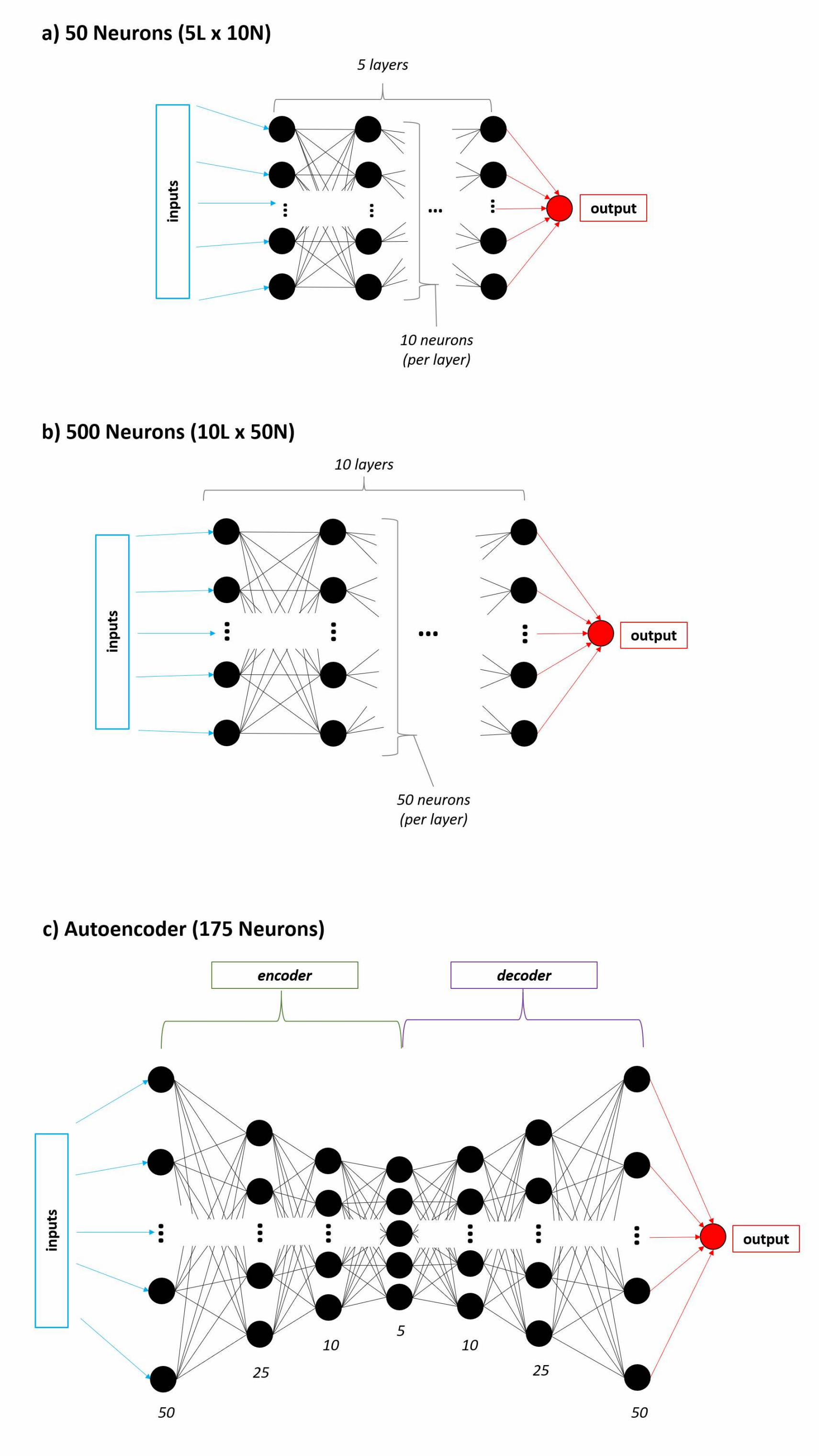


FIGURE 5.13. Schematic of Grid NN architectures tested, showing a) 50 neuron model; b) 500 neuron model; c) autoencoder.

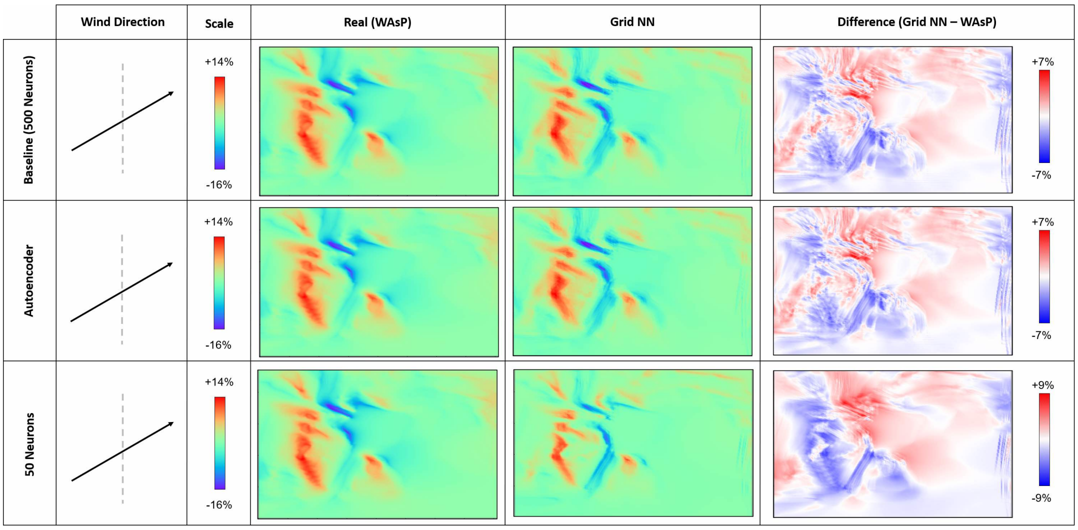


FIGURE 5.14. Comparison of different DNN architectures within the Grid NN models for orographic speedup at 100m AGL, showing (left-right) the WAsP speedups, Grid NN speedups, and the difference between them for (top-bottom): Baseline B model with 500 neurons; an autoencoder architecture; 50 neurons.

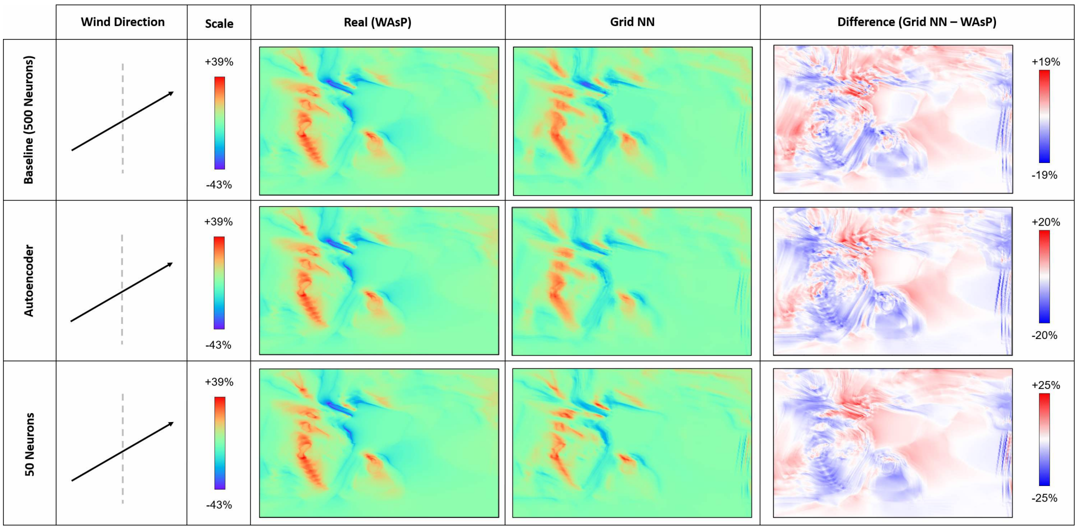


FIGURE 5.15. Comparison of different DNN architectures within the Grid NN models for orographic speedup at 10m AGL, showing (left-right) the WAsP speedups, Grid NN speedups, and the difference between them for (top-bottom): Baseline B model with 500 neurons; an autoencoder architecture; 50 neurons.

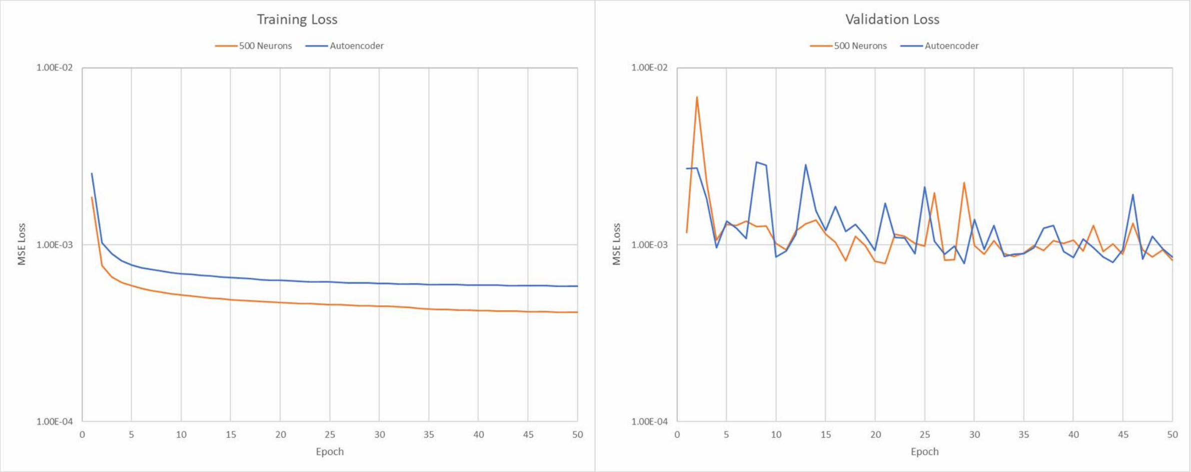


FIGURE 5.16. Comparison of the loss curves for a Grid NN for orographic speedup at 10m AGL, comparing models with 500 neurons to an autoencoder architecture; training (left) and validation (right) losses shown.

5.4.4.2 Rotation

Figure 5.17 shows the speedups at 10m AGL calculated by Grid NN models, one of which included the normalised wind direction as an input value to the DNN section of model (left), while the other rotated all input and output grids such that the relative wind direction was always from 0° and did not include the normalised wind direction as a specific input value (right). While the results are reasonably similar, the scales in the "rotated" run are not as extreme as those in the "non-rotated" run, e.g. the area of low speedup in the shadow of the hill in Site 2 is better predicted in the non-rotated run. The non-rotated speedups also appear smoother than the rotated speedups; from Table B.5 the non-rotated run also has better error statistics, particularly in terms of the maximum and minimum errors. Similar results are seen for a corresponding test carried out on data at 100m AGL, with the error statistics given in Table 5.3. It is somewhat surprising that the inclusion of wind direction as an input variable gives the Grid NN models sufficient information to predict the highly direction-dependent orographic speedup. But since the elevation gradient input is resolved parallel to the wind direction (from the Baseline A results in Figure 5.12 this alone as an input to a Grid NN gives reasonable speedup predictions) there is directional information provided to the Grid NN models without rotating the input and output grids.

5.4.4.3 Batch Normalisation

Figures 5.18 and 5.19 demonstrate the importance of including batch normalisation in the Grid NN models at 100m and 10m AGL respectively. Without batch normalisation, the speedups

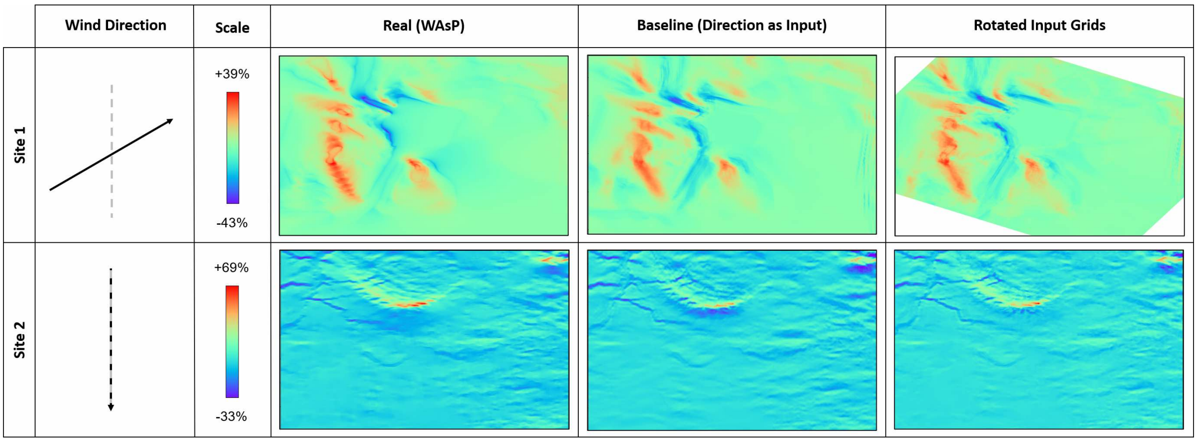


FIGURE 5.17. Comparison of Grid NN models for orographic speedups at 10m AGL, showing (left-right) WAsP; Grid NN with wind direction as an input; Grid NN with input and output grids rotated to have consistent wind direction, for (top) Site 1 and (bottom) Site 2.

predicted by the models resemble the elevation terrain maps, and do not match the WAsP calculated speedups in shape or scale. From Tables 5.3 and 5.4 there are significant increases in both MAE and error standard deviation when batch normalisation is removed, e.g. the increase in error standard deviation from 3.09% to 6.05% speedup at 10m AGL. Although the scales of all the input variables are roughly similar, it could be that the batch normalisation layers help to balance the effects of the different inputs, and so without it a single type of input dominates, which in this case is the elevation.

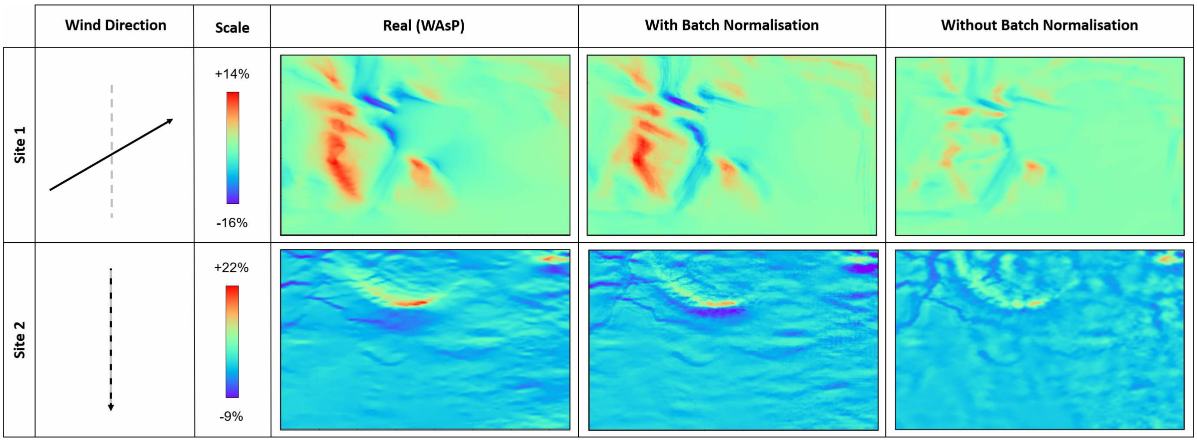


FIGURE 5.18. The effect of including batch normalisation layers in a Grid NN model for speedups at 100m AGL, with (left-right) WAsP speedups; Grid NN with batch normalisation; Grid NN without batch normalisation, for (top) Site 1 and (bottom) Site 2.

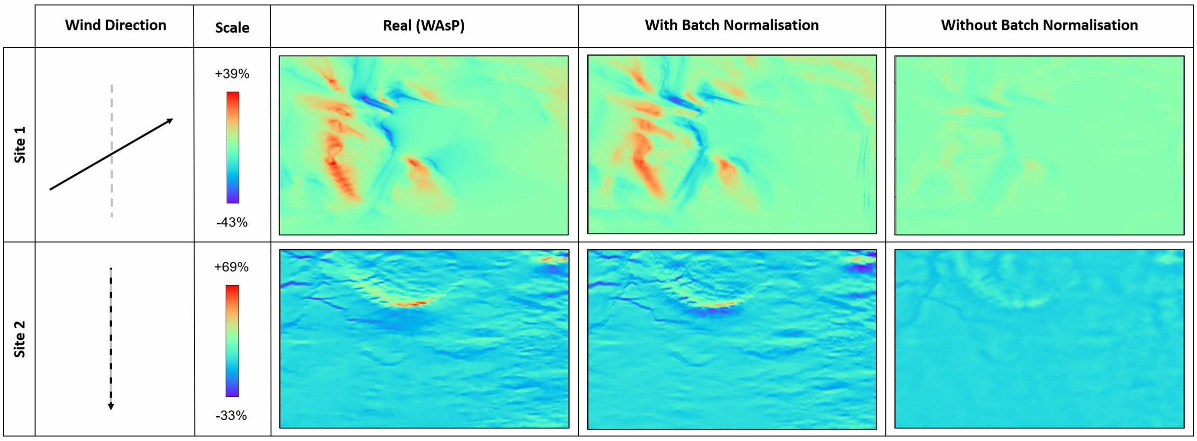


FIGURE 5.19. The effect of including batch normalisation layers in a Grid NN model for speedups at 10m AGL, with (left-right) WAsP speedups; Grid NN with batch normalisation; Grid NN without batch normalisation, for (top) Site 1 and (bottom) Site 2.

5.4.4.4 Balancing Speedup Data

A significant portion of the terrain data available for training in this work consists of plains, i.e. areas of little change in elevation; as such there is minimal change in wind speed due to orography over these areas. Examples of this include Site 2 and Site 3, which both consist of mostly flat terrain (in terms of elevation) with distinct features such as the isolated hill in Site 2 and the narrow valleys in Site 3 (Figure 5.12). The prevalence of flat terrain areas means that a significant fraction of the training data for models has little useful information, i.e. flat elevation with ~0 gradient giving 0% orographic speedup. To combat this, tests were run with the training data *balanced* such that more input-output pairs with high speedup values were used to train the model. The total number of training data points was unchanged and the validation data was unaffected. Figure 5.20 (a) shows the distribution of orographic speedup values over the training data set in question.

The first method of balancing, termed *uniform balancing* here, balanced out the bars of the histogram in Figure 5.20 (a), giving instead a distribution as in Figure 5.20 (b), where the bins of orographic speedup have the same limits but there is an equal number of data points in each bin. Any histogram bins with more data points than required had an appropriate number of random points removed from the data set. Histogram bins with fewer data points than needed had their existing data points replicated until the bin had the required size. This meant that the histogram bins at the extreme ends of the orographic speedup ranges, which had the fewest number of data points in the original data set, consisted of the same data repeated many times, which may have resulted in a degree of overfitting on these points.

The alternative balancing method trialled here was intended to lessen the artificial repetition of data points at the extreme ends of the histogram in Figure 5.20 (b). Instead, a normal distribution was fitted to the original orographic speedup data set, characterised by its mean and standard deviation (Figure 5.20 (a)). For this balanced data set, the mean was unchanged, but the standard deviation set to a fifth of the total data range, aiming to give a larger spread of data points (Figure 5.20 (c)); this was termed the *normal distribution balancing* method. Again, the limits of the histogram bins were unchanged, and the mean of each bin was used to calculate the number of data points that should be present per bin in the balanced data set.

The results from Grid NN models using these balancing techniques are shown in Figure 5.21, and the performance metrics are given in Table 5.3. The MAE and error standard deviation increase with both types of balancing compared to the baseline case. For the speedups shown, there do seem to be larger errors in the predictions at the edges of terrain features, such as the valleys in Site 3 and the top of the hill in Site 2, when the training data has been balanced. Additionally, the error statistics are uniformly worse in these runs compared to a model without balanced training data (excepting the maximum error of the normal distribution balancing method, which improves on the un-balanced (baseline) model (Table B.4)).

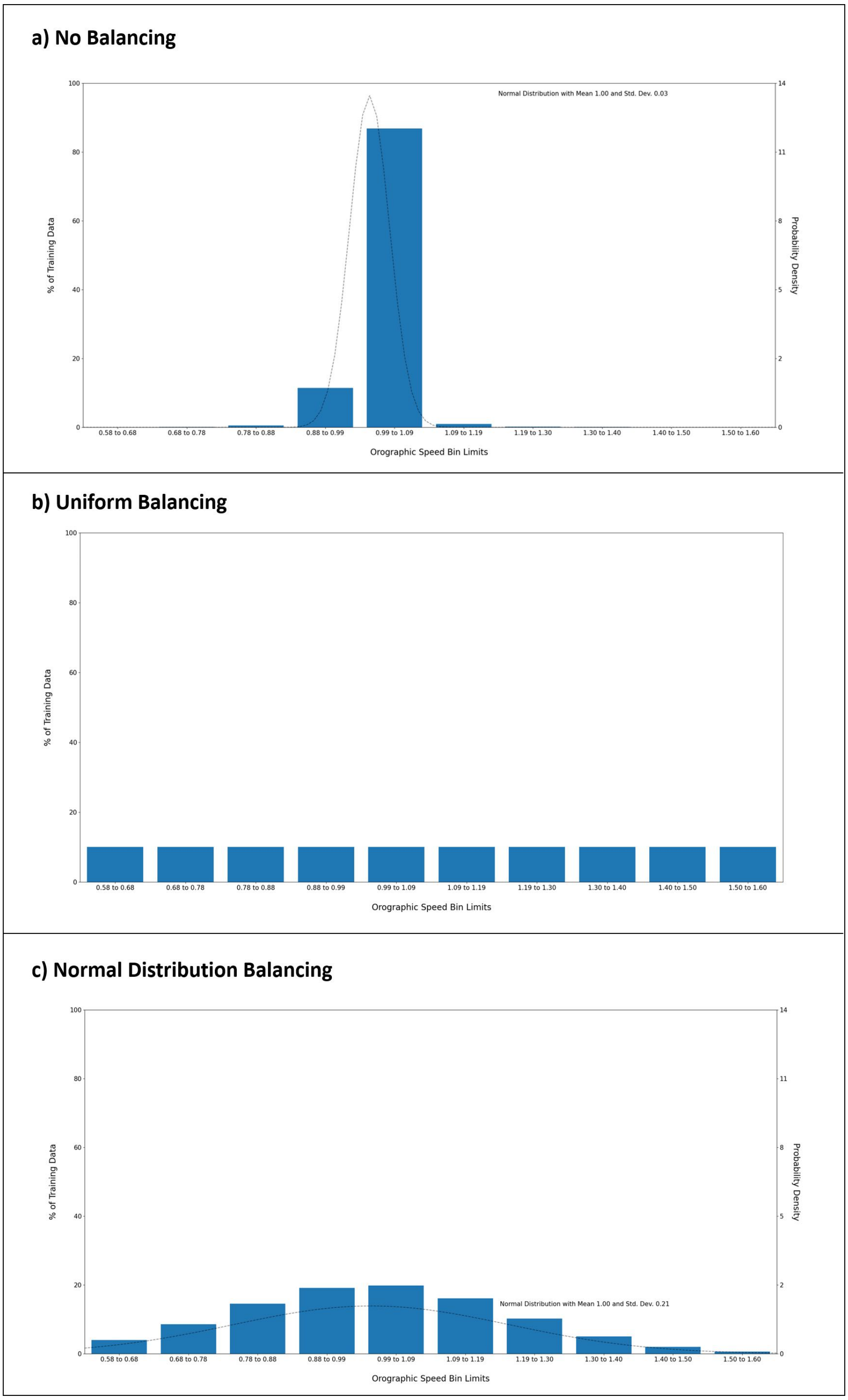


FIGURE 5.20. Histograms of orographic speedup training data at 100m AGL, with a) no balancing (and a fitted normal distribution shown); b) uniform balancing; c) normal distribution balancing (with the altered distribution shown). Note that the speedups are given as speedup factors here.

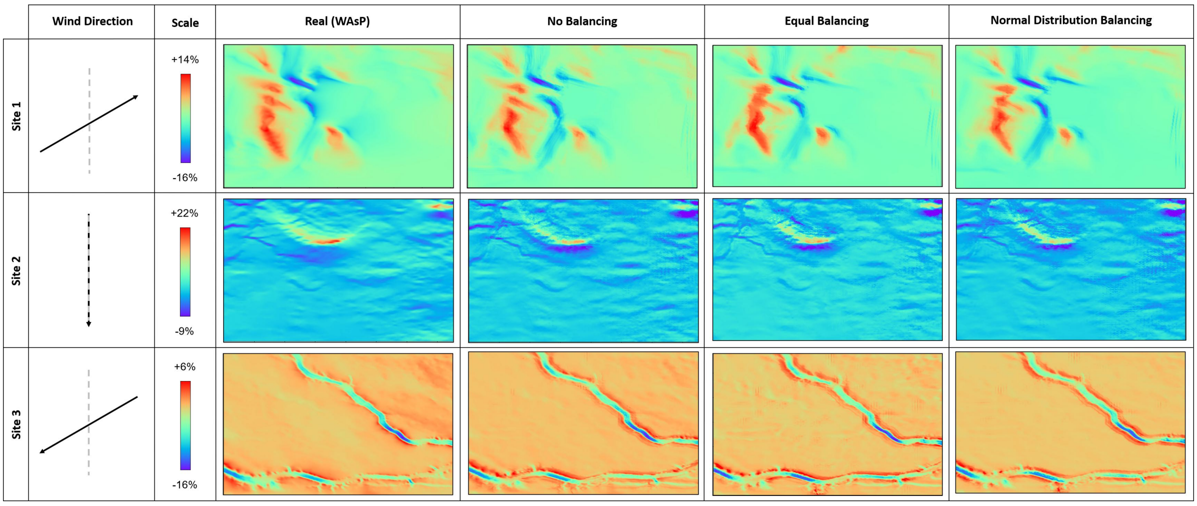


FIGURE 5.21. Comparison of orographic speedup at 100m AGL for (top-bottom) Sites 1, 2 and 3, from (left-right) WAsP; Grid NN model with no changes to the training data; Grid NN model with uniform balancing of training data; Grid NN model with normal distribution balancing of training data.

5.4.5 Final Models

While all the sensitivity studies in Sections 5.4.3 and 5.4.4 used a consistent pattern of sub-grid points as inputs, some of these sub-grid points were later found to be in different positions than intended. This issue was fixed and the final orographic speedup models at 10m and 100m AGL were re-run. Both of these models had:

- Inputs:
  - min-max normalised elevation, elevation gradient parallel to the wind direction, and roughness;
  - normalised wind direction (rather than rotating the input and output grids);
  - input sub-grid size of 1.2km square;
- Outputs:
  - single point output of orographic speedup factor;
  - no data balancing;
- Deep Neural Network:
  - 10 hidden layers of 50 neurons each with batch normalisation and ReLU activation after each hidden layer;

- Training / Validation:
  - training and validation data from 11 sites (Sites 1-11 in Appendix A) with 12 direction sectors each, where nine sectors were used for training, two for validation and one was held out for inference;
  - training of between 50 and 60 epochs, stopping at the best epoch between these limits (with patience of three);
  - training with an Adam optimiser, and MSE loss.

The loss curves of both runs show reasonable trends for training and validation (see Figure 5.22 for the loss curves at 10m AGL). For ease of comparison, the table of performance metrics is presented here for the final Grid NN models at for both heights AGL, for both validation and inference data.

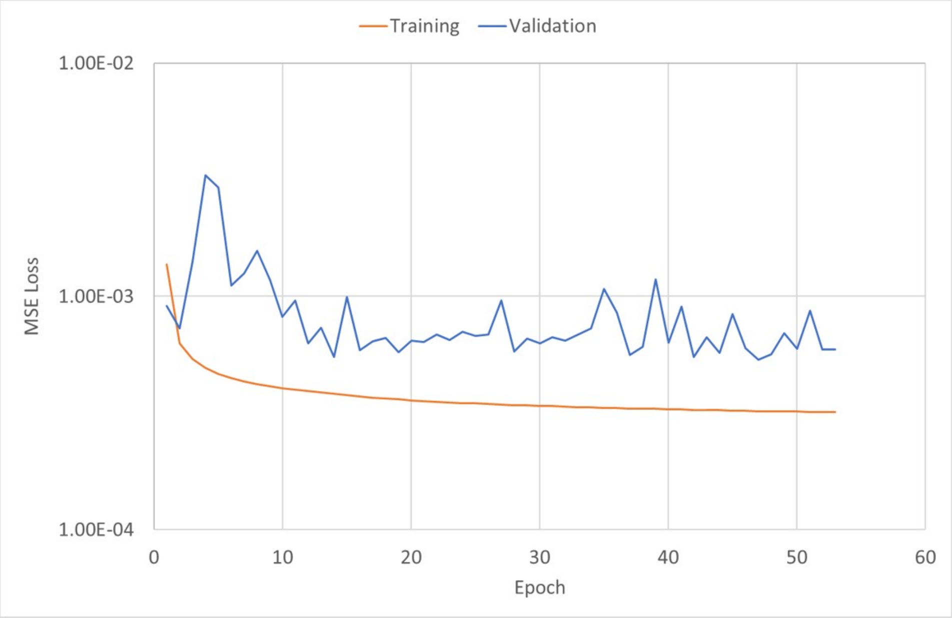


FIGURE 5.22. Loss curves for final Grid NN for orographic speedup at 10m AGL.

Run ID	Height AGL (m)	Validation / Inference	MAE	Error Std. Dev.
OS-G-36	10	Validation	1.90	2.79
OS-G-36	10	Inference (New Sites)	3.68	7.04
OS-G-37	100	Validation	0.652	1.04
OS-G-37	100	Inference (New Sites)	1.64	3.01

Table 5.5: Performance metrics for final orographic speedup Grid NN models in % speedup.

5.4.5.1 Validation

Figure 5.23 shows the orographic speedups calculated by the final Grid NN model trained on data at 10m AGL. The form and scales of the Grid NN predictions are very similar to the WAsP output speedups, but the NN output maps are less smooth than the real (WAsP) speedups across terrain. The error maps for each site have large scales, but most errors appear to be relatively small; the Grid NN model performs well across areas of flat or gradually changing terrain, and shows the highest errors at locations of steep slopes such as the edges of the valleys in Site 3. Given the relatively sparse set of input points that are used per orographic speedup value, and the fact that each point in the predicted orographic speedup maps is calculated separately, the results for Sites 1-3 are very promising. Figure 5.24 shows a scatter graph of the real (WAsP) against predicted orographic speedups for Site 1 sector 9 (wind from 240°), and shows good correlation between the machine learning model and WAsP. Site 4 has mostly flat terrain, but with a high value roughness feature in the top left of the site; the Grid NN model has emphasised the roughness in the orographic speedup predictions, which is obviously incorrect. However this effect is not seen in the other three sites, so it is difficult to determine how much this affects the overall errors. Additionally, the scales of the speedups for Site 4 are much smaller than the other sites shown, so it is possible that there are unseen roughness-based artefacts in the other plots with scales that are too small to be visible. Aside from this, the Grid NN gives accurate predictions of the orographic speedup for the rest of Site 4. The error bar chart in Figure 5.25 confirms that most of the errors are small; the full error statistics are given in Table B.6.

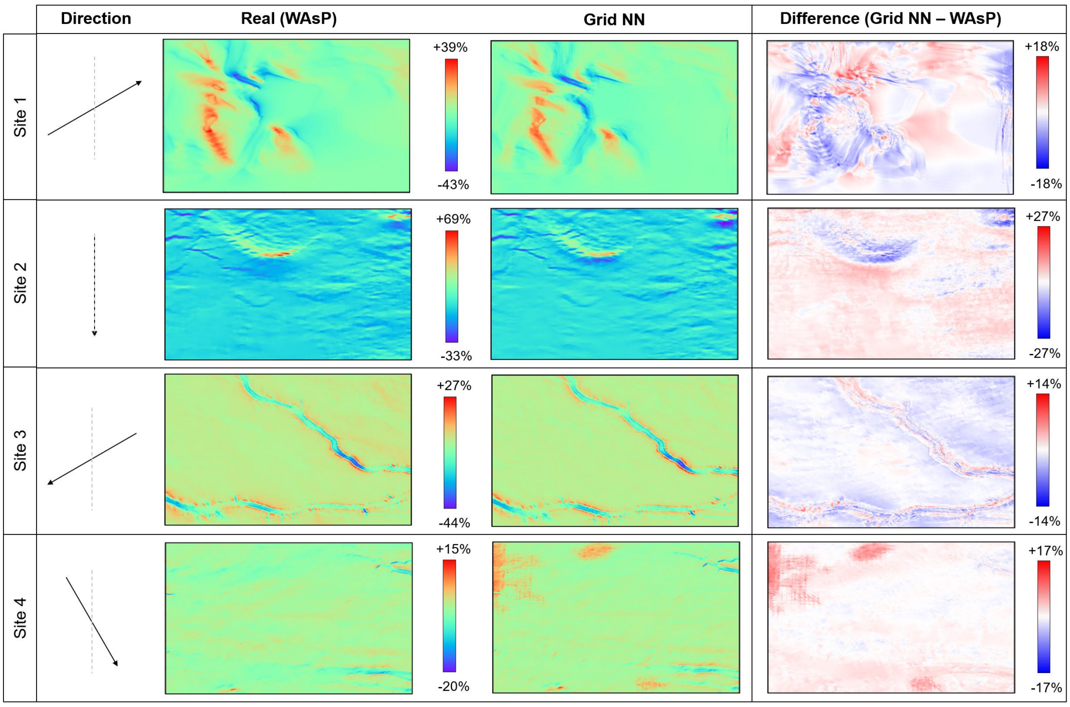


FIGURE 5.23. Final Grid NN model for predicting orographic speedup at 10m AGL for (top-bottom) Sites 1, 2, 3 and 4, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

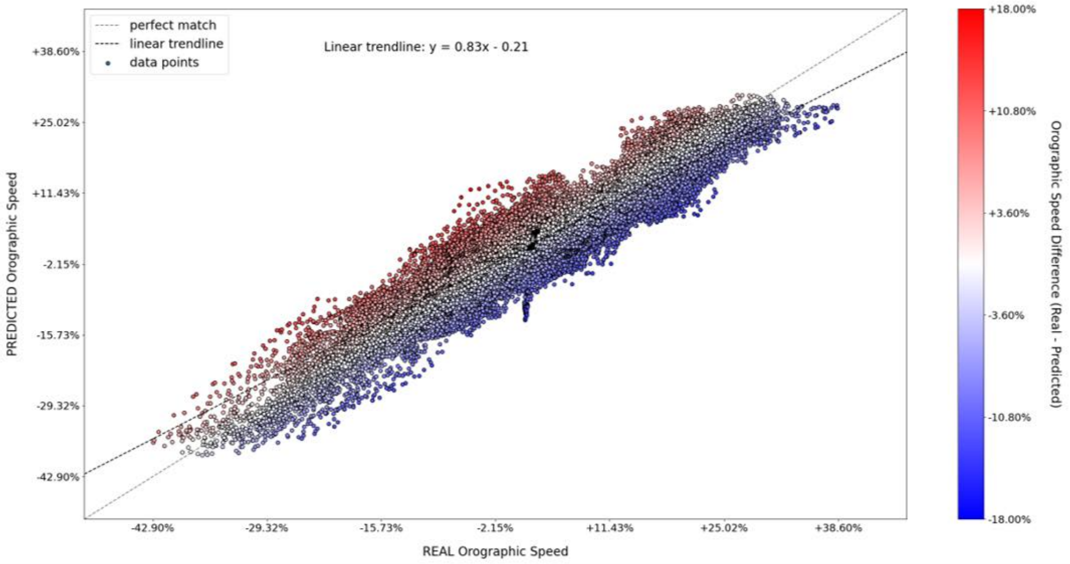


FIGURE 5.24. Scatter graph of the orographic speedup at each grid point predicted by the final Grid NN at 10m AGL compared to the WAsP calculated values, for Site 1 sector 9 (wind from 240°).

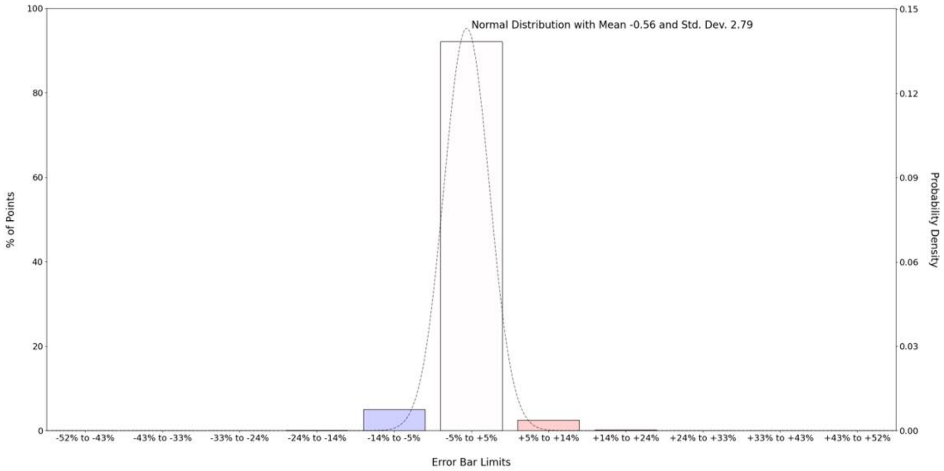


FIGURE 5.25. Bar chart of the errors (Grid NN - WAsP) in orographic speedup over the validation data using the final Grid NN model at 10m AGL.

Orographic speedups produced by the final Grid NN model trained on data at 100m AGL are shown in Figure 5.26 for Sites 1-4, and the error statistics are given in Table B.6. As with the final Grid NN at 10m AGL, the overall shapes and scales of the speedups are good, and the largest errors occur over abrupt changes in elevation. Again, this model gives accurate predictions over flat and gradually changing terrain. The orographic speedups predicted for Site 1 have quite large errors around the hill on this site, which could indicate that the input sub-grid (i.e. filter) size used is not sufficient to capture the influence of large terrain features.

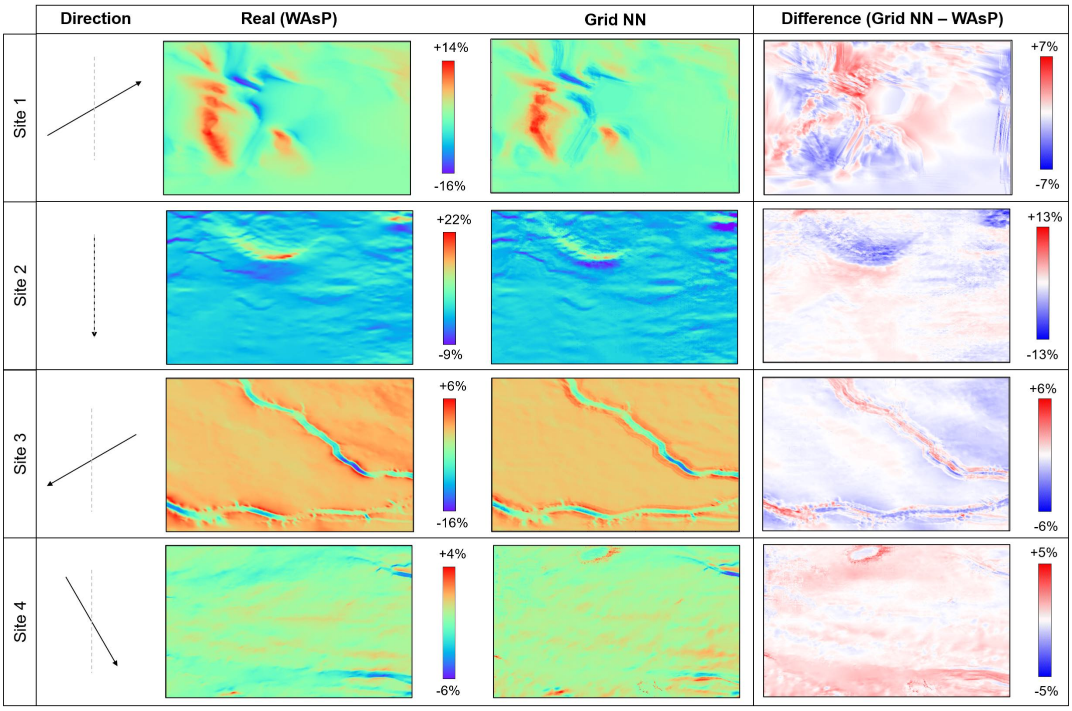


FIGURE 5.26. Final Grid NN model for predicting orographic speedup at 100m AGL for (top-bottom) Sites 1, 2, 3 and 4, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

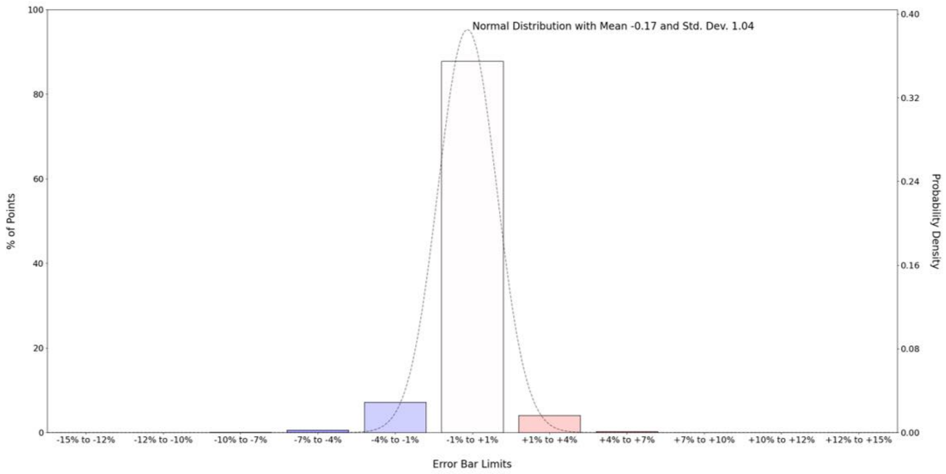


FIGURE 5.27. Bar chart of the errors (Grid NN - WAsP) in orographic speedup over the validation data using the final Grid NN model at 100m AGL.

5.4.5.2 Inference (Test Data)

Although one of every 12 direction sectors for the 11 training sites was held out for blind testing, the symmetry of the WAsP orography sub-model means that the orographic speedups for opposing wind directions are very similar. For example, if sector 1 (wind from 0°) of Site 1 had been used to train a model, and sector 7 (wind from 180°) was the test sector for this site, the model would be predicting a speedup map close to that seen in training (albeit with a different direction input). Therefore, to gauge the performance of the final orographic speedup models on truly new data, three sites (Sites 12, 13 and 14) that were not included in the training or validation data were used for inference. In terms of terrain characteristics, Site 12 is mountainous, Site 13 has a single valley feature, and Site 14 is flat but with areas of high roughness (see Appendix A).

Figures 5.28 and 5.31 show predicted orographic speedups at 10m AGL and 100m AGL from the final Grid NN models for these heights (as described in Section 5.4.5); the full error metrics are given in Table B.6. For Site 12 at 10m AGL, WAsP predicts an orographic speedup range of -99% to +124%, significantly larger than those in Figure 5.23. Figure 5.29 shows the WAsP against Grid NN predicted orographic speedup values for this site and direction, and confirms that there is generally a good match between the model and the target values, but with more underprediction of large speedup magnitudes compared to the validation data (Figure 5.24). The Grid NN model at 10m AGL has captured the general shape of the orographic speedups well, but the error map highlights that that the model has failed to capture the highest speedups at the top of the mountain ridge in Site 12. A similar issue, although less extreme, occurs at 100m AGL. In Site 14, as for Site 4, areas of high roughness in a flat terrain are emphasised in the Grid NN's orographic speedup map (see the top left corner of the orographic speedup) at both 10m and 100m AGL. The main feature of Site 13 is a single valley, which bears resemblance to

Site 3 (which has two narrow valleys). The Grid NN's speedup predictions for Site 13 correlate well in shape and scale to the ground truth for 10m AGL, but at 100m AGL there are some errors around the valley edges, similar to errors that have been discussed in this chapter.

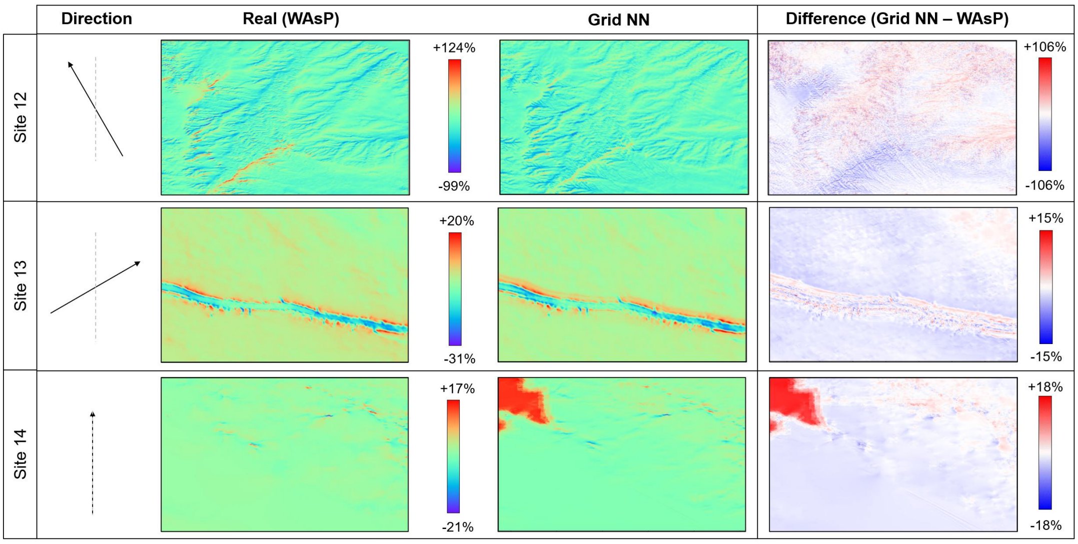


FIGURE 5.28. Final Grid NN model for predicting orographic speedup at 10m AGL for (top-bottom) Sites 12, 13 and 14, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

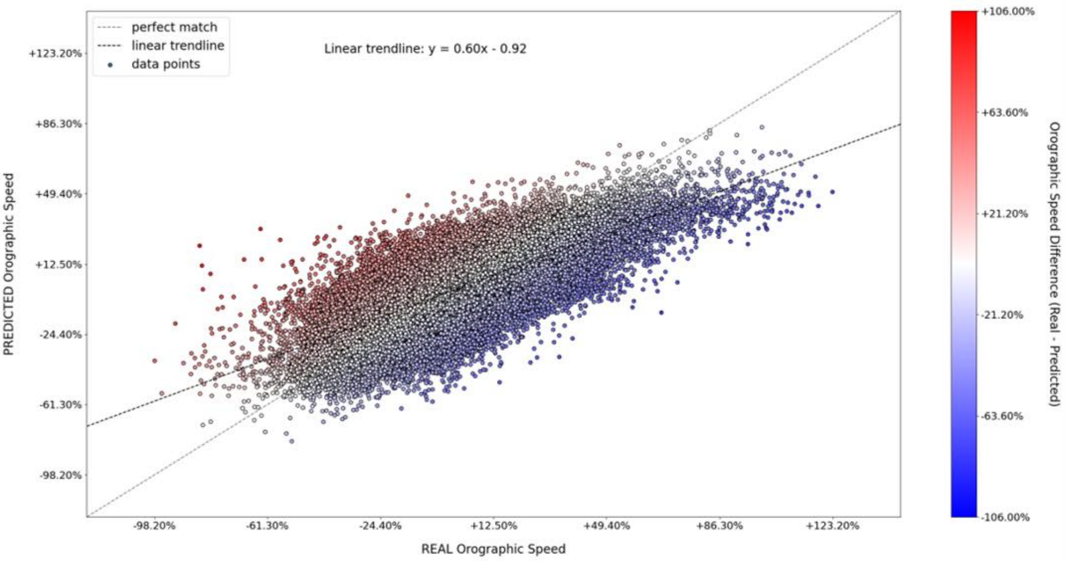


FIGURE 5.29. Scatter graph of the orographic speedup at each grid point predicted by the final Grid NN at 10m AGL compared to the WAsP calculated values, for Site 12 sector 6 (wind from 150°).

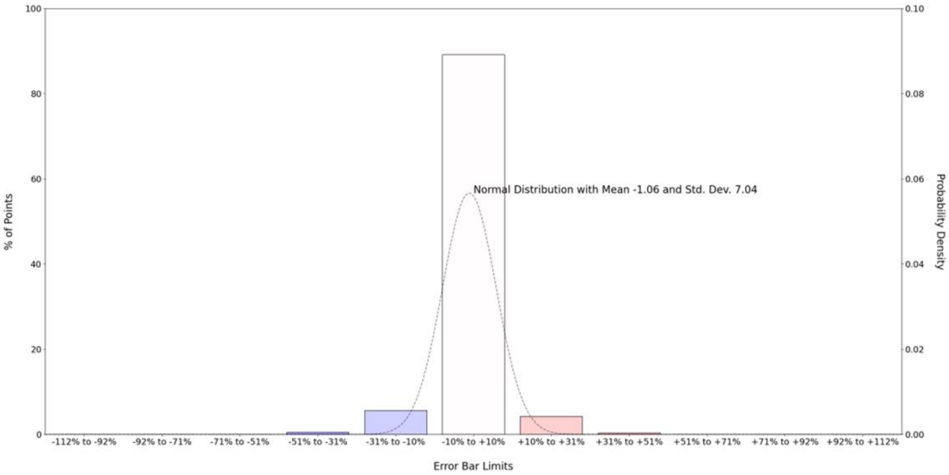


FIGURE 5.30. Bar chart of the errors (Grid NN - WAsP) in orographic speedup over three new sites using the final Grid NN model at 10m AGL.

The error bar chart in Figure 5.32 confirms that the majority of prediction errors are small at 100m AGL, with the MAE over the inference data at this height being 1.6%; for reference, the MAE over the validation data at 100m AGL was 0.65% speedup. Although the error bar chart at 10m AGL (Figure 5.30) displays a trend of small errors, the central bar containing 89% of the data points has limits of ±10% error, which is quite high. This is likely to be due to the extreme speedups at Site 12; the overall MAE over the inference data at 10m AGL is 3.7%, compared to

1.9% over the validation data, which is a large increase.

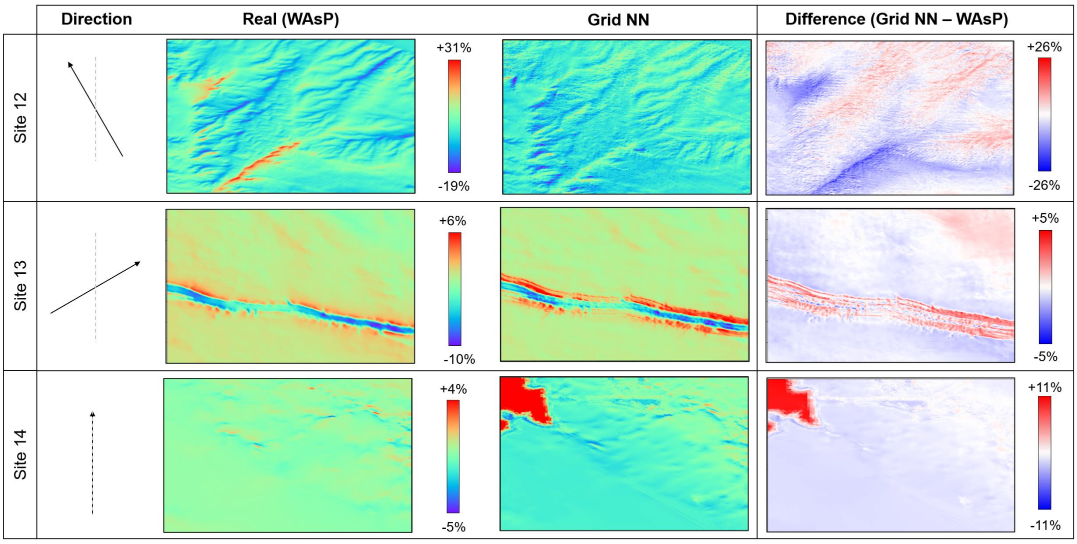


FIGURE 5.31. Final Grid NN model for predicting orographic speedup at 100m AGL for (top-bottom) Sites 12, 13 and 14, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

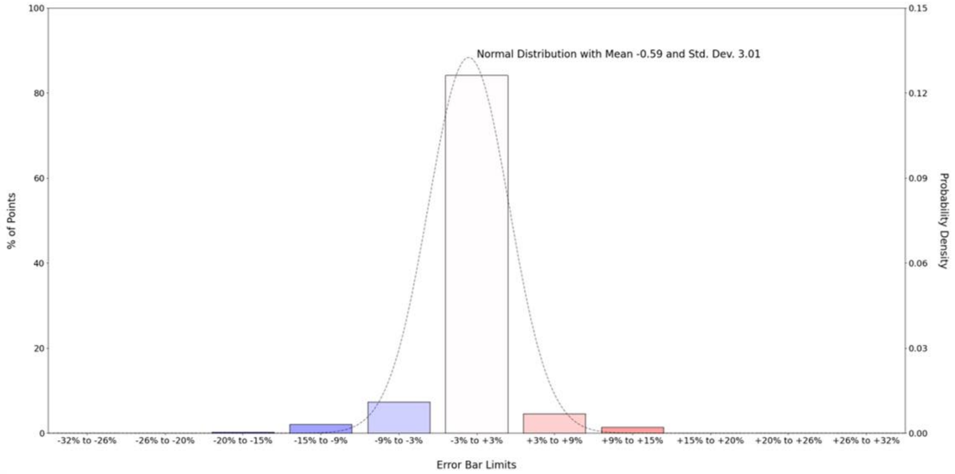


FIGURE 5.32. Bar chart of the errors (Grid NN - WAsP) in orographic speedup over three new sites using the final Grid NN model at 100m AGL.

5.5 Conclusions

This chapter introduced the Grid Neural Network method, and investigated the use of both CNNs and Grid NNs to predict the orographic speedup over terrain. The CNNs (including U-Net style networks) tested were largely unsuccessful. This work initially focussed on creating and training new CNNs, but as discussed in Section 5.3 above, the comparatively small training data set available is likely to have been a significant factor in the lack of success in finding and training a suitable CNN architecture for this problem. For context, the work of [50] used a U-Net type CNN to predict aerofoil flow with a great deal of success, with ~27,000 sets of simulation data to train on, compared to the ~250 different maps of orographic speedup available in this work. Perhaps using a pre-trained model, with a proven architecture for e.g. image transformation or aerofoil flow field prediction, would give better results if no additional training data was available.

The final Grid NN models created for speedup predictions at 10m and 100m AGL gave good results for a range of sites, using elevation, elevation gradient and roughness as inputs; examples are presented in Figure 5.33. As the Grid NN method takes in sub-grids of terrain data, using points in a pattern of concentric squares, there is a degree of similarity between this approach and the polar "zooming" grid over which WAsP calculates its orographic speedup potential flow solution [8]. There is still room for improvement in these models, particularly in smoothing maps of the output, and ensuring that the roughness inputs do not dominate over flat sites; this might be achieved by training with more sites.

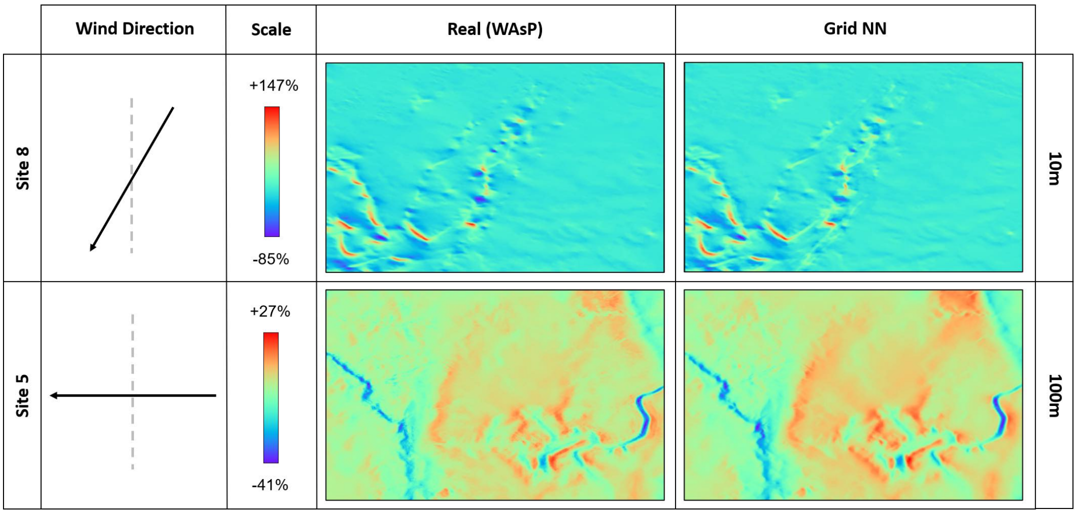


FIGURE 5.33. Examples of two sites and directions for which the final Grid NN models at (top) 10m AGL and (bottom) 100m AGL perform well. Shows the WAsP (left) and Grid NN (right) predicted orographic speedup values.

Following this investigation, the next step was to continue creating surrogate models for the orographic sub-model, but this time focussing on the orographic turn, which is presented in the next chapter.

Chapter 6

Orographic Turn Surrogate Models

The second set of investigations in this work centred on the orographic turn model. Together, the orographic speedup and turn values for a site give the change in the wind velocity due to terrain elevation, calculated with WAsP's orography sub-model. Given the success of the Grid Neural Network approach as a surrogate model for orographic speedup (Section 5.4) compared to CNNs, all investigations on orographic turn surrogate models used the Grid NN approach.

6.1 Inputs

Since WAsP uses a single orographic sub-model to calculate both the orographic speedup and turn, the inputs for the surrogate models for both of these variables were tested using the same inputs, i.e. normalised elevation, directional components elevation gradients, and roughness.

6.2 Grid Neural Networks

The Grid Neural Network method used in these tests is described in Section 5.4. As they are calculated with same WAsP sub-model, the orographic turn surrogate model investigations were informed by the results from the orographic speedup tests. For all the tests detailed here, the DNN section of the Grid NN was formed of 500 neurons arranged as 10 layers of 50 neurons, with batch normalisation followed by ReLU activation after each layer. The Grid NN models described here all output a single point value of orographic turn.

6.2.1 Sub-Grid Size

The first step in the Grid NN approach was to determine an appropriate sub-grid size to apply to the input terrain maps. Due to time constraints, it was not possible to test every combination of sub-grid sizes for each input map (e.g. sub-grid size of 3km square for the elevation map, and 1km square for the roughness map in the same model), so for each run the same sub-grid size is applied to every input variable.

Figure 6.1 shows the results of sub-grid size tests, using data from a single site (Site 1) at 10m AGL, and with inputs of min-max normalised elevation, elevation gradient parallel to the wind direction, and roughness. Figure 6.2 shows the results of similar tests for Site 2 at 100m AGL. Table 6.1 presents the MAE and error standard deviation for the tests shown. In both figures, the single point grid shows no ability to learn transformations of the input maps, and the largest (3.2km) sub-grid results lack definition. For the 10m AGL tests (Figure 6.1), the optimum sub-grid size looks to be around 1.2km or 1.6km square but both have similar error metrics, making it difficult to distinguish between these results. For the 100m AGL tests again the orographic turn predictions using 1.2km and 1.6km square sub-grids are the most promising, from Figure 6.2. The performance metrics of all tests are given in Table B.7.

Run ID	**Height AGL (m)**	**Site**	**Sub-Grid Side Length (km)**	**MAE**	**Error Std. Dev.**
OT-G-1	10	1	Single point	1.58	2.29
OT-G-2	10	1	0.8	1.27	1.77
OT-G-3	10	1	1.2	**1.19**	**1.66**
OT-G-4	10	1	1.6	1.36	1.78
OT-G-5	10	1	3.2	1.21	1.78
OT-G-16	100	2	Single point	0.506	0.734
OT-G-17	100	2	0.8	0.458	0.597
OT-G-18	100	2	1.2	**0.385**	**0.545**
OT-G-19	100	2	1.6	0.430	0.598
OT-G-20	100	2	3.2	0.443	0.586

Table 6.1: Performance metrics for orographic turn Grid NN sub-grid size investigations in °; minimum values for each set of tests are highlighted.

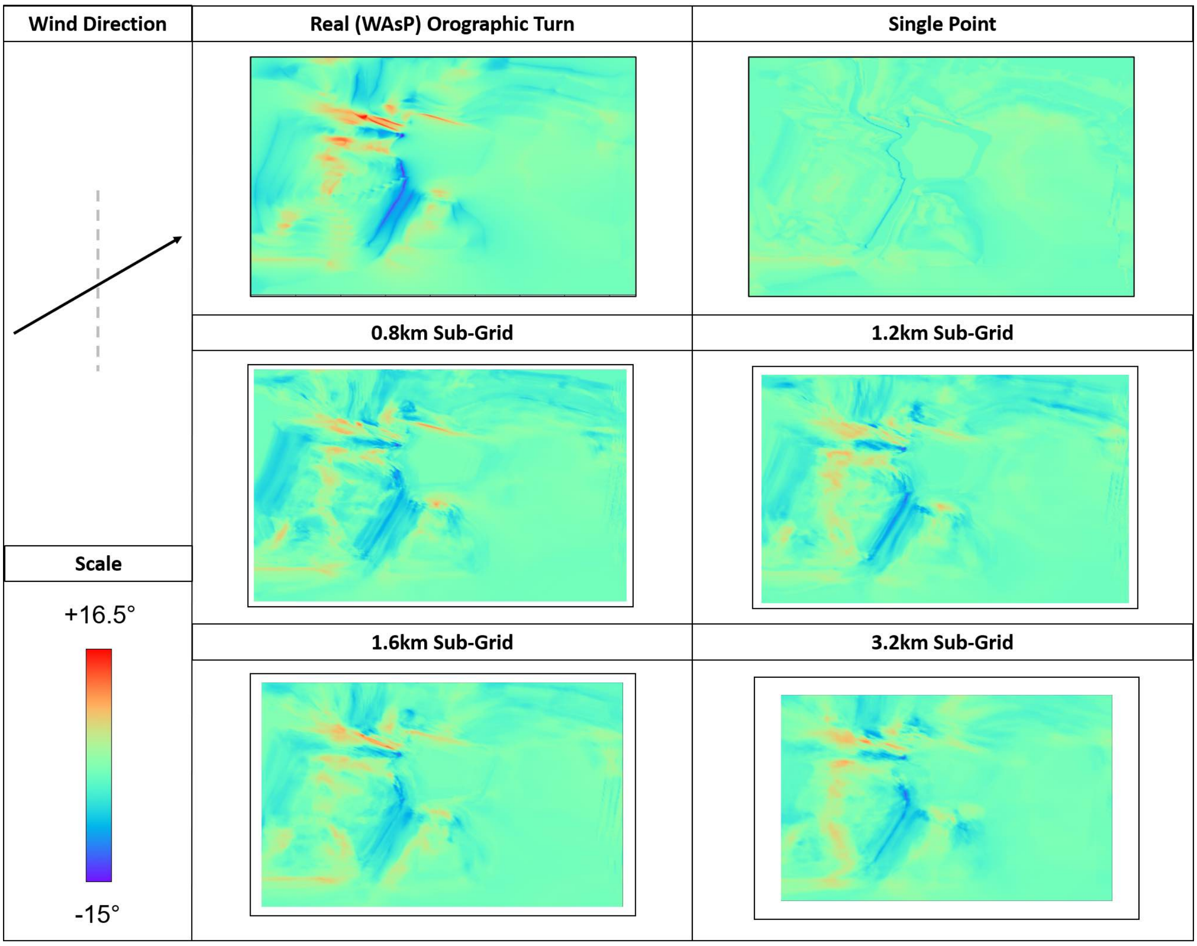


FIGURE 6.1. Comparison of the WAsP and Grid NN predicted orographic turn values for Site 1, sector 9 at 10m AGL, for the sub-grid sizes of (clockwise from top right): single point; 1.2km square; 3.2km square; 1.6km square; 0.8km square.

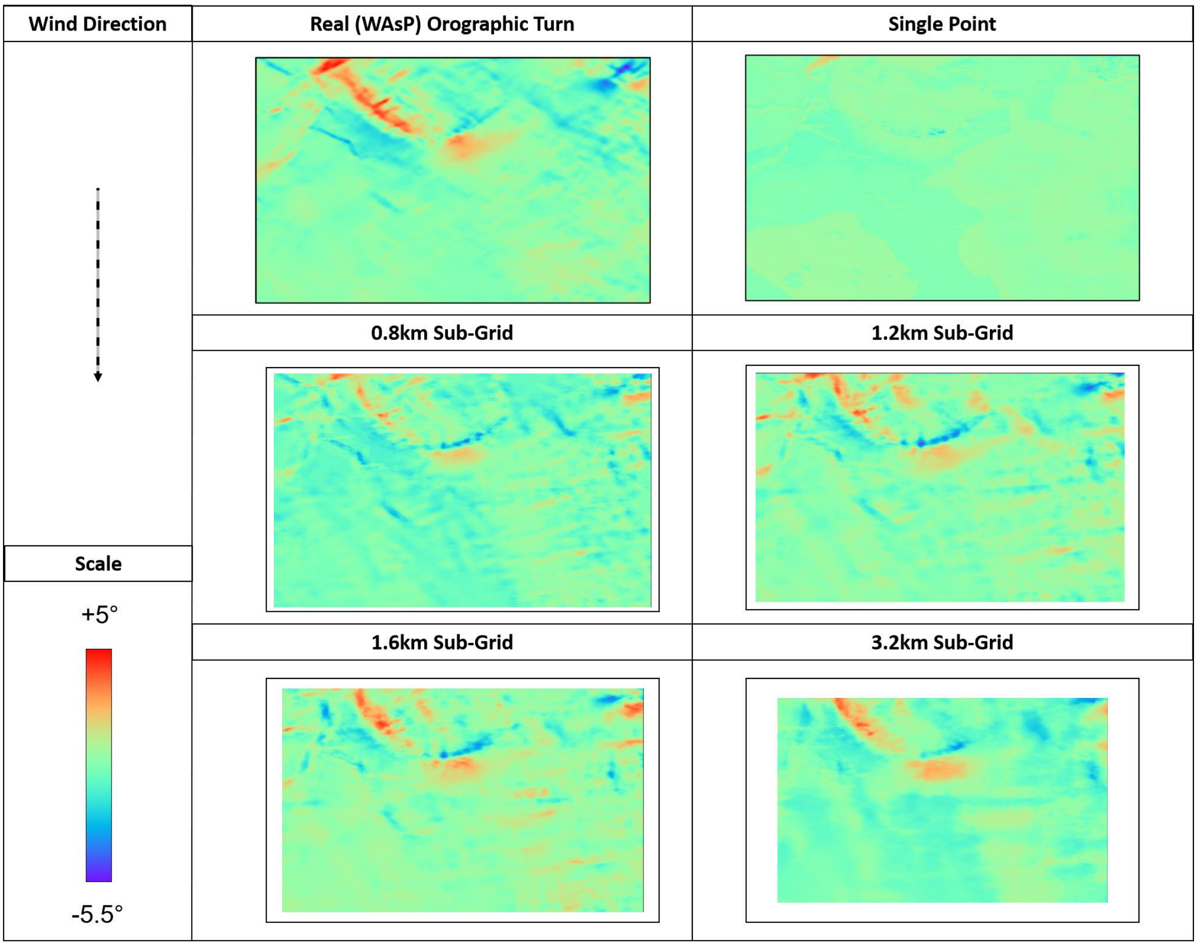


FIGURE 6.2. Comparison of the WAsP and Grid NN predicted orographic turn values for Site 2, sector 1 at 100m AGL, for the sub-grid sizes of (clockwise from top right): single point; 1.2km square; 3.2km square; 1.6km square; 0.8km square.

6.2.2 Inputs

Since the sub-grid sizes of 1.2km and 1.6km square seemed to give similar performance from the plots of the results and the error metrics, both of these configurations were then tested using input data from 11 sites (Sites 1-11 in Appendix A). An additional test was also carried out which included the elevation gradient perpendicular to the wind direction as an input; however the available computing power meant that runs including this additional input could only use data from seven sites for training, rather than 11 sites. At 10m AGL, a comparison of the results for Site 1 and Site 2 for the two grid sizes and the additional input variable are shown in Figure 6.3. The equivalent tests for the same two sites at 100m AGL are given in Figure 6.4. From these results and the performance metrics in Table 6.2, 1.2km and 1.6km sub-grid sizes are comparable in accuracy. Providing the perpendicular elevation gradient as an input gives worse error metrics (e.g. an increase in MAE from 0.993° to 1.48° at 10m AGL). For consistency with the orographic speedup surrogate models in Chapter 5, and to allow for training with the maximum number of sites possible, the 1.2km sub-grid without the perpendicular elevation gradient as input was selected for further testing.

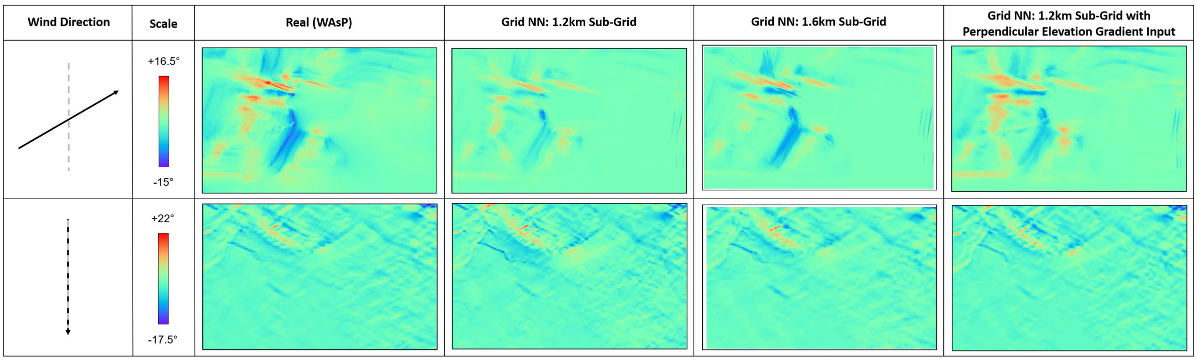


FIGURE 6.3. Comparison of the WAsP and Grid NN predicted orographic turn values at 10m AGL for Sites 1 (top) and 2 (bottom), showing (left-right): WAsP; sub-grid size of 1.2km square; sub-grid size of 1.6km square; sub-grid size of 1.2km square including perpendicular elevation gradient as an input.

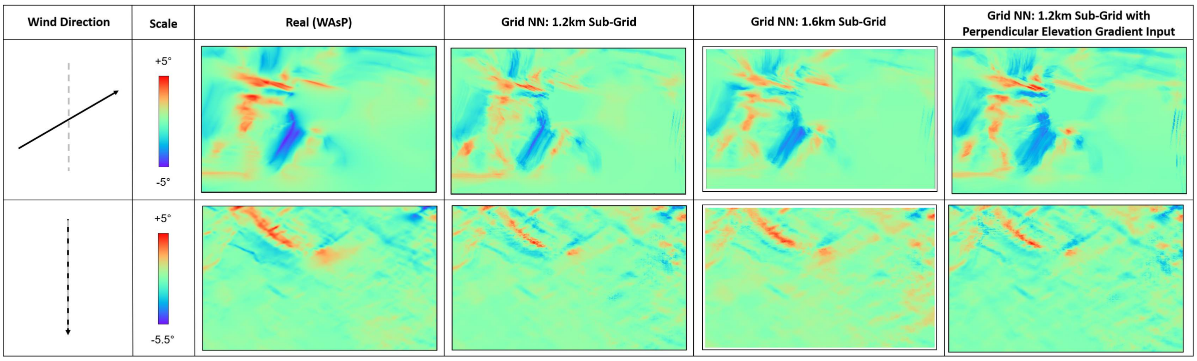


FIGURE 6.4. Comparison of the WAsP and Grid NN predicted orographic turn values at 100m AGL for Sites 1 (top) and 2 (bottom), showing (left-right): WAsP; sub-grid size of 1.2km square; sub-grid size of 1.6km square; sub-grid size of 1.2km square including perpendicular elevation gradient as an input.

Run ID	Height AGL (m)	Elevation Gradient @ 90° Input?	Sub-Grid Side Length (km)	MAE	Error Std. Dev.
OT-G-21	10	No	1.2	0.993	**2.88**
OT-G-22	10	No	1.6	**0.955**	3.03
OT-G-23	10	Yes	1.2	1.48	3.44
OT-G-26	100	No	1.2	**0.381**	**0.603**
OT-G-27	100	No	1.6	0.406	0.616
OT-G-28	100	Yes	1.2	0.483	0.715

Table 6.2: Performance metrics for orographic turn Grid NN input investigations in °; minimum values for each set of tests are highlighted.

6.2.3 Outputs

For the orographic speedup Grid NN models investigated in Section 5.4, the model outputs (i.e. speedup factors) range between 0.0 and 4.0. However, the orographic turn values calculated by both WAsP and the Grid NN models tested here vary between ±180°, a far larger numerical range. To investigate whether the magnitude of the output affected the accuracy of the model predictions, a brief sensitivity study was undertaken where the orographic turn values were either converted to radians, or normalised to a scale of ±1.0, before being used to train the Grid NN models. These tests used a sub-grid size of 1.2km square, did not include the perpendicular elevation gradient as an input, and kept all other conditions the same as for Section 6.2.2.

The results of these tests are given for Sites 1 and 2 in Figure 6.5 (10m AGL) and Figure 6.6 (100m AGL), and the performance metrics are in Table 6.3. The plots and error metrics have been post-processed such that the errors and orographic turn plots are in °. From these results there does not seem to be much of a case for converting the orographic turn to radians, or normalising to a range of ±1.0. There is no consistent improvement when the turn is converted to radians, and at 10m AGL the test with normalised orographic turn output collapsed (Figure 6.5), despite having performance metrics that are not significantly worse than similar runs (Table B.8).

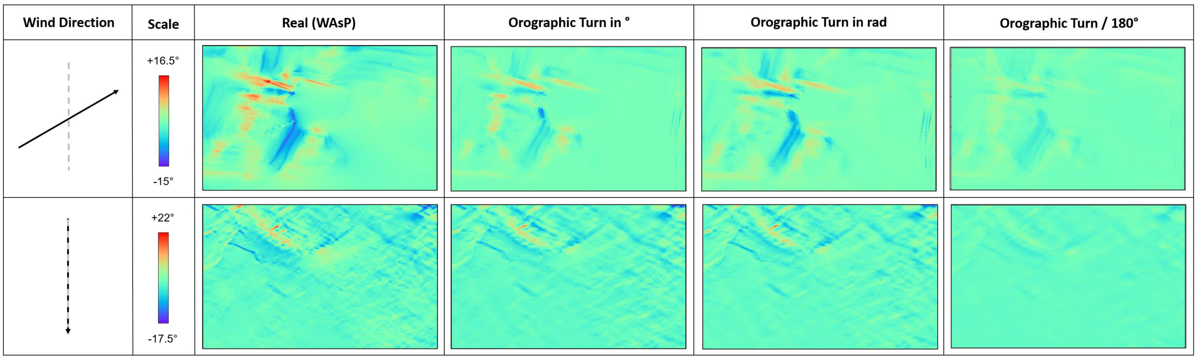


FIGURE 6.5. Comparison of the WAsP and Grid NN predicted orographic turn values at 10m AGL for Sites 1 (top) and 2 (bottom), with orographic turn predicted in (left-right): °; radians; orographic turn / 180 (i.e. normalised to between ±1). The plots are all converted to ° in post-processing.

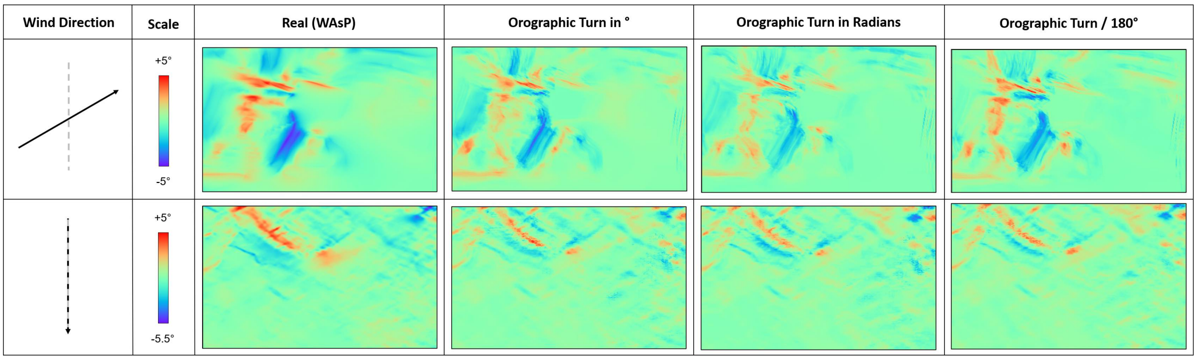


FIGURE 6.6. Comparison of the WAsP and Grid NN predicted orographic turn values at 100m AGL for Sites 1 (top) and 2 (bottom), with orographic turn predicted in (left-right): °; radians; orographic turn / 180 (i.e. normalised to between ±1). The plots are all converted to ° in post-processing.

Run ID	Height AGL (m)	Unit	MAE	Error Std. Dev.
OT-G-21	10	°	0.993	2.88
OT-G-24	10	rad	**0.968**	**2.82**
OT-G-25	10	-	1.15	3.17
OT-G-26	100	°	0.381	**0.603**
OT-G-29	100	rad	**0.372**	0.624
OT-G-30	100	-	0.401	0.619

Table 6.3: Performance metrics for orographic turn Grid NN output investigations in °; minimum values for each set of tests are highlighted.

6.2.4 Final Models

As outlined in Section 5.4.5, although the tests described above were all carried out with a consistent pattern of sub-grid points as inputs, some of these sub-grid points were found to be different position than intended. This issue was fixed and the final orographic turn models at 10m and 100m AGL were re-run. Both of these models had:

- Inputs:
  - min-max normalised elevation, elevation gradient parallel to the wind direction, and roughness;
  - normalised wind direction (rather than rotating the input and output grids);
  - input sub-grid size of 1.2km square;
- Outputs:
  - single point output of orographic turn in °;
  - no data balancing;
- Deep Neural Network:
  - 10 hidden layers of 50 neurons each, with batch normalisation and ReLU activation after each layer;
- Training / Validation:
  - training and validation data from 11 sites (Sites 1-11 in Appendix A) with 12 direction sectors each, where nine sectors were used for training, two for validation and one was held out for inference;
  - training of between 50 and 60 epochs, stopping at the best epoch between these limits (with patience of three);
  - training with an Adam optimiser method, and MSE loss.

Overfitting is potentially a problem in these runs as the loss curves (see Figure 6.7 for the losses at 10m AGL) show more improvement in the training data than for the validation data. The step changes in the orographic turn at ±180° (discussed further in Section 6.2.4.1) could be a factor in the large validation losses, as these loss patterns are not seen in either the orographic speedup (Chapter 5) or roughness speedup loss curves (Chapter 7). However, this could be investigated in greater depth in the future. For ease of comparison, the table of performance metrics is presented here for the final Grid NN models at both heights AGL, for both validation and inference data.

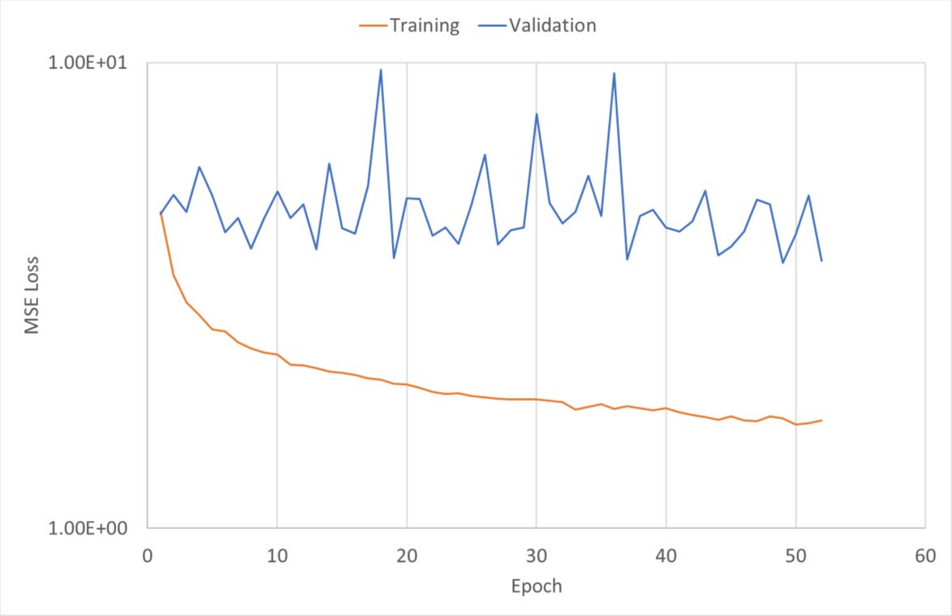


FIGURE 6.7. Loss curves for final Grid NN for orographic turn at 10m AGL.

Run ID	Height AGL (m)	Validation / Inference	MAE	Error Std. Dev.
OT-G-31	10	Validation	0.901	2.66
OT-G-31	10	Inference (New Sites)	1.39	3.11
OT-G-32	100	Validation	0.397	0.635
OT-G-32	100	Inference (New Sites)	0.446	0.786

Table 6.4: Performance metrics for the final orographic turn Grid NN models in °.

6.2.4.1 Validation

Figure 6.8 shows the orographic turn predictions for Sites 1-4 from the final Grid NN model at 10m AGL. The outputs from this model have good accuracy in areas where the shape of the turn follows the shape of an input variable closely (e.g. the valleys in Site 3), but it also performs well over flat areas where there is little or no turn, which are present in all of the sites. The scales of the model-predicted turn are close to those calculated by WAsP, with the error maps showing mostly small differences (as in Figure 6.9). Most of the large differences occur at points of sharp changes in the turn, e.g. the edges of valleys in Site 3, and (similar to the orographic speedup final model) this model has failed to recognise areas of turn in the "shadow" of the hill in Site 2, which can be clearly seen in the error map. The model performs quite poorly on Site 1, where the effect of the large hill on the orographic turn seems to be beyond the capability of the Grid NN model. For this site, with the wind coming from 240°, the scatter plot in Figure 6.10 shows that there is reasonable agreement between the real and predicted outputs for small turn values, but that the larger positive and negative turn values are not predicted well. The Mean Absolute Error over all sites is less than 1°.

The maximum and minimum errors at 10m AGL, which are both large outlier values, occur for a single site and direction, shown in Figure 6.11. This site contains a winding valley feature, for which WAsP calculates orographic turn values that change rapidly from +180° to -180°. For the Grid NN model, such sudden changes in output over a smooth, small change in the input (given that the elevation is relatively constant within the valley) may be very difficult to learn, hence the large errors here. There is also a question on whether it would be physically plausible to turn the incoming wind by 180° in a valley, as calculated by WAsP.

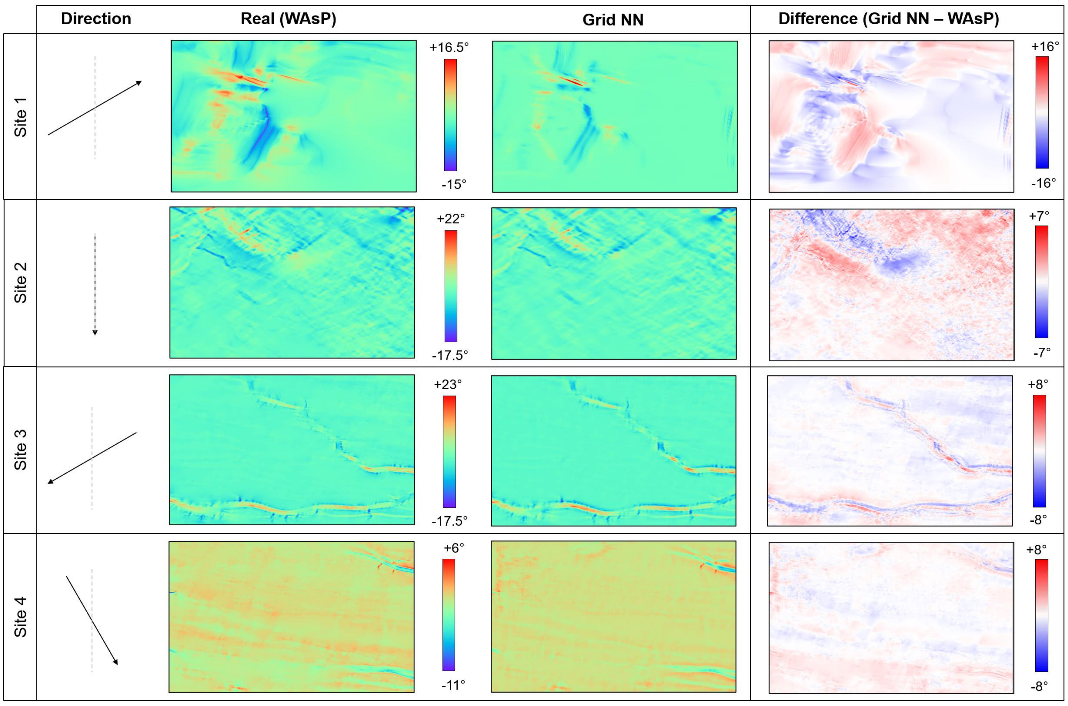


FIGURE 6.8. Final Grid NN model for predicting orographic turn (in °) at 10m AGL for (top-bottom) Sites 1, 2, 3 and 4, showing (left-right): WAsP calculated turn; Grid NN predicted turn; the difference (Grid NN - WAsP).

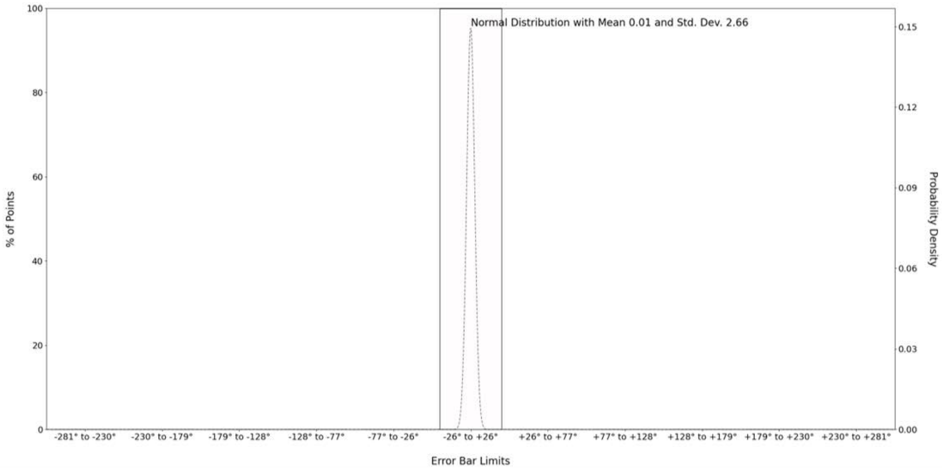


FIGURE 6.9. Bar chart of the errors (Grid NN - WAsP) in orographic turn over the validation data using the final Grid NN model at 10m AGL.

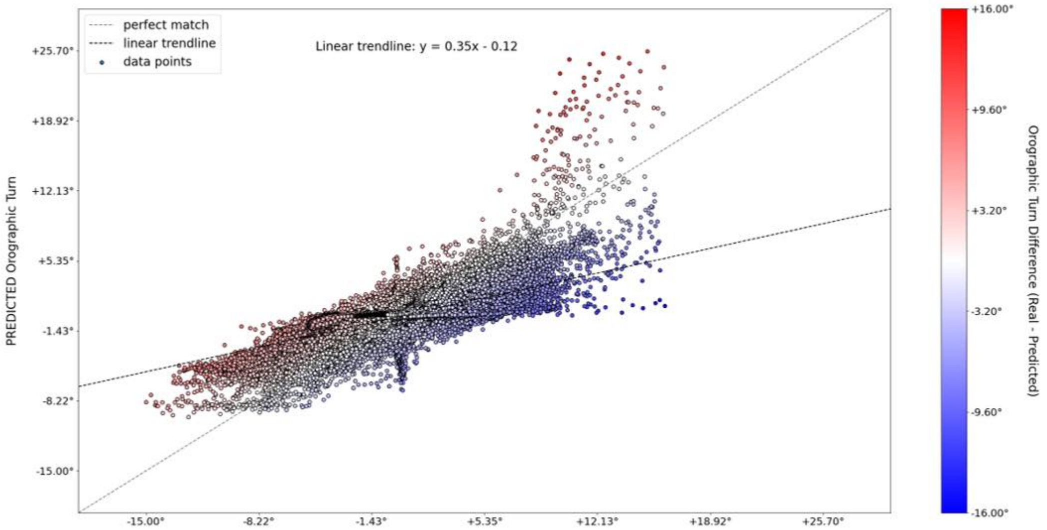


FIGURE 6.10. Scatter graph of the orographic turn at each grid point predicted by the final Grid NN at 10m AGL compared to the WAsP calculated values, for Site 1 sector 9 (wind from 240°).

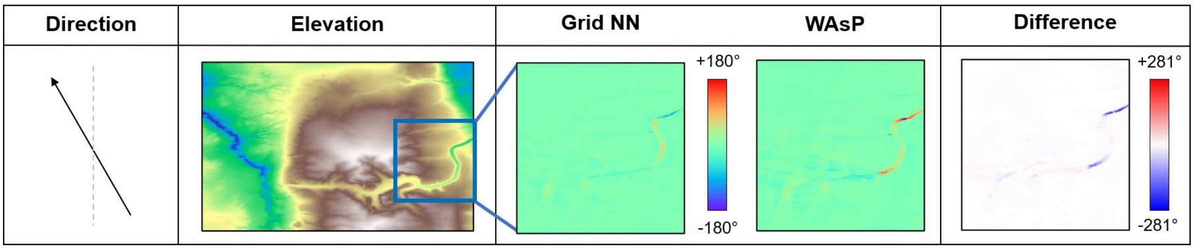


FIGURE 6.11. Orographic turn output from the final Grid NN model at 10m AGL, for Site 5, sector 6; the detail shown is of a valley with high orographic turn values.

The final Grid NN model at 100m AGL has the same setup as that at 10m AGL, as they emulate data from the same WAsP potential flow model (discussed in Section 5.4.5). The magnitude of the orographic turn decreases with height, as the influence of the terrain orography lessens; this leads to much smaller error values for 100m AGL than 10m AGL (Table 6.4). In the Grid NN predictions of the turn at 100m AGL for Site 4 (Figure 6.12), the roughness map input is visible in the model output (as was the case for orographic speedup in Figure 5.26). This seems to be more likely to occur for flat sites, and could have been exacerbated by the small orographic turn values at 100m AGL. The Grid NN model does not capture the largest turn values in Sites 1 and 2, but for Site 3 where the areas of high turn match the terrain features the model performs well. Figure 6.13 shows that over 90% of the errors in this model are less than 1°.

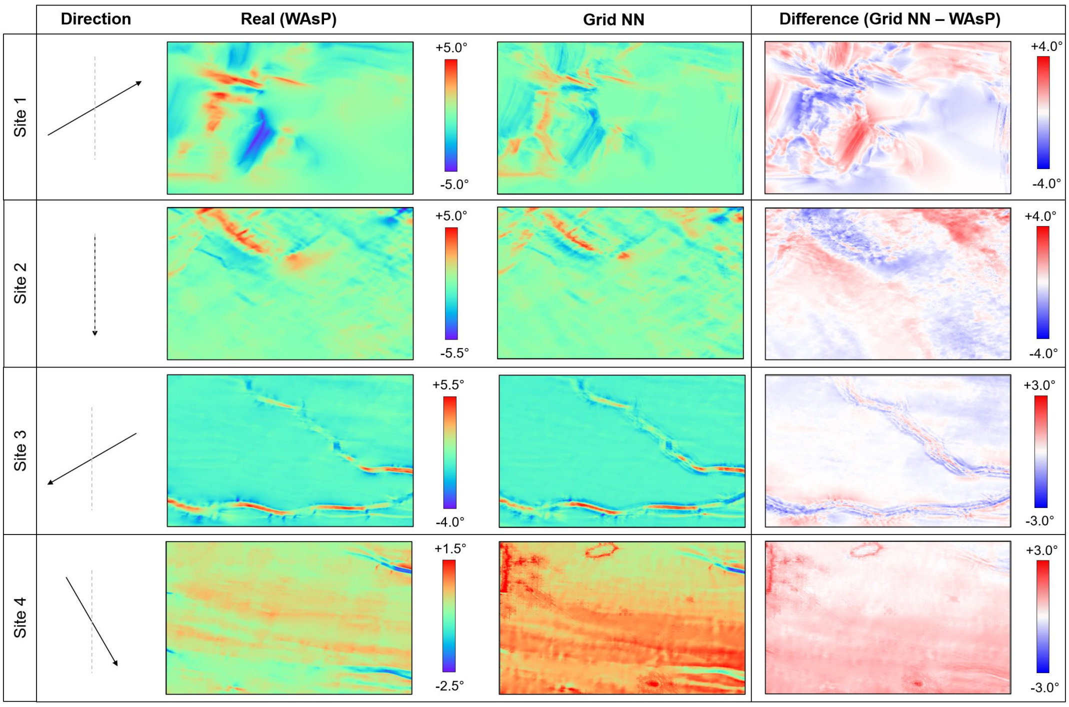


FIGURE 6.12. Final Grid NN model for predicting orographic turn (in °) at 100m AGL, showing (left-right) Grid NN predicted turn, WAsP calculated turn and the difference (Grid NN - WAsP) for (top-bottom) Sites 1, 2, 3 and 4.

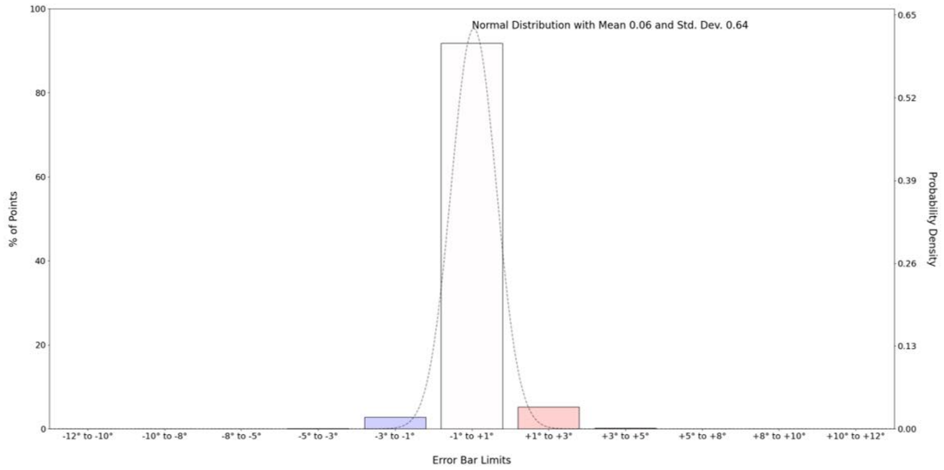


FIGURE 6.13. Bar chart of the errors (Grid NN - WAsP) in orographic turn over the validation data using the final Grid NN model at 100m AGL.

6.2.4.2 Inference (Testing)

For the reasons set out in Section 5.4.5.2, three new sites were used for inference on the final orographic turn Grid NN models at 10m and 100m AGL, with the predictions displayed in Figures 6.14 and 6.17 respectively. At 10m AGL, there are some small areas of very high turn calculated by WAsP (see Site 12 and Figure 6.15), which resulted in some very high error Grid NN point predictions despite the overall error being small. At 100m AGL the orographic turn predictions for Site 12 seem better, but any significant areas of high turn (such as the tops of ridges, or valleys) are not captured by the Grid NN, leading to the conclusion that the model has learned to apply high turn values primarily at steep slopes. The roughness in Site 14 incorrectly dominated the orographic turn predictions at 10m AGL, but this has improved slightly at 100m AGL. The turn values in the valley in Site 13 are predicted well at both heights. The full performance metrics are detailed in Table B.9 and the overall errors are shown in Figures 6.16 and 6.18. The MAE values are 1.4° at 10m AGL and 0.45° at 100m AGL for the inference sites. This compares to MAE values of 0.90° and 0.40° at 10m and 100m AGL respectively over the validation data. Potential reasons for the large increase in error at 10m AGL could be the high turn areas at 10m AGL as discussed above.

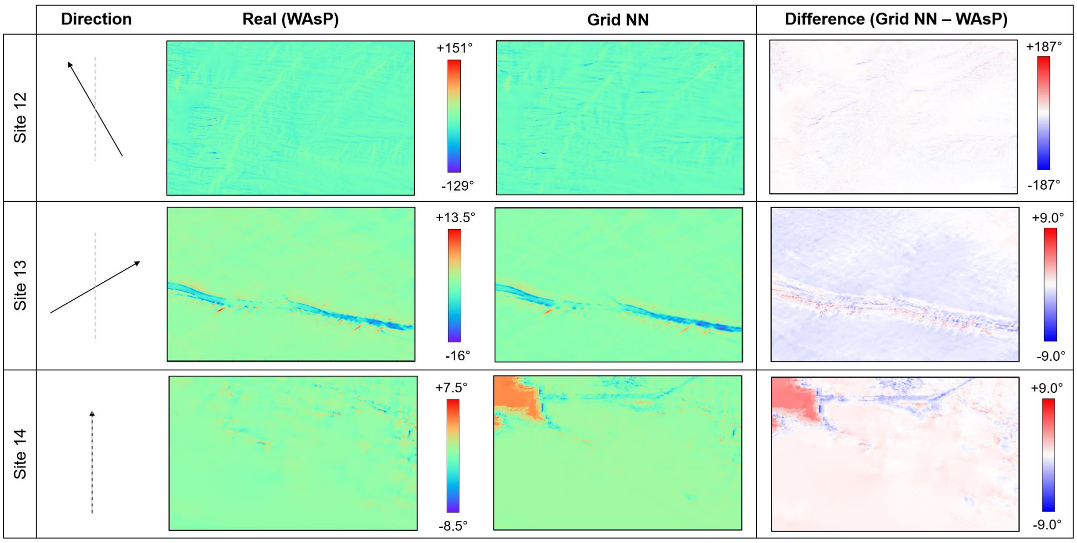


FIGURE 6.14. Final Grid NN model for predicting orographic turn (in °) at 10m AGL for (top-bottom) Sites 12, 13 and 14, showing (left-right): WAsP calculated turn; Grid NN predicted turn; the difference (Grid NN - WAsP).

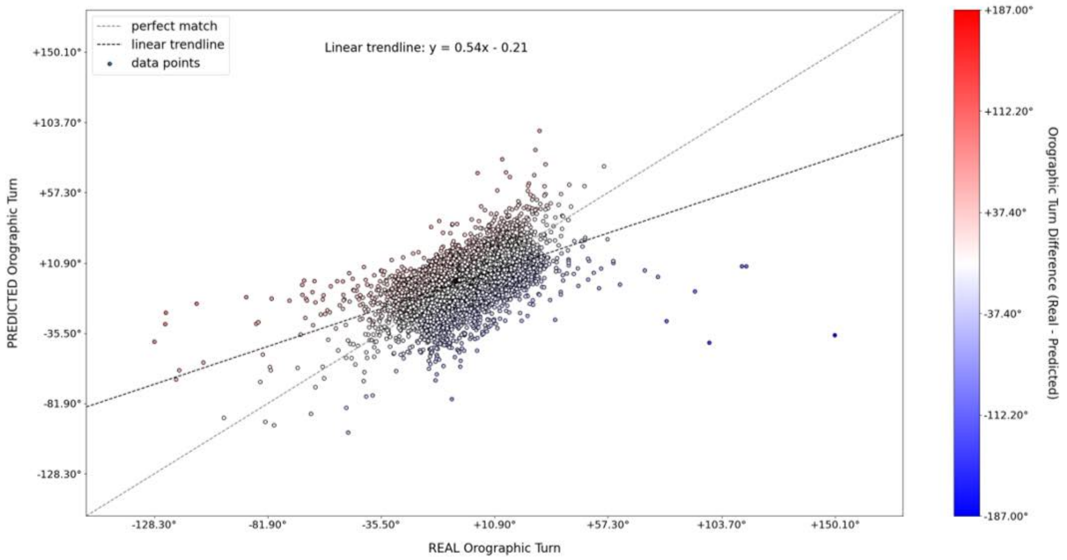


FIGURE 6.15. Scatter graph of the orographic turn at each grid point predicted by the final Grid NN at 10m AGL compared to the WAsP calculated values, for Site 12 sector 6 (wind from 150°).

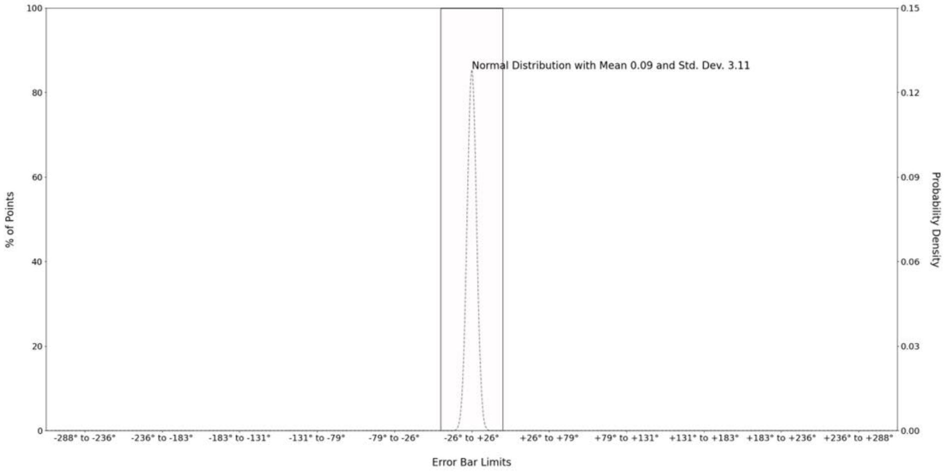


FIGURE 6.16. Bar chart of the errors (Grid NN - WAsP) in orographic turn over three new sites using the final Grid NN model at 10m AGL.

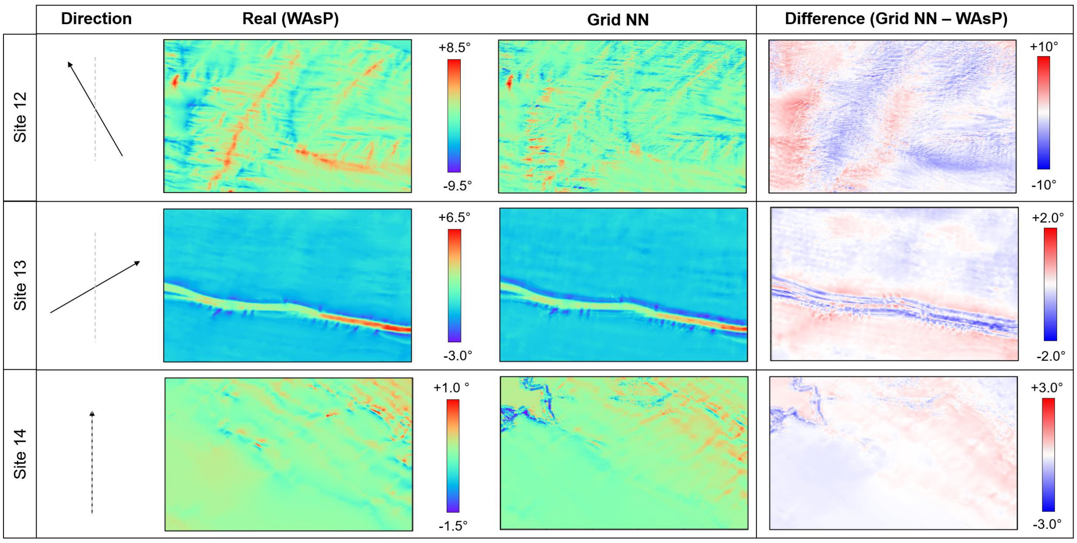


FIGURE 6.17. Final Grid NN model for predicting orographic turn (in °) at 100m AGL for (top-bottom) Sites 12, 13 and 14, showing (left-right): WAsP calculated turn; Grid NN predicted turn; the difference (Grid NN - WAsP).

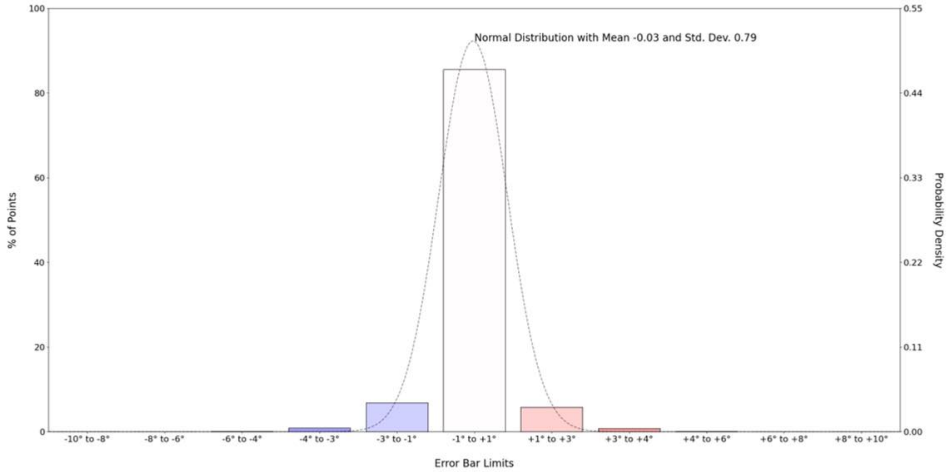


FIGURE 6.18. Bar chart of the errors (Grid NN - WAsP) in orographic turn over three new sites using the final Grid NN model at 100m AGL.

6.3 Conclusions

The Grid Neural Network approach has proven reasonably successful as a surrogate model for orographic turn, using a sub-grid size of 1.2km square, at both 10m and 100m AGL. Figure 6.19 displays examples of a site and direction at each height for which the Grid NN models perform well. The effects of certain complex terrain features are captured, such as valleys where the turn values do not change rapidly. However, these models could benefit from training with more sites, and possibly from different combinations of input variables. The poor performance both of the final models demonstrated on Site 1 might be due to the scale of the hill in this terrain being larger than the sub-grid size, as the model does seem able to give better predictions of the effect of the smaller hill in Site 2, and the valleys in Site 3. This would suggest that adding input points further from the prediction point may help, despite larger sub-grids having proven unsuccessful.

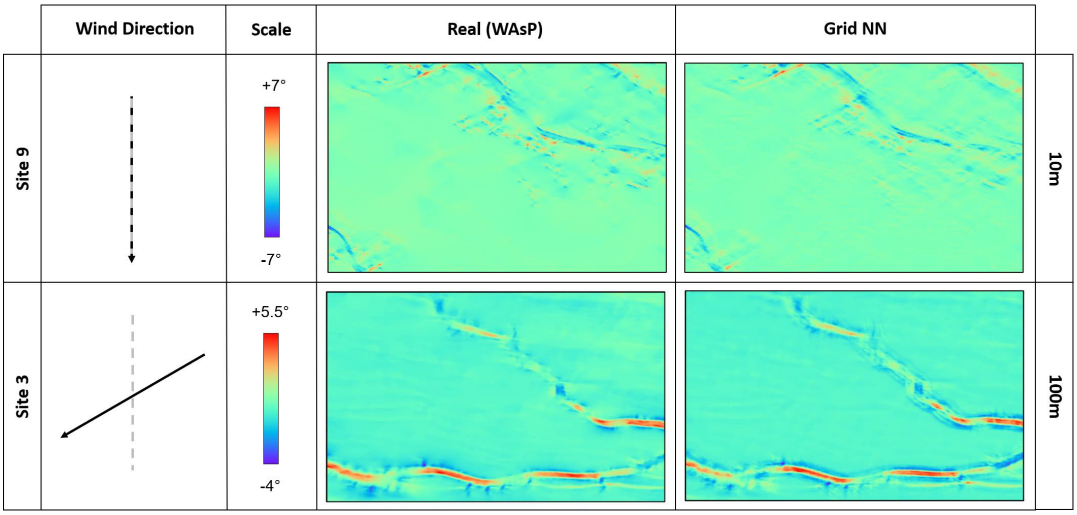


FIGURE 6.19. Examples of two sites and directions for which the final Grid NN models at (top) 10m AGL and (bottom) 100m AGL perform well. Shows the WAsP (left) and Grid NN (right) predicted orographic turn values.

Having created surrogate models for both components of the WAsP orography sub-model, roughness speedup was the final variable to be investigated, as detailed in the next chapter.

Chapter 7

Roughness Speedup Surrogate Models

The WAsP roughness speedup sub-model has a number of fundamental differences when compared to the orographic speedup sub-model. The changes in wind speed from terrain roughness are due to the growth of an Internal Boundary Layer at a roughness transition, which alters the flow downstream of the transition to an extent dependent on the roughness values on either side of the transition. The roughness speedup therefore changes significantly with height, as the new flow regime and turbulence introduced at a roughness transition permeates vertically upwards with downstream distance, meaning that flow at 100m AGL will be affected by the same roughness transition much further downstream than at 10m AGL.

7.1 Inputs

WAsP uses roughness, meso-roughness and roughness changes to determine the roughness speedup across a terrain (Section 2.1.3), hence they are all likely to be necessary inputs to a surrogate model. Roughness values are between 0.0 and 1.5, with 0.0 used for water (lakes, sea, etc) and values >1 used for tall forests. For the sites used in this work, most roughness values were between 0.0 and 0.5, with only one site having roughness values above 0.5. The meso-roughness has similar values to the roughness over a terrain, but since it is calculated per sector and is the effective roughness experienced by the geostrophic wind, it has "roughness persistence" features downstream of roughness changes. The roughness changes over a terrain are integer values, and are also dependent on the wind direction; the maximum number of roughness changes permitted per sector is 10 [54]. An example of the roughness, meso-roughness and roughness changes for a single site and direction are shown in Figure 2.1.

These variables have significantly different scales, with the scale of roughness changes being around 10 times larger than the roughness and meso-roughness values. In order to ensure that no single type of input would dominate the predictions, the roughness and meso-roughness values were transformed in some tests (where specified) to have similar ranges to the roughness changes. From the equations for roughness speedup specified in the European Wind Atlas [8], the basis for the WAsP model, the wind speed is dependent on the natural log of the inverse of the roughness. Hence, to even out the scales of all the input variables, tests were run which included the natural log of the roughness and meso-roughness, and the natural log of the inverse of the roughness and meso roughness. To avoid errors from zero division or taking the logarithm of zero, areas of water which have a roughness value of 0.0 were assigned a roughness value of 2E-4, which is the value that the WAsP 11 manual [54] states is the true roughness length of water, and is a small value in this context.

Since wind resource grids at 100m AGL are likely to be of more use to wind energy practitioners than 10m AGL (as it is closer to the height of wind turbine hubs), and the roughness speedup patterns at 100m AGL bear less resemblance to the terrain features, initial tests used data from 100m AGL.

7.2 Inputs Testing

The first investigation into roughness speedups was to find which inputs were of most importance. Using Grid Neural Networks with one type of input variable each, an optimum grid size was found for each input, with the results shown in Figure 7.1 and metrics given in Table B.14. The inputs used were roughness, meso-roughness and roughness changes, with no transformations applied. However, it is clear that none of these models performed particularly well, and hence no single type of input contains enough information to calculate roughness speedups over terrain. The subsequent runs therefore combine these inputs and their various transformations. These single input variable tests were not repeated for 10m AGL due to the lack of success at 100m AGL.

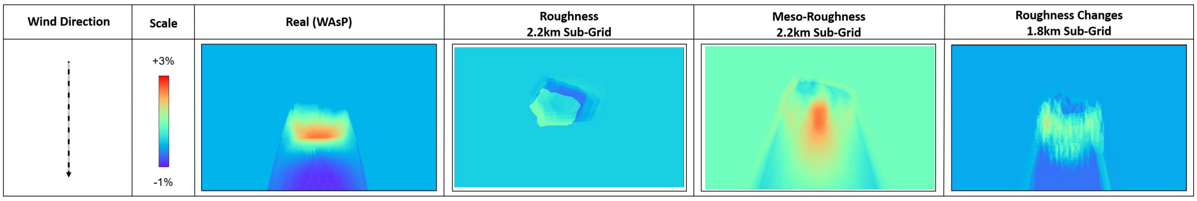


FIGURE 7.1. Best results from Grid NN models with a single input variable type (left-right): roughness, meso-roughness; roughness changes for Site 1 at 100m AGL.

7.3 Convolutional Neural Networks

Roughness speedups, in particular those at large vertical heights, require information from a large surrounding area of terrain - the recommended minimum map size to accurately calculate speedups in the WAsP manual is 20km x 20km [54]. At these sizes, the problem becomes almost an image transformation task, and so Convolutional Neural Networks were again considered as a potential surrogate model architecture. Note that unlike the CNNs tested for orographic speedup (Section 5.3), in this investigation all input and output grids were rotated such that the wind direction was always from 0°.

7.3.1 Initial Tests

Testing of the CNNs for roughness speedups began by using speedup predictions at 100m AGL. As with the orographic speedup CNNs, the first networks tested used comparatively small filters in an autoencoder-style structure. Unfortunately, the majority of these runs "collapsed", giving no useful outputs, for example by predicting a uniform speedup value at every coordinate point. Possible reasons for this could be the relatively small data set available for training, inappropriate filter sizes, or that the input data grids did not encompass a sufficiently large area to encompass sufficient information to calculate roughness speedups at 100m AGL. Few of the initial CNN tests showed any promise, with an example of the results in Figure 7.2 (error metrics in Table B.10).

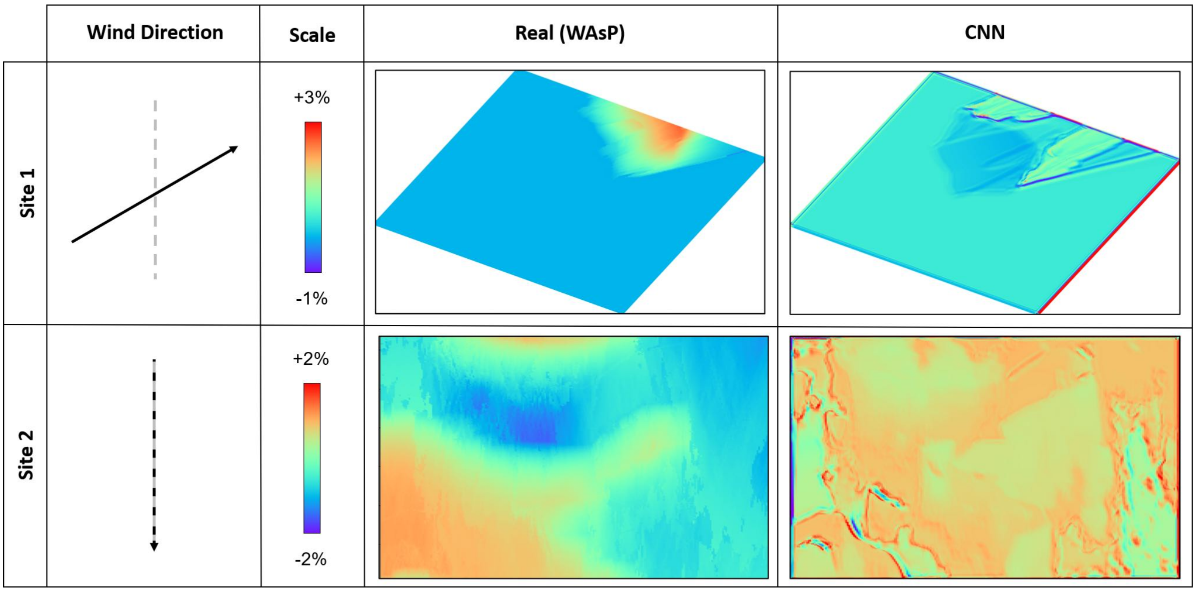


FIGURE 7.2. Results from a CNN for predicting the roughness speedup at 100m AGL, for Site 1 (top) and Site 2 (bottom), with (left-right) WAsP calculated speedup and CNN predicted speedup.

Although the roughness speedup values at 100m AGL may be more useful to industry, they proved to be difficult to model with simple CNNs. However the terrain roughness has a greater influence on the wind flow close to the ground; this can be seen for Site 1 in Figure 2.2, where the scale of the roughness speedup at 10m AGL is between +14% and -7% while at 100m AGL it is between +3% and -1%. For training surrogate wind flow models, the larger magnitudes of roughness speedups at 10m AGL could give more information for the Neural Networks to train on. In addition, the shape of the roughness speedups at 10m AGL is closely correlated to the roughness terrain features; again from Figure 2.1 and Figure 2.2 the direct influence of the roughness, meso-roughness and roughness changes on the roughness speedups is easier to see at 10m AGL than at 100m AGL.

7.3.2 Transfer Learning from 10m to 100m AGL

To give the CNNs to best chance of learning relations between the terrain variables and the roughness speedup, the variables and their transformations were provided as inputs to the CNNs. The full set of inputs was:

- roughness, natural log of roughness, and natural log of inverse roughness;
- meso-roughness, natural log of meso-roughness, and natural log of inverse meso-roughness;
- difference between roughness and meso-roughness;
- roughness changes.

as shown in Figure 7.3 for Site 1 sector 9.

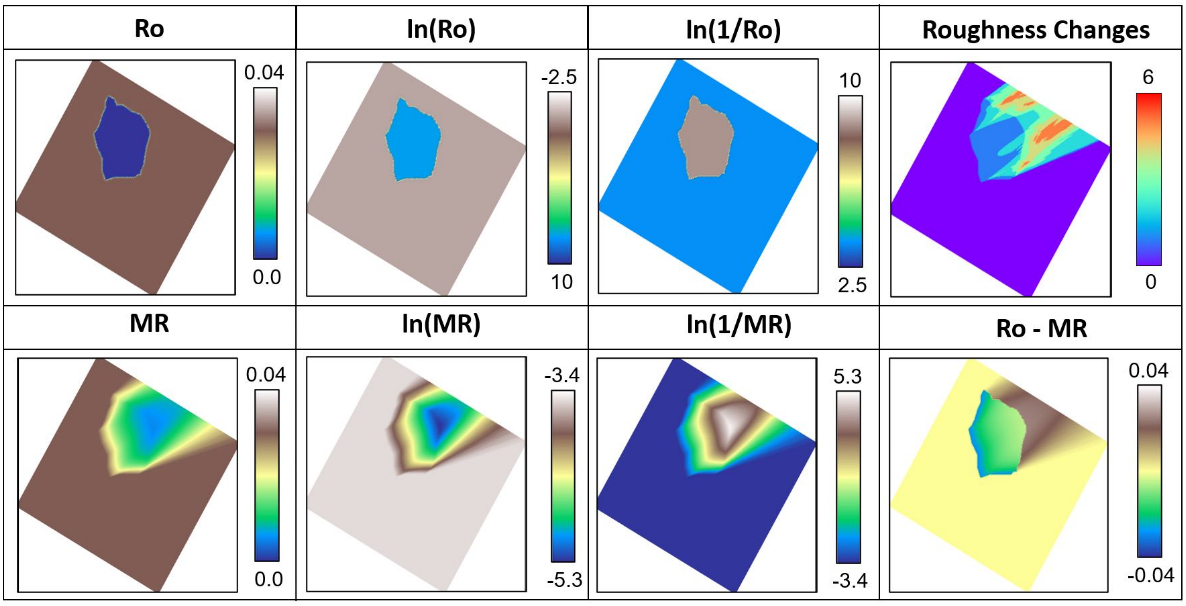


FIGURE 7.3. All inputs provided for Site 1 sector 9 roughness speedup CNNs, where "Ro" is roughness and "MR" is meso-roughness.

As the roughness speedups at 10m AGL contain more information for training an NN, and the speedups at different heights are related, it was thought that transfer learning might achieve good speedup predictions at multiple heights. Transfer learning has been successfully used within the image processing field, where trained models such as ImageNet [59] which have been proven to work, are applied to different purposes (e.g. [69]). For the current problem, a CNN trained on data at 10m AGL could serve as a baseline model, then have additional training using data at e.g. 25m or 100m AGL.

Figure 7.4 shows the results of transfer learning on a baseline run using only data from Site 1, followed by runs with additional training at 25m AGL or 100m AGL. The error statistics for these runs are presented in Table B.11. In the baseline CNN at 10m AGL the model has been unable to completely learn that the area of high speedup is downwind of the grass-to-lake roughness transition. This is exacerbated in the transfer learning models, with the roughness transition emphasised in the CNN predictions, unlike the WAsP calculated speedups. The error limits stay between ±30% for all three runs, which is high compared to the actual speedup values. At 25m and 100m AGL there is evidence (Figure 7.4) that the CNNs have learned to move the high speedup areas further downstream of the roughness transition than at 10m AGL. However it seems likely (judging by their shape and position) that these speedup areas are re-scaled versions of the meso-roughness and roughness changes, rather than new relations learned by the CNN.

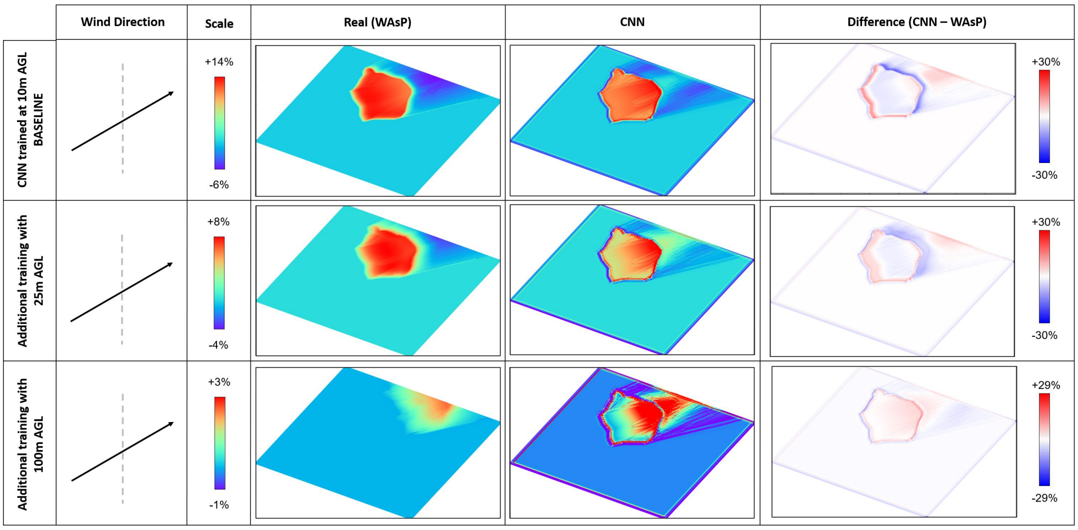


FIGURE 7.4. Transfer Learning of a CNN for roughness speedup for Site 1 only, showing (left-right): WAsP; CNN; difference between them. Starting with a baseline model trained on data from 10m AGL (top), which is then trained with additional data from 25m AGL (middle), or alternately trained with additional data from 100m AGL (bottom). Note that the 10m AGL model was the baseline for both additional heights.

7.3.3 Batch Normalisation

From this point on, the CNNs used training data from multiple sites at 10m AGL for training each model. A reasonably successful CNN architecture was tested in Figure 7.5, where the inclusion of batch normalisation layers makes a significant improvement to the results (see Table B.12 for the error metrics and CNN architecture). The batch normalisation process (detailed in Section 3.6.1) normalised the output from each convolutional layer to have a mean of 0.0 and standard deviation of 1.0. Given the effect that this has on the results in Figure 7.5, it is assumed that the varying scales of the inputs, and the variation between sites, were evened out by the batch normalisation, allowing the different inputs to be combined to give a reasonable approximation of the speedup. As seen in Section 7.3.2, the CNN with batch normalisation has not learned to position the area of high speedup in Site 1 downstream of the roughness transition, and instead predicts it to occur in the same position as the roughness feature (the lake).

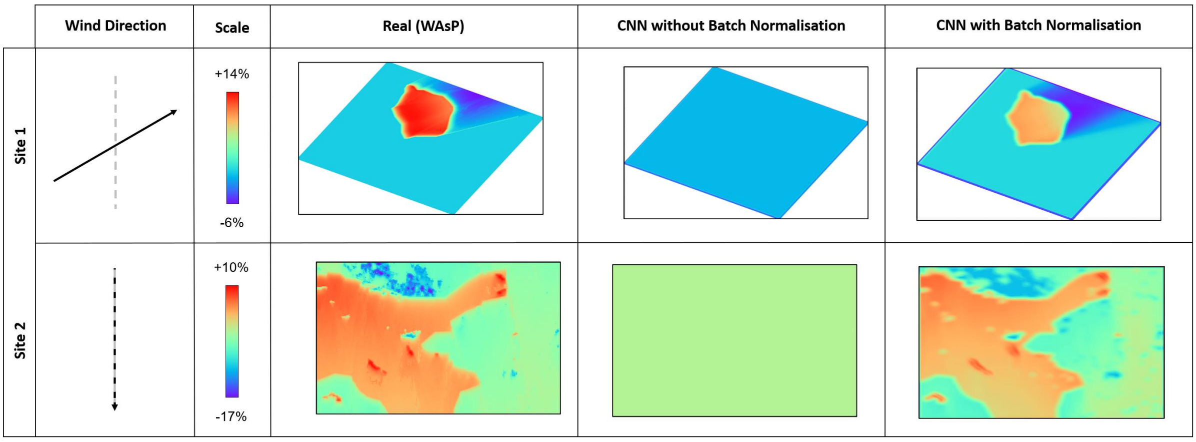


FIGURE 7.5. Effect of adding batch normalisation layers into CNNs for predicting roughness speedup at 10m AGL, for Site 1 (top) and Site 2 (bottom).

7.3.4 Filter Sizes

As detailed in Table B.13, the filter sizes for the CNN shown in Figure 7.5 (with batch normalisation) are small, and so do not take in information from a large input area. To determine whether greater terrain areas were required as inputs to the CNN, a set of tests were carried out with an extra convolutional layer added to the batch normalised model in Figure 7.5. The filter size of this additional initial convolutional layer was larger than any of the following layers, and was changed for each of tests. The results of these runs are given in Figure 7.6 and Table B.13; it can be seen that the CNN predicted speedup maps lose clarity when the additional large filter convolutional layer is included. Edge effects can also be seen when a 51x51 filter is tested, and while the scales of the speedups improve slightly with large filters for Site 1, the same cannot be said of Site 2. Overall, the addition of a large filter convolutional layer did not improve on the baseline CNN.

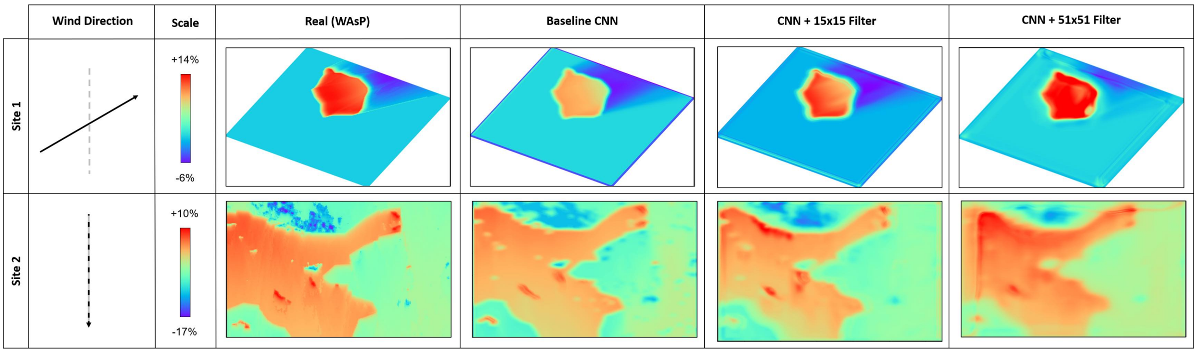


FIGURE 7.6. Effect of adding filters of increasing size to the baseline roughness speedup CNN as the first layer for Site 1 (top) and Site 2 (bottom). Shows (left-right): WAsP; baseline CNN; additional 15x15 filter; additional 51x51 filter. Note: all input grid sizes are 250x250.

7.4 Grid Neural Networks

7.4.1 Input and Output Sub-Grid Size

Having exhausted trials of CNNs, and with the Grid NN code updated to take in multiple input variables (as discussed in Section 5.4.1), the Grid NN approach for roughness speedups was investigated. As with the orographic speedup investigations, the first stage was to find appropriate sub-grid sizes for both the inputs and outputs, using data from a single site per test. The inputs used for each test were:

- natural log of inverse roughness;
- natural log of inverse meso-roughness;
- difference of the natural logs of inverse roughness and inverse meso-roughness;
- number of roughness changes.

These transformations were chosen as the scale of all these inputs is similar, between ±10. The Grid NN parameters were:

- DNN architecture of 1,000 neurons, split into 10 layers of 100 neurons;
- trained for between 5 and 10 epochs;
- Adam optimiser with MSE loss;

This is a similar structure to the orographic speedup models tested in Section 5.4.5, but with more capacity as there are more input features and a complex relationship between the inputs and the outputs, particularly at 100m AGL. Unlike the orographic speedup and turn Grid NN sub-grid size tests in Chapters 5 and 6, both the input and the output sub-grid sizes were investigated here. During the post-processing stage of plotting the validation speedup maps, the predictions from the Grid NN for each coordinate point were averaged. For example, if a Grid NN model were to output the speedup values of a square sub-grid of 3x3 adjacent points (Figure 5.9), then the speedup at every coordinate point in the final output map would be predicted by nine different iterations of the Grid NN, and the value plotted would be their average. The intended effect of this was to smooth out the speedup predictions, which could be especially useful for the roughness speedups at 100m AGL.

Given the results of the sub-grid size tests on orographic speedup and turn, only a small sample of sub-grid sizes was tested:

- 1.2km square;
- 2.2km square;
- 3.2km square.

The output sub-grids tested were:

- single point output;
- 25 points forming a 5x5 square with side length 0.2km;
- 17 points forming a concentric squares pattern with side length approximately half the input sub-grid length.

Figure 7.7 shows the effect of increasing sub-grid size (with single point output) at 10m and 100m AGL. Figure 7.8 shows the difference in using the three output sub-grid methods at 10m AGL (with 1.2km square input sub-grid) and 100m AGL (with 3.2km square input sub-grid). These tests were repeated for two other sites, with comparable levels of prediction accuracy to Site 1 at 10m AGL, but with very poor results at all input and output sub-grid configurations at 100m AGL.

From Figure 7.7 and Table 7.1 the optimal input sub-grid size for 10m AGL is around 1.2km square, as it has the lowest MAE and minimises the "halo" effect seen around the area of high speedup. It is difficult to find an optimum input sub-grid size at 100m AGL from Figure 7.7, but from Table 7.2, 3.2km has the best performance of the three sub-grids tested. This makes sense given the speedups at this height are affected by terrain features further upstream than at 10m AGL, and hence require a larger area of terrain data. There is a pronounced smoothing effect (Figure 7.8) when using multiple output sub-grid points, but when the data from each of the Grid NN output neurons was plotted separately they were almost identical (as would be expected). There is little benefit to learning the same input-output relations multiple times, and so the idea of multiple output sub-grid points was not continued.

These results show clearly the workings of the Grid NN approach, which "re-prints" the input maps at various positions, scaling the values appropriately. This leads to a "halo" effect around the lake in Site 1 at 10m AGL, and at 100m AGL rather than an area of zero speedup, copies of the lake roughness feature are visible (Figure 7.7).

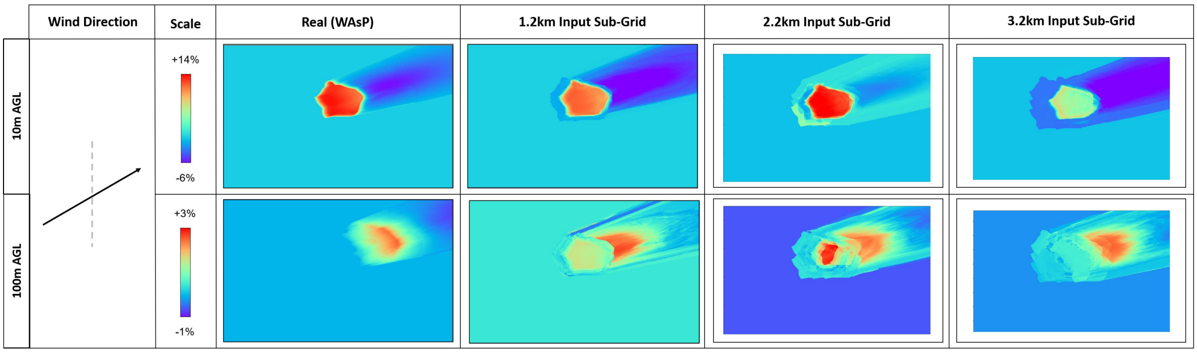


FIGURE 7.7. Input sub-grid size investigations for Grid NN roughness speedup models at 10m (top) and 100m (bottom) AGL for Site 1. All runs used a single point output.

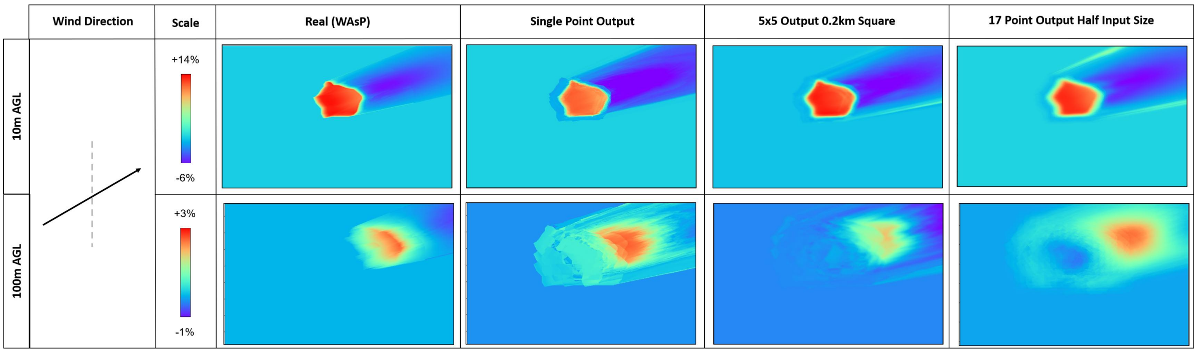


FIGURE 7.8. Output sub-grid size investigations for Grid NN roughness speedup models at 10m (top) and 100m (bottom) AGL for Site 1. At 10m AGL the input sub-grid size is 1.2km square; at 100m AGL the input sub-grid size is 3.2km square.

Run ID	Input Sub-Grid Size (km)	Output Sub-Grid Size (km)	MAE	Error Std. Dev.
RS-G-4	1.2	Single Point	**0.437**	0.781
RS-G-5	2.2	Single Point	0.960	1.35
RS-G-6	3.2	Single Point	1.45	1.71
RS-G-7	1.2	0.2	0.482	**0.681**
RS-G-8	1.2	0.6	0.438	1.09

Table 7.1: Performance metrics for roughness speedup Grid NN input and output sub-grid configuration investigation at 10m AGL, in % speedup; minimum values are highlighted.

Run ID	Input Sub-Grid Size (km)	Output Sub-Grid Size (km)	MAE	Error Std. Dev.
RS-G-9	1.2	Single Point	0.543	0.255
RS-G-10	2.2	Single Point	0.550	0.576
RS-G-11	3.2	Single Point	0.279	0.323
RS-G-12	3.2	0.2	0.256	**0.200**
RS-G-13	3.2	1.6	**0.241**	0.354

Table 7.2: Performance metrics for roughness speedup Grid NN input and output sub-grid configuration investigation at 100m AGL, in % speedup; minimum values are highlighted.

7.4.2 Network Size

A study into the appropriate DNN size was undertaken, focussing on the roughness speedups at 100m AGL. These runs were trained on the data from seven[1] sites (a consistent subset of sites 1-11 in Appendix A), using the input variables described in Section 7.4.1, with a 3.2km square input sub-grid and single point output. The DNN architectures tested were:

- 100 neurons (10 layers of 10 neurons);
- 500 neurons (10 layers of 50 neurons);
- 1,000 neurons (10 layers of 100 neurons).

Note that Site 4 is presented here rather than Site 2, as Site 2 showed uniformly poor results at all network sizes; Site 3 was not one of the seven sites used to train these models. From Figure 7.9, 100 neurons is insufficient to capture the input to output transformations, and there is little overall benefit to using 1,000 neurons rather than 500 neurons. However, Table 7.3 shows that both the MAE and standard deviation of the error decrease when the number of neurons is increased to 1,000; though the decrease in MAE is only 0.3% speedup from 500 to 1,000 neurons. The model with 500 neurons does not match the scale of the increase in speed downstream of the lake in Site 1, but it does give a good approximation to zero speedup ahead of the lake. However with 1,000 neurons, there is an area of decrease in speed ahead of the lake, and a slight loss in

[1] As four input variable maps were used to train the roughness speedup Grid NNs compared to three variables needed to train the orography Grid NNs, only seven sites could be used rather than 11 due to computational constraints.

definition in the "roughness persistence" downstream of the lake. The outputs for Site 4 look approximately the same for 500 and 1,000 neurons. To avoid overfitting caused by having excess model capacity, it was judged best to use Grid NN models with 500 neurons at 100m AGL; 500 neurons also proved to be a suitable network size for runs at 10m AGL, using an input sub-grid size of 1.2km square and single point output.

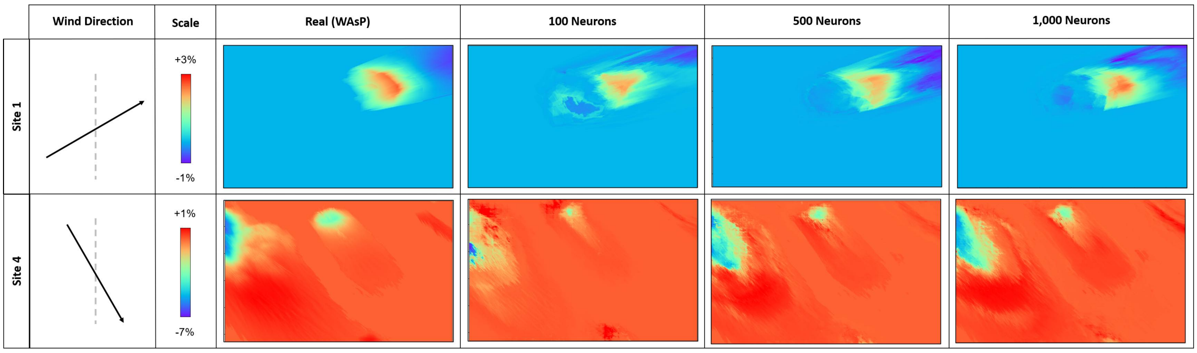


FIGURE 7.9. Network size investigation for Grid NN roughness speedup models at 100m AGL showing (left-right): WAsP; 100 neurons; 500 neurons; 1,000 neurons.

Run ID	Neurons	MAE	Error Std. Dev.
RS-G-14	50	0.418	0.842
RS-G-15	500	0.369	0.844
RS-G-16	1000	**0.339**	**0.776**

Table 7.3: Performance metrics for roughness speedup Grid NN network size tests, in % speedup; minimum values are highlighted.

7.4.3 Overfitting

Examining the loss curves from the two 500 neuron Grid NN models (for 100m and 10m AGL) at this point revealed that both models were likely to be overfitting their training data (see loss curves for the 100m AGL model in Figure 7.10). The results from the 500 neuron model at 10m AGL at this stage looked very promising for the validation data from all seven sites used. By contrast, at 100m AGL there were several validation sites for which the 500 neuron Grid NN produced very poor results; therefore the 100m AGL models were suspected to be more affected by overfitting.

A Neural Network which is overfit to its training data has learned to reproduce the training data outputs, rather than learning to apply generalised input-output relations which can be used on the validation or other new data. To test whether the 500 neuron Grid NN at 100m AGL was overfitting the training data, a sample of training sites and sectors were plotted and compared

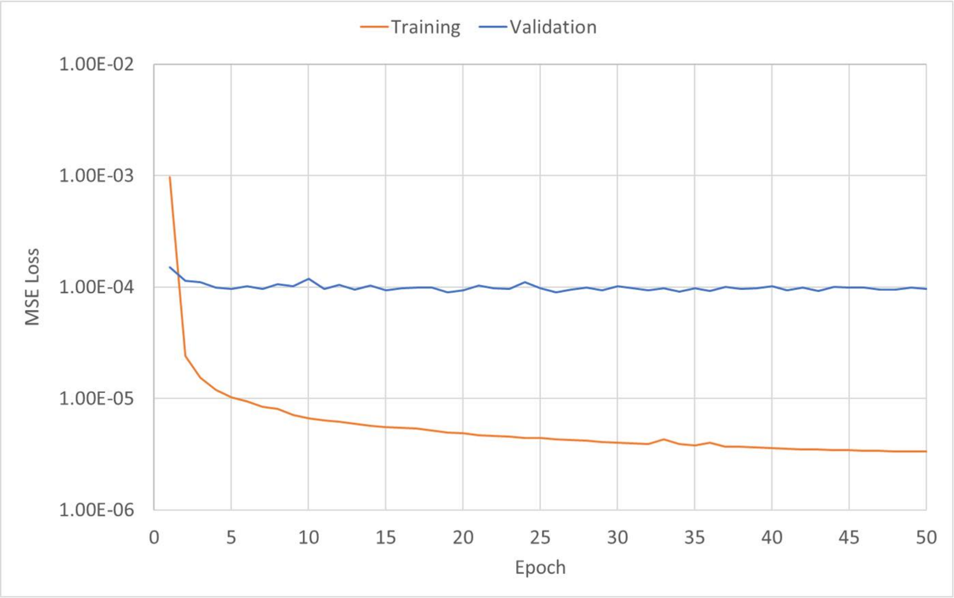


FIGURE 7.10. Loss curves for Grid NN for roughness speedup at 100m AGL, showing potential overfitting.

against the validation sites and sectors, seen in Figure 7.11 for Site 2. This does suggest that the model is overfit, as it is able to produce very good roughness speedup maps for the training data, but cannot predict the validation data with any accuracy. By contrast, this model produced good predictions on both the training and validation data for Site 1 (not shown), so it is not clear which factors affect the overfitting of the model.

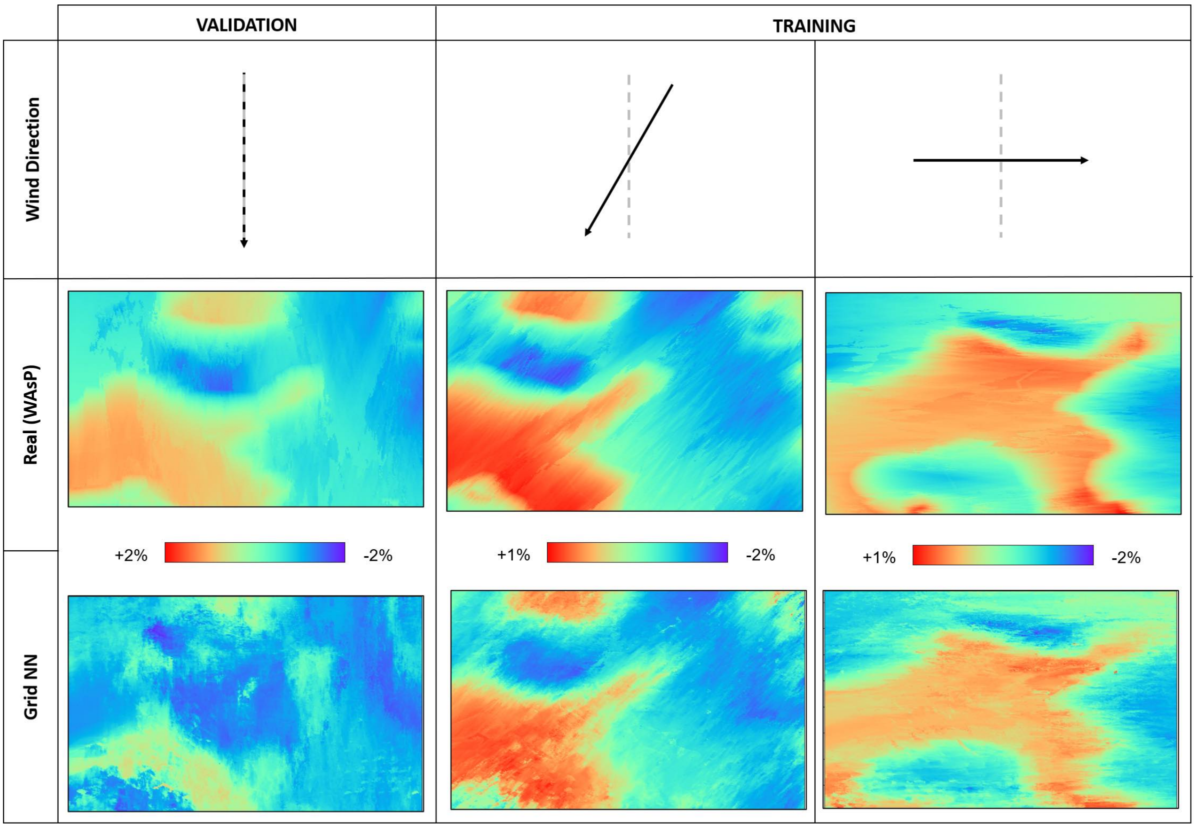


FIGURE 7.11. Comparison of the roughness speedup output from a Grid NN model at 100m AGL for Site 2 (left-right): validation sector 1; training sector 2; training sector 10.

To combat the issue of overfitting in the 100m AGL run, the widely used techniques of dropout (Section 3.6.3) and weight regularisation (Section 3.6.4) were applied to re-runs of this model. For two runs, dropout probabilities of 10% and 25% were applied to all layers (except the output) after batch normalisation. For another two runs, L2 weight regularisation was implemented with decay factors of 0.1 and 0.25. Unfortunately, neither of these methods proved successful (results from one run using each method are shown in Figure 7.12), as the models with weight regularisation collapsed, and the models employing dropout gave significantly worse performance than the baseline run. It is possible that the parameters used for weight regularisation and dropout were non-optimal, but it was not deemed worthwhile to conduct a full investigation of this.

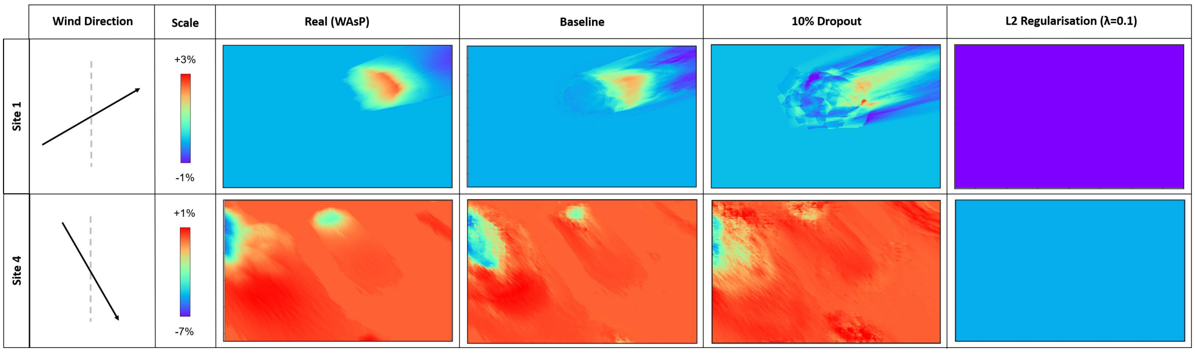


FIGURE 7.12. Application of techniques to prevent overfitting in a Grid NN model at 100m AGL, for Site 1 (top) and Site 4 (bottom), using (left-right) WAsP; no dropout or weight regularisation; dropout with 10% probability; weight regularisation with decay factor 0.1. Note that Site 2 showed uniformly poor output so is not included.

7.4.4 Combining Grid NN and CNN

While the Grid NN approach provided much more promising results than the CNN tests, there were still improvements to be made to the roughness speedup predictions, especially at 100m AGL. The conclusion from the CNNs tested was that the input maps were too dissimilar to the output maps for the CNN filters to learn the input-output transformations with the limited data available. To overcome this, it was considered whether it would be possible to combine both modelling approaches, by using the Grid NN predicted speedups as inputs to a CNN. The Grid NN outputs provided a "first pass" at the roughness speedups, and the CNN was intended to fine tune them. To test this idea, the outputs from a Grid NN model at 100m AGL, along with the terrain data, were used as inputs to a series of CNN architectures. Unfortunately, the majority of these CNNs gave very poor outputs; an example of one of the most promising runs is shown in Figure 7.13. This combination of approaches did not show much merit, and a full search of all the CNN parameters was not thought to be worthwhile at this stage.

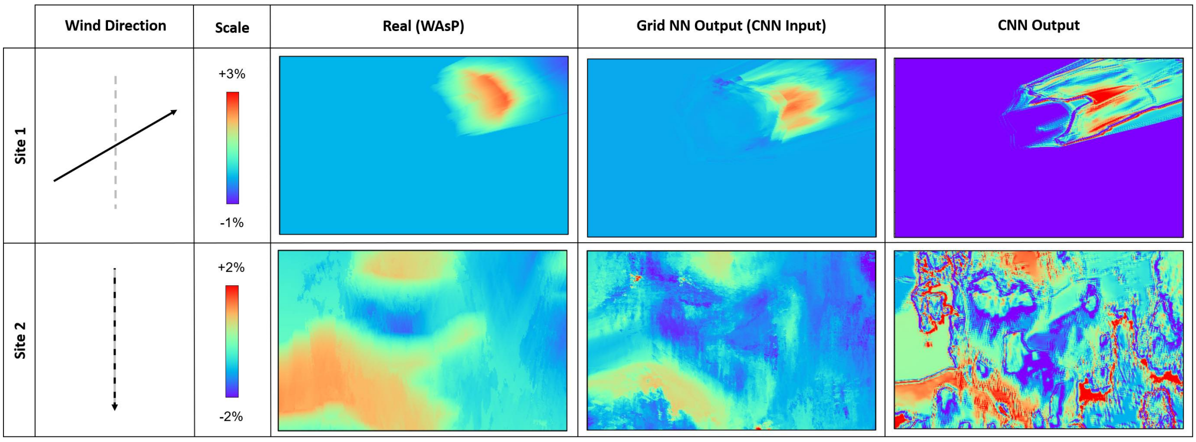


FIGURE 7.13. Combining the Grid NN and CNN approaches for roughness speedup prediction at 100m AGL, by using the output from a Grid NN as the input to a CNN (as well as other CNN inputs as detailed in Table B.18). For Site 1 (top) and Site 2 (bottom), plots shown are (left-right): WAsP calculated speedups; Grid NN predicted speedups used as CNN inputs; CNN predicted speedups.

7.4.5 Final Models

As with the orographic speedup and turn final Grid NN models, the final roughness speedup Grid NN models for each of 10m and 100m AGL had a fix applied to the input sub-grid patterns, but otherwise were the same as the 500 neuron runs described in the previous sections. Each of the final Grid NN roughness speedup models had the following parameters:

- Inputs:
  - natural log of inverse roughness, natural log of inverse meso-roughness, difference of the natural logs of inverse roughness and inverse meso-roughness, number of roughness changes;
  - normalised direction (rather than rotating the input and output grids to have consistent direction);
- Outputs:
  - single point output of roughness speedup factor;
  - no data balancing;
- Deep Neural Network:
  - 10 layers of 50 neurons, each followed by batch normalisation and ReLU activation;
- Training / Validation:

- trained on data from seven unique sites (a consistent subset of sites 1-11 in Appendix A);
- trained for between 50 and 60 epochs, using early stopping to terminate training at the best epoch with a patience of three epochs;
- trained with MSE loss and the Adam optimiser.

The loss curves of the final Grid NN models still show a degree of overfitting (Figure 7.14 presents the loss curves at 100m AGL), which has been discussed in Section 7.4.3. For ease of comparison, a summarised table of performance metrics is presented here for the final Grid NN models at both heights AGL, for both validation and inference data.

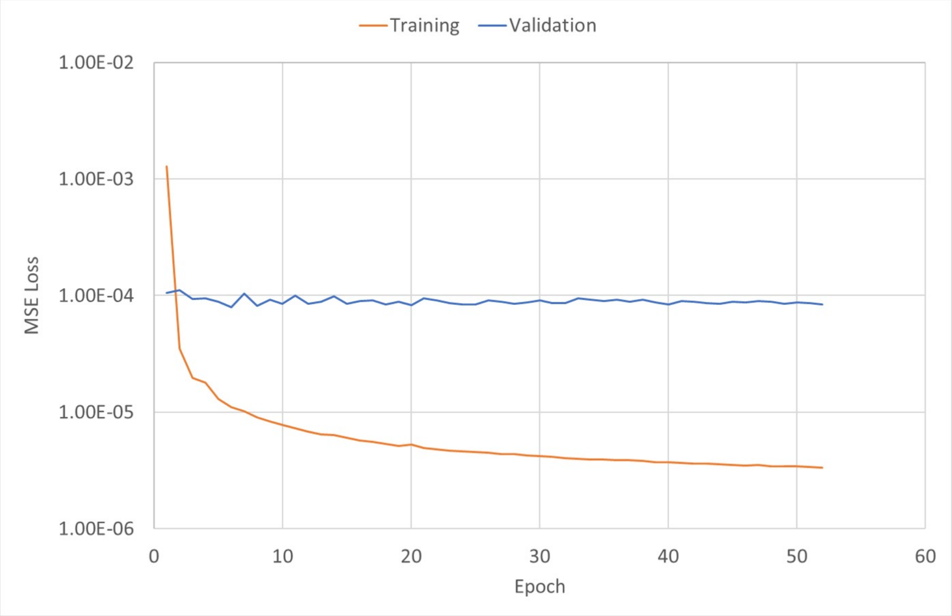


FIGURE 7.14. Loss curves for final Grid NN for roughness speedup at 100m AGL.

Run ID	Height AGL (m)	Validation / Inference	MAE	Error Std. Dev.
RS-G-22	10	Validation	0.768	1.45
RS-G-22	10	Inference (New Sites)	0.472	1.16
RS-G-23	100	Validation	0.517	0.980
RS-G-23	100	Inference (New Sites)	0.384	0.953

Table 7.4: Performance metrics for the final roughness speedup Grid NN models in % speedup.

7.4.5.1 Validation

Note that for consistency with the final orographic speedup and turn models, the final roughness speedup models were run on the validation data from sites 1-11, despite being trained on a subset of seven of these sites.

The final Grid NN model for predicting roughness speedup values at 10m AGL used an input sub-grid size of 1.2km square; the speedups predicted by this model are given in Figure 7.15. With the exception of Site 2, the speedups produced by this Grid NN match the ground truth well in terms of scale and shape. For Site 2 there is a noticeable difference in the scales of the speedup, but the detailed shapes of the speedup areas are captured well. The error maps show that the largest differences occur at sharp transitions between different speedup magnitudes, which matches the errors seen at sharp elevation slopes in the orographic speedup models (Section 5.4.5.1). A common strength between the Grid NN models for orographic and roughness speedups was the ability of these models to predict the correct speedups over flat areas, or terrain with no roughness changes. The close correlation of the input maps of roughness, meso-roughness and roughness changes to the speedup at 10m AGL meant that the Grid NN could re-scale and combine the input maps to give a good approximation of the speedup. However, the number of neurons in the Grid NN is sufficiently large that more complex relations are likely to have been learned. One significant area of improvement over the CNNs tested is shown in Site 1, where the Grid NN has learned to apply the increase in speed due to the transition from grass to lake, slightly downstream of the transition. The error bar plot in Figure 7.16 confirms that most differences (91%) between the model and WAsP outputs are within ±2% speedup, with very few extreme errors. These results are still in need of some smoothing, and while the methods tested here (multiple output points, additional CNN) were not successful, it would be a useful avenue to try in the future.

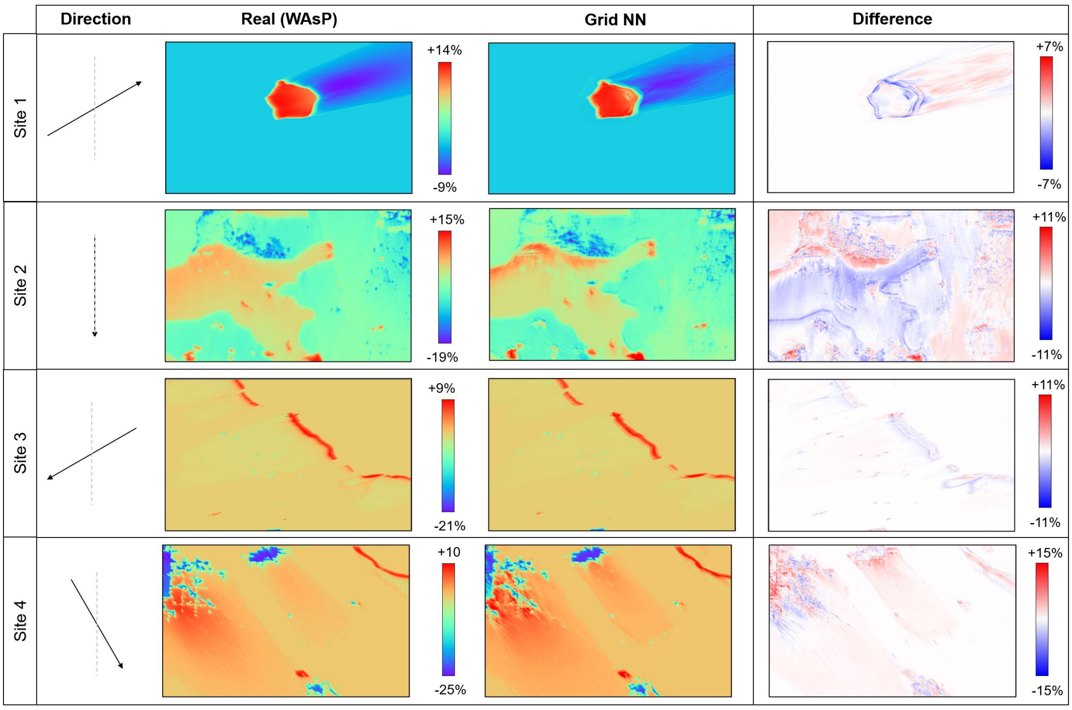


FIGURE 7.15. Final Grid NN model for predicting roughness speedup at 10m AGL for (top-bottom) Sites 1, 2, 3 and 4, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

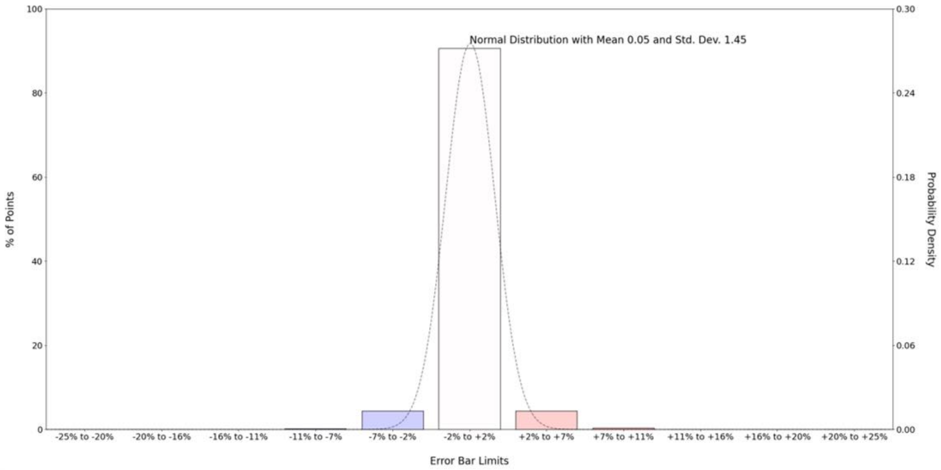


FIGURE 7.16. Bar chart of the errors (Grid NN - WAsP) in roughness speedup over the validation data using the final Grid NN model at 10m AGL.

Results from the final Grid NN model for roughness speedups at 100m AGL are given in Figure 7.17; this model used an input sub-grid size of 3.2km square. The loss curves of this model showed evidence of overfitting, as was discussed in Section 7.4.3 for previous runs. The outputs from the model for Sites 1 and 4 were reasonable, with some errors in areas of abrupt change in speedup, and in Site 1 where the model has had to "erase" the shape of the lake from the speedup map. With Figure 7.17, Site 1 demonstrates the ability of the Grid NN approach to handle multiple spatial scales, as these models have learned to apply the effects of roughness transitions on speedup at different distances downstream at different heights, unlike the CNNs that were tested. There is generally good correlation between the predicted and ground truth speedup values, as seen in Figure 7.18. The results for Site 3 have the correct speedup shape, but have incorrect values; however as the overall scale of these speedups are between ±1% there is likely to be small overall error for this site. Site 2, as always, gives very poor performance at 100m AGL. There are some areas of high and low speeds in this site which approximate the WAsP speedup map, but judging from previous results it is likely that the model has overfit the training data from this site. There is no obvious reason for Site 2 to be more challenging than the others for the Grid NN to predict; it could be that as the sub-grid is smaller than the large roughness area, the sub-grids do not contain sufficient information to predict the changes in speed caused by this terrain feature. The majority of errors are between ±1% speedup as shown in Figure 7.19.

Since the input and output grids are not rotated, the directional information comes from both the normalised wind direction input, and the input maps themselves, which is impressive given the relatively sparse patterns of input sub-grid points provided and the highly direction-dependent speedups.

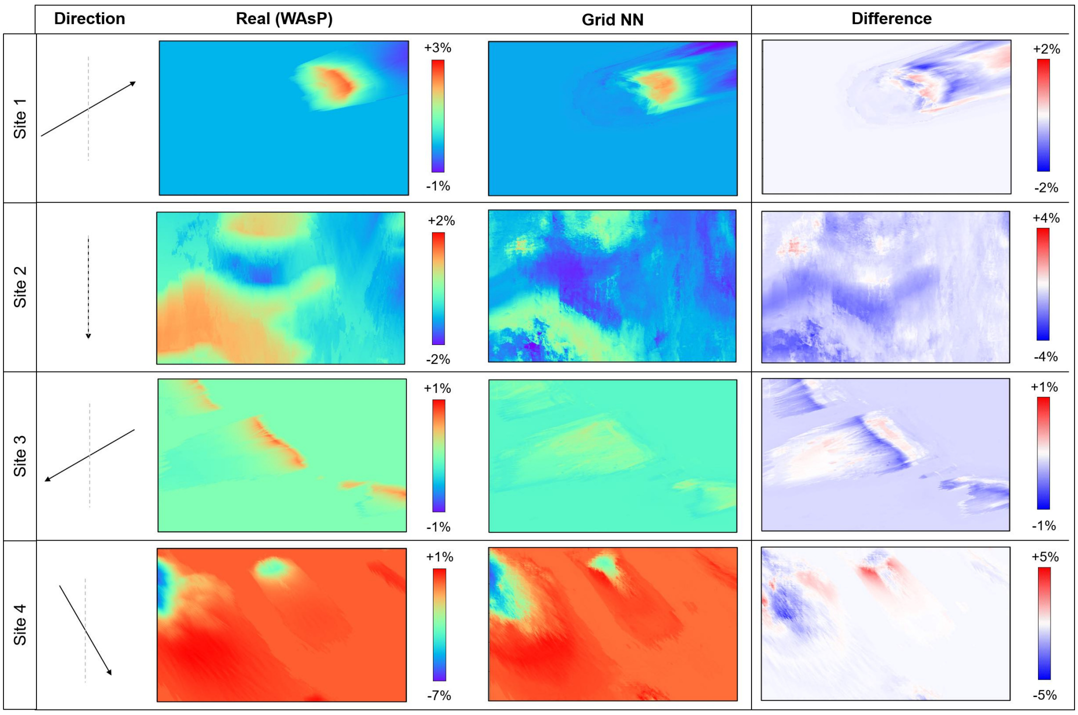


FIGURE 7.17. Final Grid NN model for predicting roughness speedup at 100m AGL for (top-bottom) Sites 1, 2, 3 and 4, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

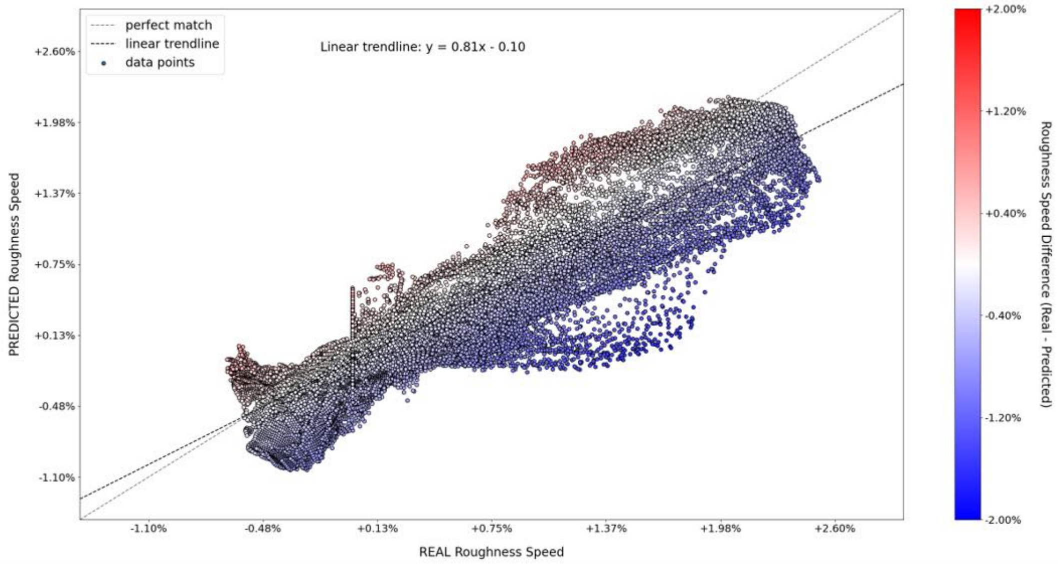


FIGURE 7.18. Scatter graph of the roughness speedup at each grid point predicted by the final Grid NN at 100m AGL compared to the WAsP calculated values, for Site 1 sector 9 (wind from 240°).

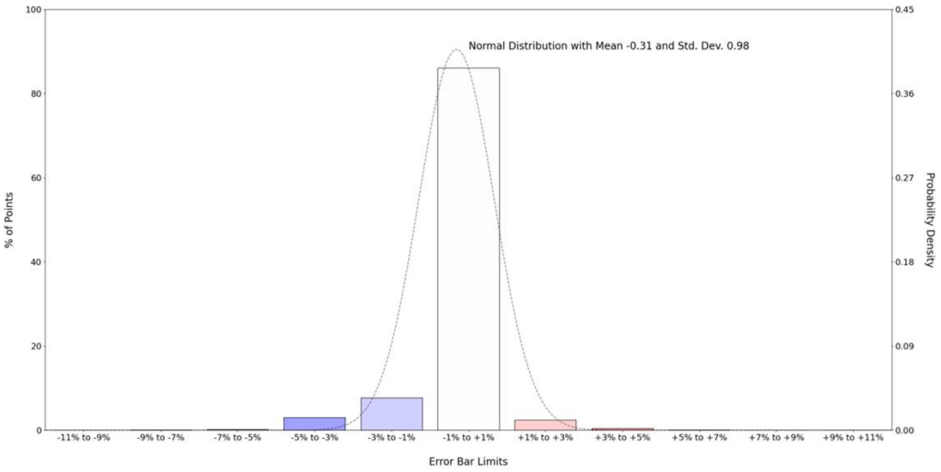


FIGURE 7.19. Bar chart of the errors (Grid NN - WAsP) in roughness speedup over the validation data using the final Grid NN model at 100m AGL.

7.4.5.2 Inference (Testing)

When the final Grid NN model for roughness speedup at 10m AGL was run on Sites 12-14 for inference, the results (Figure 7.20) were very promising, with 94% of errors between ±2.4% speedup (Figure 7.21). Two of the sites (Sites 12 and 14) tested contained well-defined areas of roughness and hence speedup, which the Grid NN can predict well at this height. The roughness speedup over the valley in Site 13 is also calculated with reasonable accuracy by the Grid NN, although the error maps show that the "roughness persistence" area downstream of the valley feature has not been captured. The MAE at 10m AGL over these new sites was 0.47% (Table 7.4), compared to 0.77% for the validation data. This increase in accuracy over the test sites is unusual, but it is likely to be due to the (unintentional) choice of inference sites with well-defined areas of constant roughness, as the Grid NN model predicts the speedups over these terrains well.

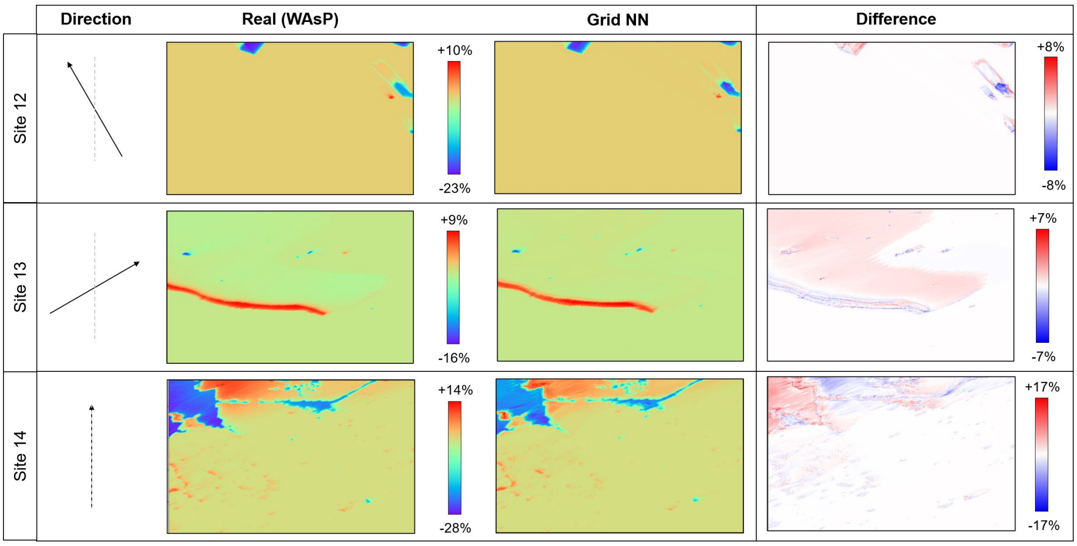


FIGURE 7.20. Final Grid NN model for predicting roughness speedup at 10m AGL for (top-bottom) Sites 12, 13 and 14, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

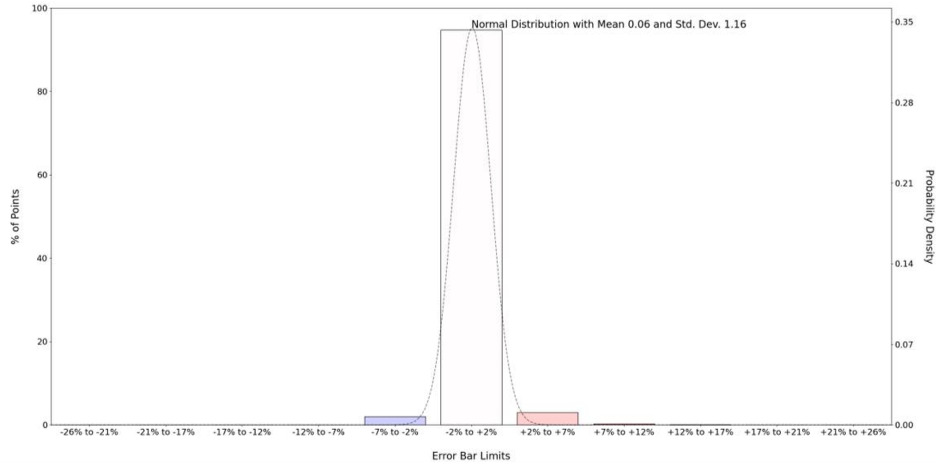


FIGURE 7.21. Bar chart of the errors (Grid NN - WAsP) in roughness speedup over three new sites using the final Grid NN model at 10m AGL.

The inference results for the final Grid NN at 100m AGL are given in Figure 7.22. Sites 12 and 13 had very low roughness speedup magnitudes at 100m AGL, so the errors are exacerbated in Figure 7.22. Sites 12 and 14 contained relatively small areas of speedup changes in the directions shown in this figure. The speedups predicted for these sites at 100m AGL by the final Grid NN have room for improvement, as the slowdown shape in Site 14 is badly defined, and the magnitudes of the speedups in the other sites are inaccurate. The scatter graph of the real versus predicted speedup values over Site 12 (with wind from 150°) in Figure 7.23 shows poor correlation between the model and WAsP values, but the magnitudes of the speedups are small, and from Figure 7.22 most of this site has constant roughness speedup. The error bar chart in Figure 7.24 confirms that most errors are small, with no general under- or overprediction from the Grid NN. Overall, the MAE over these sites at 100m AGL is 0.38%, compared with an MAE of 0.52% over the validation data; again the decrease in MAE over the inference sites as unexpected, but likely to be due to the Grid NN's ability to predict large areas of zero speedup well.

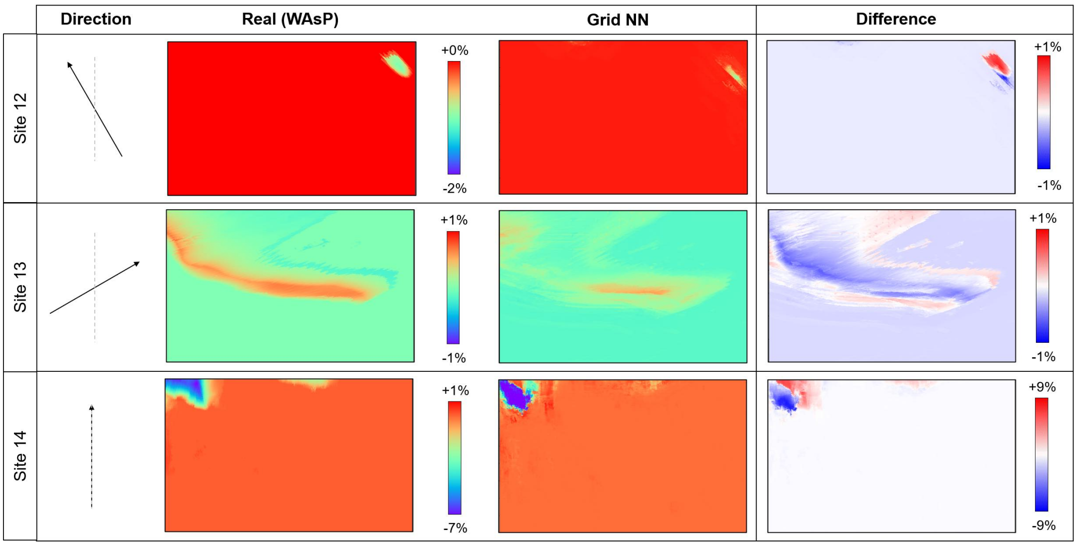


FIGURE 7.22. Final Grid NN model for predicting roughness speedup at 100m AGL for (top-bottom) Sites 12, 13 and 14, showing (left-right): WAsP calculated speedup; Grid NN predicted speedup; the difference (Grid NN - WAsP).

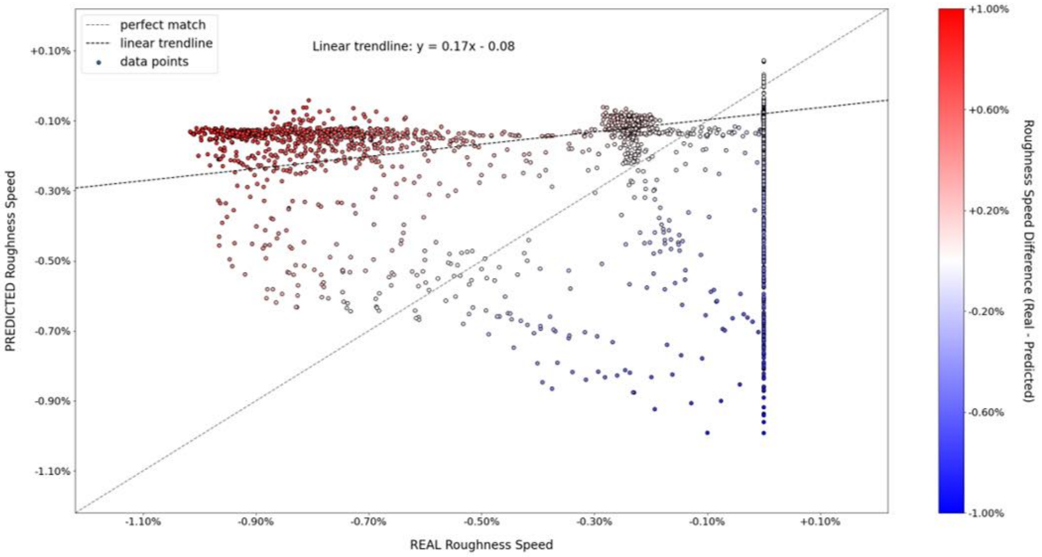


FIGURE 7.23. Scatter graph of the roughness speedup at each grid point predicted by the final Grid NN at 100m AGL compared to the WAsP calculated values, for Site 12 sector 6 (wind from 150°).

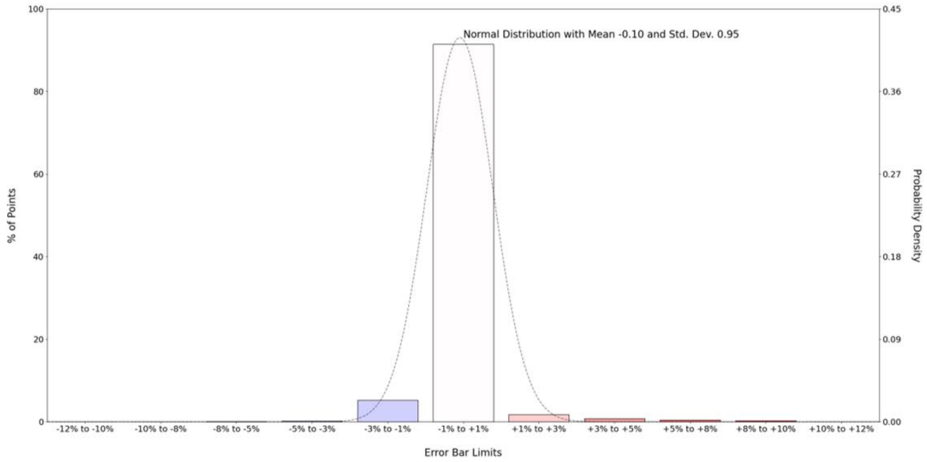


FIGURE 7.24. Bar chart of the errors (Grid NN - WAsP) in roughness speedup over three new sites using the final Grid NN model at 100m AGL.

7.5 Conclusions

The investigations into roughness speedup surrogate modelling presented here have given further insight into the workings of the Grid Neural Network approach, demonstrating the strength of these models when the output can be approximated well by re-scaling and combining the input variables. The results from the final Grid Neural Network models at 10m AGL had superior performance to those at 100m AGL, but there were some sites for which the final Grid NN model at 100m AGL performed well, as shown in Figure 7.25. This work has provided evidence that the Grid NN method can out-perform Convolutional Neural Networks at calculating the wind flow over rough terrain with a limited amount of training data, and appropriate sub-grid sizing.

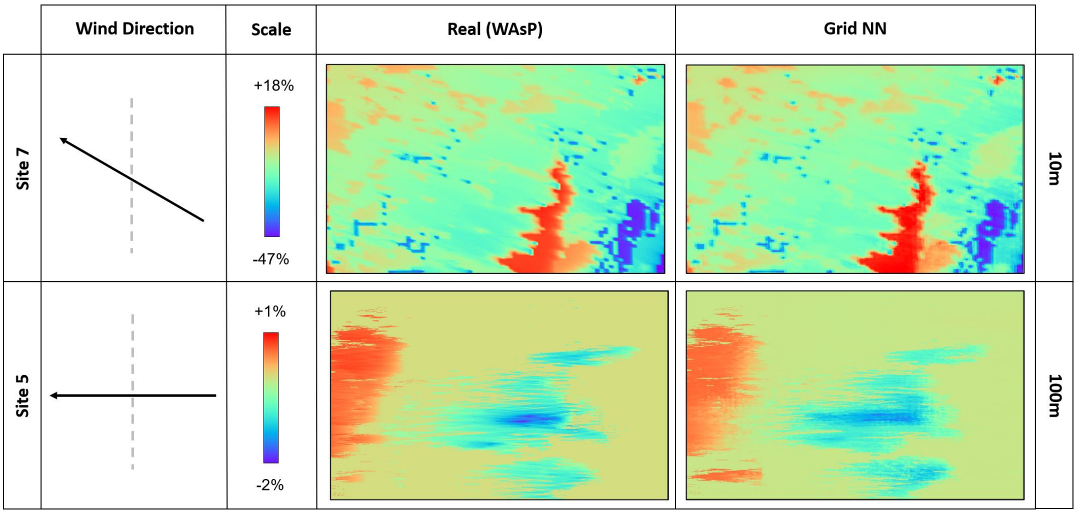


FIGURE 7.25. Examples of two sites and directions for which the final Grid NN models at (top) 10m AGL and (bottom) 100m AGL perform well. Shows the (left) WAsP and (right) Grid NN predicted roughness speedup values.

This is the final WAsP variable for which surrogate modelling was investigated, and together with the data-driven orographic speedup and turn models, produced an interesting set of results to consider. Conclusions and suggestions for further work follow in the next chapter.

Chapter 8

Conclusions and Further Work

8.1 Conclusions

In this work, grid-based machine learning techniques were used to create data-driven models for the calculation of terrain-induced wind velocity changes, with the aim of emulating the WAsP orography and roughness sub-models at multiple heights above ground. Investigations were undertaken into the creation of separate machine learning surrogate models for predicting three different variables at 10m and 100m AGL: orographic speedup, orographic turn and roughness speedup.

Recent research projects into the creation of data-driven CFD models, particularly those with grid-based inputs and outputs, have demonstrated that Convolutional Neural Networks can work well as surrogate flow models. Examples of recent research relevant to this work are the papers of Bhatnagar et al. [52] and Thuerey et al. [50], both of which use 2-D spatial grids containing an aerofoil, along with flow conditions (such as free stream speeds and angle of attack) as inputs to autoencoder-style CNNs, to predict the velocity and pressure fields around the aerofoil on 2-D grids. Given the similarities between this problem and that of predicting the wind flow-field over a 2-D grid of terrain, it was initially thought that these architectures would also be successful as WAsP surrogate models.

The first set of investigations focussed on predicting the orographic speedup (i.e. the change in wind speed due to terrain elevation), starting with CNNs based on those described above in [52] and [50], with autoencoder and U-Net [49] style architectures. Despite testing with a range of filters and hyperparameters, even the most promising of these CNN models had mean absolute errors of ~2% speedup (at 100m AGL) and gave visibly poor performance. A possible reason for the lack of success with this type of Neural Network could be the relatively small data set available for training and testing; whereas Thuerey et al. [50] had ~27,000 pairs of input-output data, in this work there were ~250 pairs of input-output grid data.

Following a lack of success in building Convolutional Neural Networks for these tasks, the novel Grid Neural Network method was developed here, intended to find optimal filter sizes to inform the CNN tests. This Grid NN approach split the terrain and speedup data grids into smaller sub-grids (analogous to CNN filters), taking input-output data pairs for model training and validation from these sub-grids to train a feed-forward, fully-connected Deep Neural Network. Models with Grid NN architectures proved to be trainable with far less data than the CNNs, with the optimal sub-grid sizes providing an understandable measure of the spatial input information required to calculate a given output value. Looking at the outputs produced by various Grid NN tests, this network appears to "re-scale" and "re-print" combinations of the input variables, which can result in areas of poor performance around edges in the terrain, such as the perimeter of roughness areas or steep elevation slopes, where the influence of the edge is extended by the Grid NN beyond that seen in WAsP. This Grid NN method proved more successful than the CNNs tested for orographic speedup, achieving generally good correlation between the predicted and ground truth speedup values at both 10m and 100m AGL for a range of sites with different complex terrain features. There are similarities between the Grid NN method, which takes input points in a concentric squares pattern from each sub-grid, and the zooming polar grid of the WAsP orography model. This work found an optimal sub-grid size (akin to a radius of influence) of around 1.2km x 1.2km, applied to the terrain data to give good orographic speedup predictions from the Grid NN models at both 10m and 100m AGL. The final models used the sub-grid size of 1.2km square, with a DNN of 500 neurons (with batch normalisation and ReLU activation after each layer), and produced MAE values of 3.7% and 1.6% speedup on blind test data for 10m and 100m AGL respectively.

As the WAsP orography sub-model calculates the changes in both the wind speed and direction (turn) over terrain, the orographic turn surrogate model tests were heavily influenced by the results of the orographic speedup tests, and so CNNs were not trialled for orographic turn. Instead, Grid NN models were investigated using different input sub-grid sizes, input variables and output normalisation. From this, an optimal input sub-grid size of 1.2km x 1.2km was found (applicable at both 10m and 100m AGL), and using a DNN configuration with 500 neurons (with batch normalisation and ReLU activation), the final Grid NN models calculated orographic turn values for blind test data with MAE of 1.4° and 0.45° for 10m and 100m AGL respectively. The orographic turn Grid NN models struggled to predict accurate turn around large terrain features, such as large hills, which may be related to the size of the input sub-grids, or could mean that more input variables are needed (such as additional directional elevation gradients). Since CNNs were not tested for the orographic turn, it is not possible to say whether they would be suitable surrogate models here, but based on the performance of CNNs for orographic speedup it is unlikely with the amount of training data available here.

After the orography sub-model, machine learning models were tested to emulate the WAsP roughness sub-model, i.e. the change in wind speeds due to the terrain roughness. Again, the first stage was to use Convolutional Neural Networks in the style of autoencoders, testing multiple layers, filter sizes, and hyperparameters. The roughness speedups at 10m AGL were of larger magnitudes, and produced speedup maps strongly correlated to the terrain maps, and so most tests used the data at this height. The inclusion of batch normalisation layers was crucial to give any reasonable results, and convolutional filter sizes were required to be small to avoid excessive blurring and edge effects. Transfer learning was attempted to achieve good speedup predictions at 100m AGL based on pre-trained CNNs at 10m AGL, but this was not successful. The CNNs did not learn to apply the wind speed changes downstream (i.e. away from) the changes in roughness, which is the physical basis of this sub-model in WAsP. The MAE of the most promising CNN at 10m AGL was 1.7% speedup.

Moving on from the CNNs, Grid Neural Networks were used to predict the roughness speedup at both 10m and 100m AGL. The input and output sub-grid sizes were investigated, giving optimal input sub-grids of 1.2km x 1.2km at 10m AGL and 3.2km x 3.2km at 100m AGL, and single point outputs for both. The Grid NN models excelled at situations where the output grid was of a very similar form to the input grids; the best example of this is the strong performance of the final Grid NN model for roughness speedup at 10m AGL. Significant issues with overfitting of the Grid NN models were found at 100m AGL, but attempts to prevent this through weight decay and dropout were unsuccessful. Notably this Grid NN was able to learn to apply roughness speed changes downstream of the roughness transitions that had caused them, which the CNNs were unable to achieve. The Grid NN method is perhaps not applicable as a surrogate model for roughness speedups at multiple vertical heights, as the physical basis of the speedups and how they are calculated in WAsP is very different to the Grid NN approach. The Mean Absolute Errors for the final Grid NN models for roughness speedup over blind test data at 10m and 100m AGL were 0.47% and 0.38% speedup respectively.

Despite the connections between the filters of CNNs and the sub-grids of the Grid NNs, the Grid NN models created here had superior performance compared to the CNNs tested for all three types of output variables. There was a relatively limited amount of training data available in this work, particularly compared to the volume of data used to train CNNs for similar fluid flow surrogate models; it is possible that CNNs would work as surrogate models for WAsP if trained with enough data. Another route that could have been taken would be to apply transfer learning to pre-trained CNN models, adjusting them to this particular use case. However, the Grid NN method proved to be versatile and easy to train with a small number of real sites (11 sites with 12 wind directions each), and produced models with good performances for a range of terrain features (valleys, hills, plains) and two different ground truth WAsP sub-models (orography and roughness). Examples of the predictions from these Grid NN models for each type of output are presented in Figure 8.1.

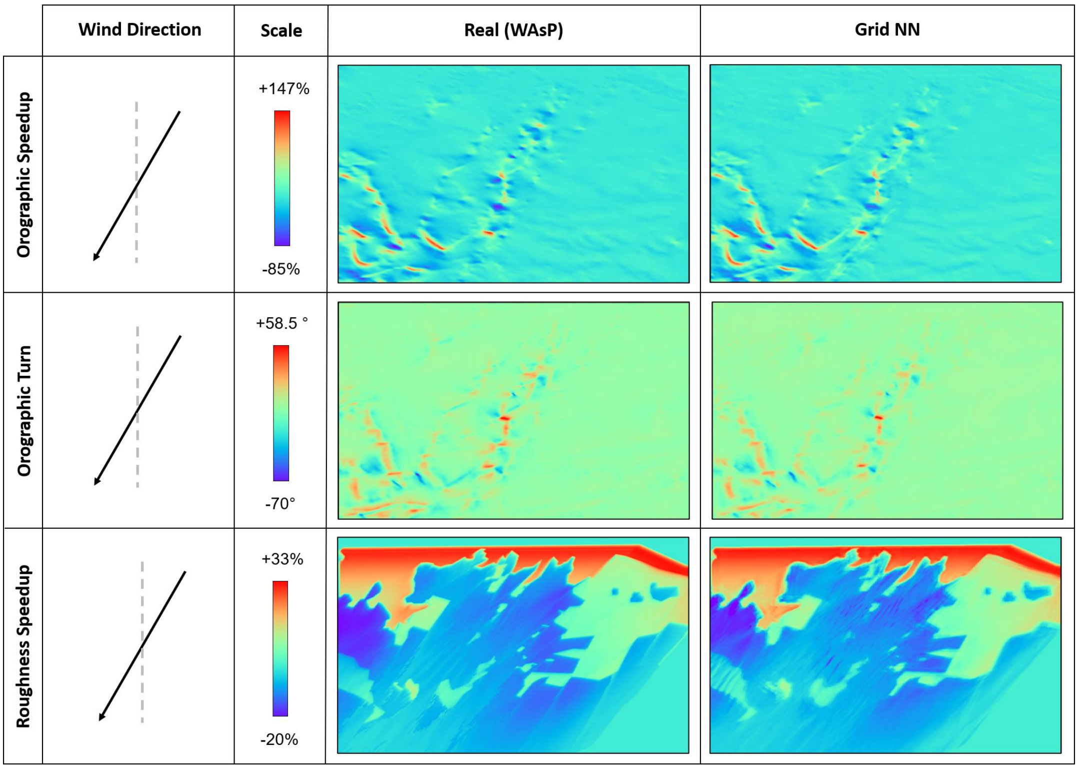


FIGURE 8.1. Examples of predictions from the final Grid NN models at 10m AGL for a single site and direction (Site 8 with wind from 30°). Shows (top-bottom) orographic speedup, orographic turn and roughness speedup from (left-right) WAsP and Grid NNs.

Validation of the WAsP model in Bowen & Mortensen [70] showed that the errors in mean wind speeds compared to measured data are between ±2% for a range of sites, which is a good level of agreement. The errors in the surrogate models created must be combined with the known accuracy level of the WAsP model itself, but given that this has been proven relatively small, the WAsP surrogate models that perform well (e.g. orographic speedup at 10m AGL) can be considered reasonably accurate flow models over terrain which does not contain steep slopes.

This work has proven that an appropriately sized set of filters over terrain data, coupled with a sufficiently large Deep Neural Network, can calculate the changes in wind speed and direction due to the terrain with small errors. While the WAsP model used as ground truth here is a lower fidelity model than full Computational Fluid Dynamics, it is still industry-standard, and the techniques used in this project could be investigated for data-driven CFD, which would be of great interest for both industry and academia. There is significant interest in the wind energy community on wind flow models which can be run within seconds, for use in wind farm control; the surrogate models created here could potentially be further developed to be suitable for such an application. Additionally, while the offshore wind market continues to grow, it is still easier to install fixed offshore rather than floating wind turbines, which are likely to be close to coasts. Hence, terrain-based wind resource modelling remains a valid stage of wind farm location assessment, to determine the effects on the wind of terrain both inland and coastal.

8.2 Further Work

There are a number of investigations that could be explored further in future work, namely:

- conducting a more in-depth hyperparameter search on the optimiser choice, learning rate, momentum and loss function for the Grid NN models;
- further investigation into the prevention of overfitting in the Grid NN models, particularly for orographic turn and roughness speedup;
- extending the surrogate models to calculate speedups or turn at user-specified heights, either by incorporating the height as an input to the Grid NN models, or interpolating/extrapolating from separate Grid NN models at set heights;
- combining the orographic speedup and turn, and the roughness speedup, predicted by the individual Grid NN models to produce data-driven Wind Resource Grids;
- investigating the use of the techniques that have been found to be successful here to form surrogate models for Computational Fluid Dynamics Wind Resource Grids.

BIBLIOGRAPHY

[1] IPCC, “Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change,” 2021.

[2] United Nations, “Paris Agreement,” 2015.

[3] European Commission, “EU Climate Target Plan 2030,” 2020.

[4] UK Department for Business, Energy & Industrial Strategy, “2020 UK greenhouse gas emissions, provisional figures,” 2020.

[5] IRENA, “Future of wind: Deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation paper,” 2019.

[6] Y. Jiang, G. Huang, Q. Yang, Z. Yan, and C. Zhang, “A novel probabilistic wind speed prediction approach using real time refined variational model decomposition and conditional kernel density estimation,” *Energy Conversion and Management*, vol. 185, pp. 758–773, 2019.

[7] N. Jensen, “A note on wind generator interaction,” 1983.

[8] I. Troen and E. Lundtang Petersen, “European Wind Atlas,” *Risø National Laboratory*, 1989.

[9] J. Bleeg, M. Purcell, R. Ruisi, and E. Traiger, “Wind Farm Blockage and the Consequences of Neglecting Its Impact on Energy Production,” *Energies*, vol. 11, no. 6, 2018.

[10] M. P. van der Laan, S. J. Andersen, M. Kelly, and M. C. Baungaard, “Fluid scaling laws of idealized wind farm simulations,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[11] J. M. I. Strickland and R. J. A. M. Stevens, “Effect of thrust coefficient on the flow blockage effects in closely-spaced spanwise-infinite turbine arrays,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[12] Stovall, T. and Pawlas, G. and Moriarty, P., “Wind farm wake simulations in OpenFOAM,” *48th AIAA Aerospace Sciences Meeting*, 2010.

[13] P. S. Jackson and J. Hunt, “Turbulent Wind Flow over a Low Hill,” *Quarterly Journal of the Royal Meteorological Society*, vol. 101, pp. 924–933, 1975.

[14] P. A. Taylor and H. W. Teunissen, “The Askervein Hill Project: Overview and Background Data,” *Boundary Layer Meteorology*, vol. 39, pp. 15–39, 1987.

[15] J. Berg, J. Mann, A. Bechmann, M. S. Courtney, and H. E. Jørgensen, “The Bolund Experiment, Part I: Flow Over a Steep, Three-Dimensional Hill,” *Boundary-Layer Meteorology*, vol. 141, no. 2, pp. 219–243, 2011.

[16] J. Mann, N. Angelou, J. Arnqvist, D. Callie, E. Cantero, R. C. Arroyo, M. Courtney, J. Cuxart, E. Dellwik, J. Gottschall, S. Ivanell, P. Kühn, G. Lea, J. C. Matos, J. M. L. M. Palma, L. Pauscher, A. Peña, J. S. Rodrigo, S. Söderberg, N. Vasiljevic, and C. V. Rodrigues, “Complex terrain experiments in the New European Wind Atlas,” *Philosophical Transactions of the Royal Society*, vol. 375, 2016.

[17] J. S. Rodrigo, P. Gancarski, R. C. Arroyo, P. Moriarty, M. Chuchfield, J. W. Naughton, K. S. Hansen, E. Machefaux, T. Koblitz, E. Maguire, F. Castellani, L. Terzi, S.-P. Breton, Y. Ueda, J. Prospathopoulos, G. S. Oxley, C. Peralta, X. Zhang, and B. Witha, “IEA-Task 31 WAKEBENCH: Towards a protocol for wind farm flow model evaluation. Part 1: Flow-over-terrain models,” *Journal of Physics: Conference Series*, vol. 524, 2016.

[18] K. Solbakken and Y. Birkelund, “Evaluation of the Weather Research and Forecasting (WRF) model with respect to wind in complex terrain,” *Journal of Physics: Conference Series*, vol. 1102, 2018.

[19] J. Finnigan, K. Ayotte, I. Harman, G. Katul, H. Oldroyd, E. Patton, D. Poggi, A. Ross, and P. Taylor, “Boundary-Layer Flow Over Complex Topography,” *Boundary-Layer Meteorology*, vol. 177, no. 2-3, pp. 247–313, 2020.

[20] R. S. Arthur, K. A. Lundquist, D. J. Wiersema, J. Bao, and F. K. Chow, “Evaluating Implementations of the Immersed Boundary Method in the Weather Research and Forecasting Model,” *Monthly Weather Review*, vol. 148, no. 5, pp. 2087–2109, 2020.

[21] Y. Birkelund, S. Alessandrini, O. Byrkjedal, and L. D. Monache, “Wind power prediction in complex terrain using analog ensembles,” *Journal of Physics: Conference Series*, vol. 1102, 2018.

[22] S. Robert, L. Foresti, and M. Kanevski, “Spatial prediction of monthly wind speeds in complex terrain with adaptive general regression neural networks,” *International Journal of Climatology*, vol. 33, no. 7, pp. 1793–1804, 2013.

[23] J. Steiner, R. Dwight, and A. Viré, “Data-driven turbulence modeling for wind turbine wakes under neutral conditions,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[24] L. E. Andersson, B. Doekemeijer, D. van der Hoek, J.-W. van Wingerden, and L. Imsland, “Adaptation of Engineering Wake Models using Gaussian Process Regression and High-Fidelity Simulation Data,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[25] R. N. King, C. Adcock, J. Annoni, and K. Dykes, “Data-Driven Machine Learning for Wind Plant Flow Modelling,” *Journal of Physics: Conference Series*, vol. 1037, 2018.

[26] J. Chatterjee and N. Dethlefs, “Temporal Causal Inference in Wind Turbine SCADA Data Using Deep Learning for Explainable AI,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[27] L. Schröder, N. K. Dimitrov, and J. A. Sørensen, “Uncertainty propagation and sensitivity analysis of an artificial neural network used as wind turbine load surrogate model,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[28] T. Regan, C. Beale, and M. Inalpolat, “Wind Turbine Blade Damage Detection Using Supervised Machine Learning Algorithms,” *Journal of Vibration and Acoustics*, vol. 139, no. 6, 2017.

[29] M. J. van den Broek and J.-W. van Wingerden, “Dynamic Flow Modelling for Model-Predictive Wind Farm Control,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[30] D. van der Hoek, B. Doekemeijer, L. E. Andersson, and J.-W. van Wingerden, “Predicting the benefit of wake steering on the annual energy production of a wind farm using large eddy simulations and Gaussian process regression,” *Journal of Physics: Conference Series*, vol. 1618, 2020.

[31] I. Schicker, P. Papazek, A. Kann, and Y. Wang, “Short-range wind speed predictions for complex terrain using an interval-artificial neural network,” *Energy Procedia*, vol. 125, pp. 199–206, 2017.

[32] M. Khodayar and J. Wang, “Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting,” *IEEE Transactions on Sustainable Energy*, vol. 10, no. 2, pp. 670–681, 2019.

[33] M. Ibrahim, A. Alsheikh, Q. Al-Hindawi, S. Al-Dahidi, and H. ElMoaqet, “Short-Time Wind Speed Forecast Using Artificial Learning-Based Algorithms,” *Comput Intell Neurosci*, vol. 2020, 2020.

[34] Y. Liu, H. Qin, Z. Zhang, S. Pei, Z. Jiang, Z. Feng, and J. Zhou, “Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model,” *Applied Energy*, vol. 260, 2020.

[35] H. Wilms, M. Cupelli, A. Monti, and T. Gross, "Exploiting Spatio-Temporal Dependencies for RNN-based Wind Power Forecasts," 2019.

[36] J. Zhang and X. Zhao, "Spatiotemporal wind field prediction based on physics-informed deep learning and LIDAR measurements," *Applied Energy*, vol. 288, 2021.

[37] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[38] J. Bleeg, "A Graph Neural Network Surrogate Model for the Prediction of Turbine Interaction Loss," *Journal of Physics: Conference Series*, vol. 1618, 2020.

[39] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans Neural Netw*, vol. 20, no. 1, pp. 61–80, 2009.

[40] J. Park and J. Park, "Physics-induced graph neural network: An application to wind-farm power estimation," *Energy*, vol. 187, 2019.

[41] K. Duraisamy, Z. J. Zhang, and A. P. Singh, "New Approaches in Turbulence and Transition Modeling Using Data-driven Techniques," 2015.

[42] B. D. Tracey, K. Duraisamy, and J. J. Alonso, "A Machine Learning Strategy to Assist Turbulence Model Development," 2015.

[43] L. Ladicky, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross, "Data-driven Fluid Simulations using Regression Forests," 2014.

[44] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," 2014.

[45] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine Learning for Fluid Mechanics," *Annual Review of Fluid Mechanics*, vol. 52, no. 1, pp. 477–508, 2020.

[46] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, T. Ying, J. Lskovec, and P. W. Battaglia, "Learning to Simulate Complex Physics with Graph Networks," *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[47] T. Pfaff, M. Forunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning Mesh-Based Simulation with Graph Networks," 2020.

[48] B. Ummenhofer, L. Prantl, N. Theurey, and V. Koltun, "Lagrangian Fluid Simulation with Continuous Convolutions," 2020.

[49] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015.

[50] N. Theurey, K. Weissenow, L. Prantl, and X. Hu, "Deep Learning Methods for RANS Simulations of Airfoil Flows," 2020.

[51] S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," *Journal of Fluid Mechanics*, vol. 879, pp. 217–254, 2019.

[52] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of Aerodynamic Flow Fields Using Convolutional Neural Networks," 2019.

[53] D. J. Gagne II, S. E. Haupt, D. W. Nychka, and G. Thompson, "Interpretable Deep Learning for Spatial Analysis of Severe Hailstorms," *Monthly Weather Review*, vol. 147, no. 8, pp. 2827–2845, 2019.

[54] N. Mortensen, D. Heathfield, O. Rathmann, and M. Nielsen, "Wind Atlas Analysis and Application Program: WAsP 11 Help Facility," 2014.

[55] I. Troen, "A high resolution spectral model for flow in complex terrain," *Symposium on turbulence and diffusion*, 1990.

[56] A. M. Sempreviva, S. Larsen, N. G. Mortensen, and I. Troen, "Response of Neutral Boundary Layers to Changes of Roughness," *Boundary Layer Meteorology*, vol. 50, pp. 205–225, 1990.

[57] D. P. Kingma and J. L. Ba, "Adam - A Method for Stochastic Optimization," 2017.

[58] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*.
MIT Press, 2016.
`http://www.deeplearningbook.org`.

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012.

[60] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Proceedings of ICLR 2015*, 2015.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[62] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2018.

[63] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *ECCV*, pp. 818–833, 2014.

[64] S. Ioffe and C. Szegedy, "Batch Normalization - Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

[65] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2015.

[66] A. Krogh and J. A. Hertz, “A Simple Weight Decay Can Improve Generalization,” 1992.

[67] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks,” 2016.

[68] S. Elfwing, E. Uchibe, and K. Doya, “Sigmoid-Weighted Linear Units for Neural Network Approximation in Reinforcement Learning,” 2017.

[69] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich, “ImageNet Pre-trained CNNs for JPEG Steganalysis,” in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2020.

[70] A. J. Bowen and N. G. Mortensen, “WAsP prediction errors due to site orography,” 2004. Forskningscenter Risoe, Risoe-R No. 995(EN).

APPENDIX A

APPENDIX A

This appendix shows the elevation and roughness maps for each of the 19 terrain areas used for training Grid Neural Networks in this work. Note that the scales for roughness and elevation are unique to each site, in order to best show the features of that site. These plots are not to scale, and do not have coordinates, in order to preserve anonymity.

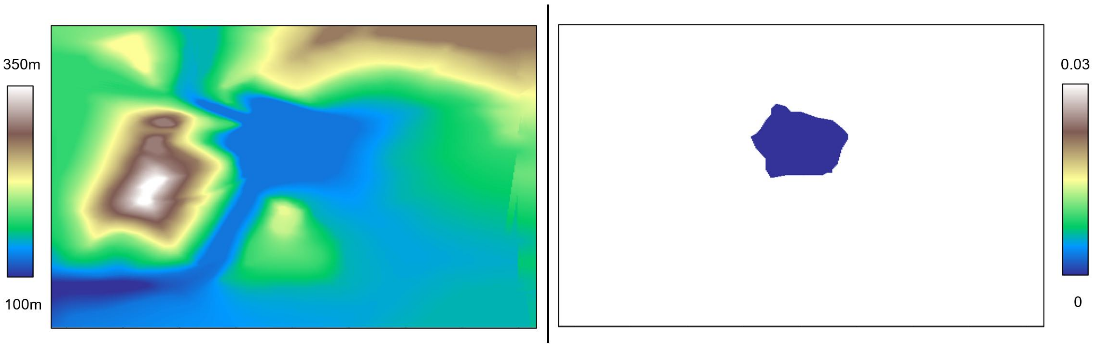


FIGURE A.1. Site 1 elevation (L) and roughness (R) maps.

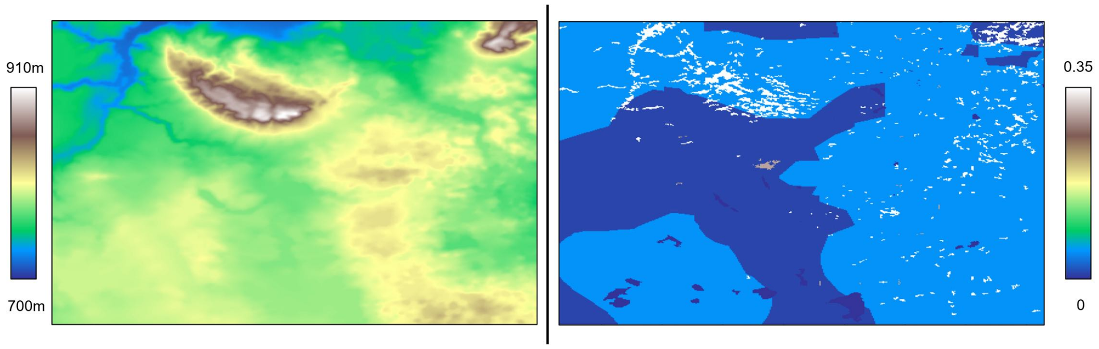


FIGURE A.2. Site 2 elevation (L) and roughness (R) maps.

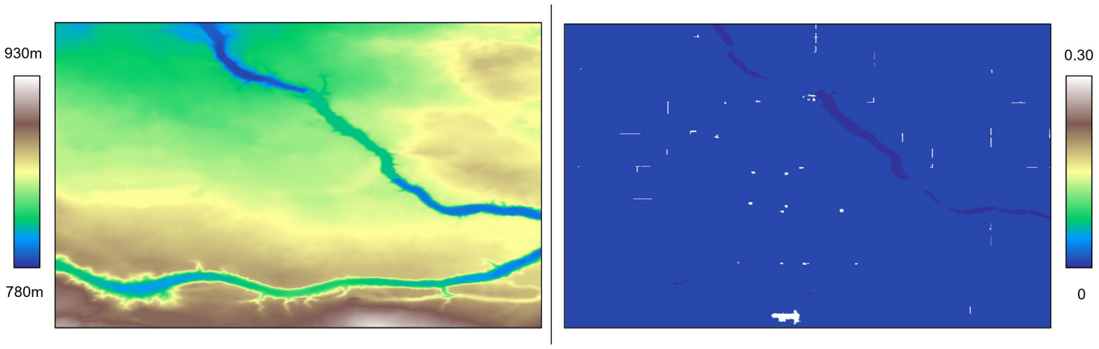


FIGURE A.3. Site 3 elevation (L) and roughness (R) maps.

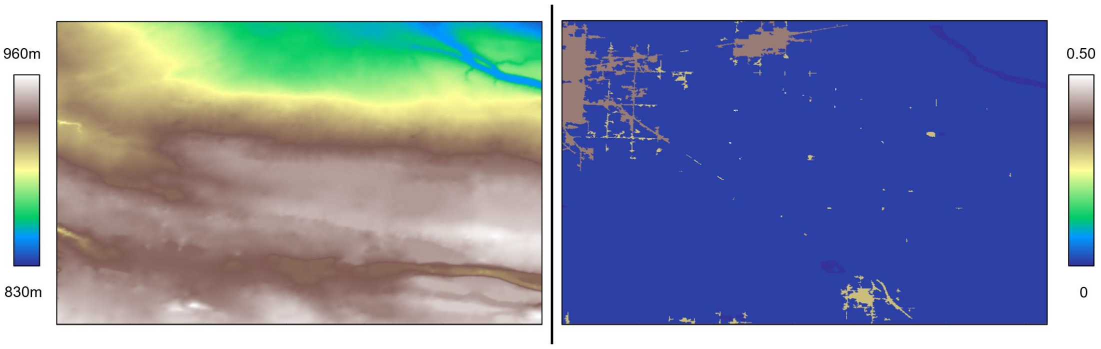


FIGURE A.4. Site 4 elevation (L) and roughness (R) maps.

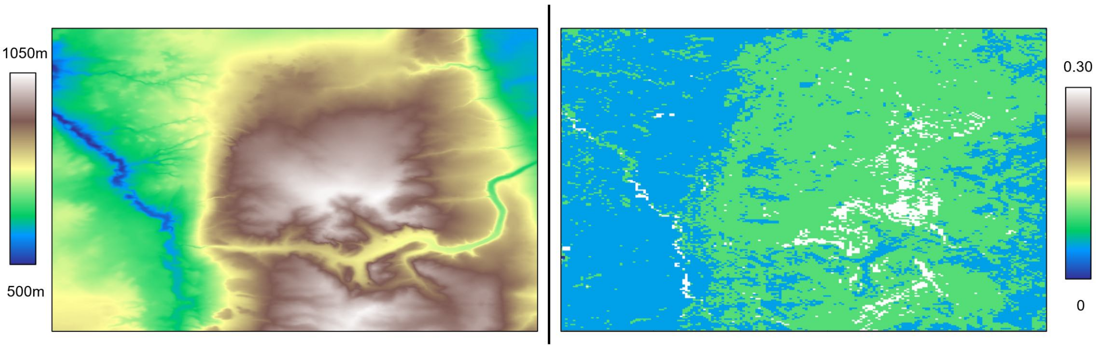


FIGURE A.5. Site 5 elevation (L) and roughness (R) maps.

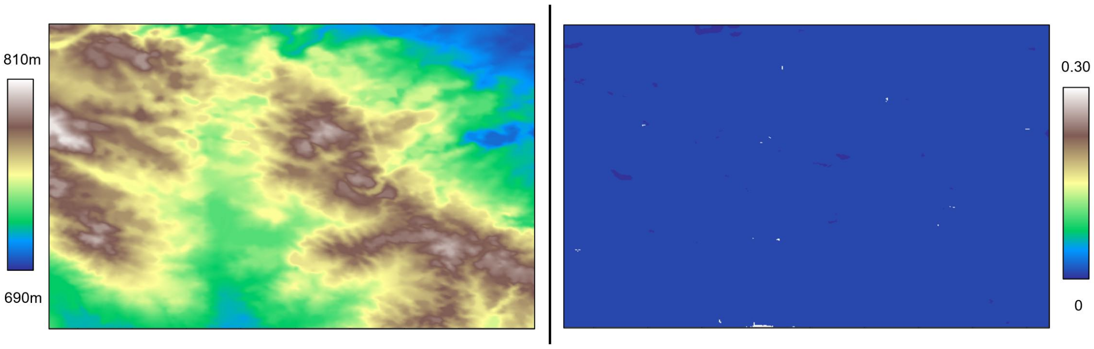


FIGURE A.6. Site 6 elevation (L) and roughness (R) maps.

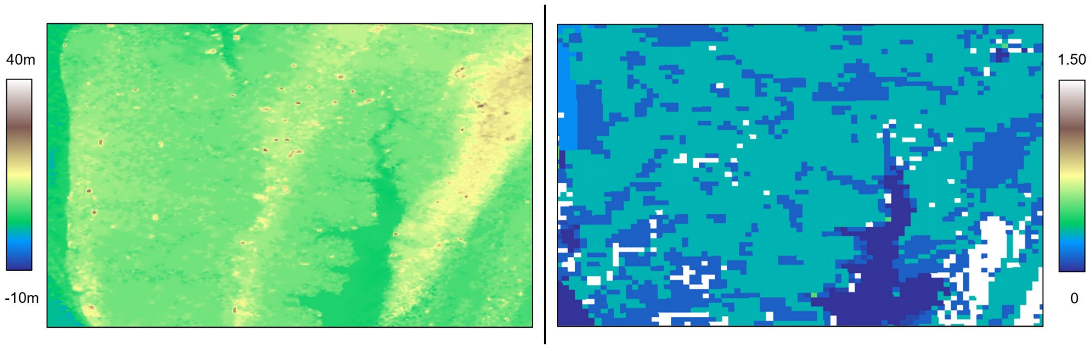


FIGURE A.7. Site 7 elevation (L) and roughness (R) maps.

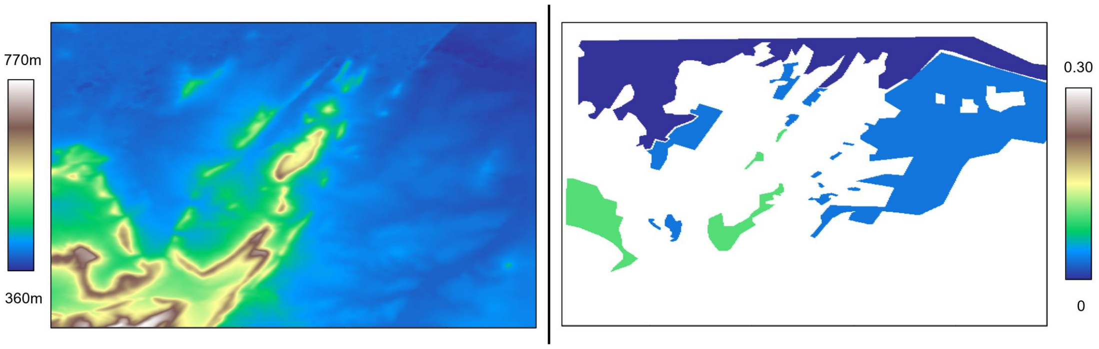


FIGURE A.8. Site 8 elevation (L) and roughness (R) maps.

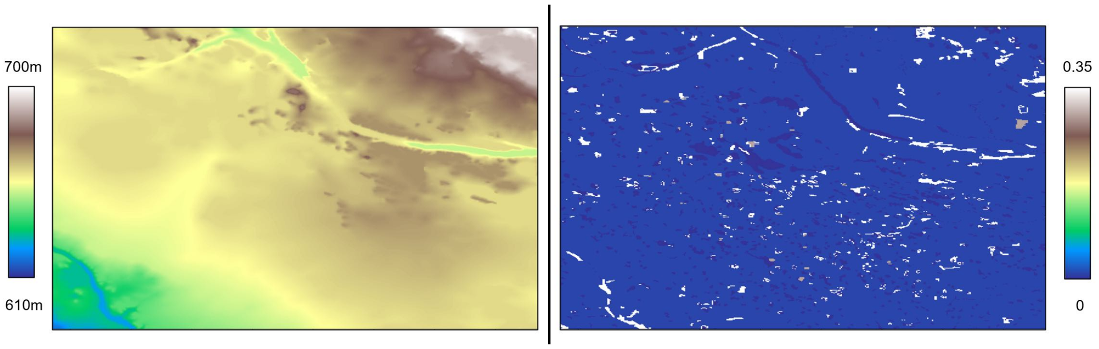


FIGURE A.9. Site 9 elevation (L) and roughness (R) maps.

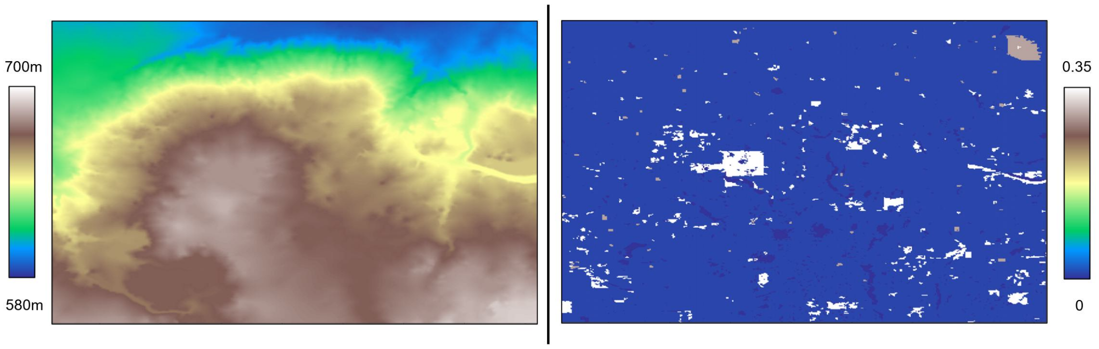


FIGURE A.10. Site 10 elevation (L) and roughness (R) maps.

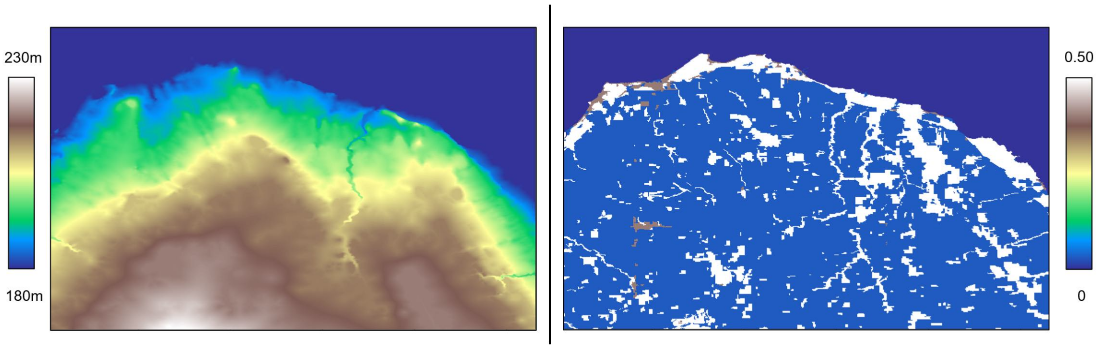


FIGURE A.11. Site 11 elevation (L) and roughness (R) maps.

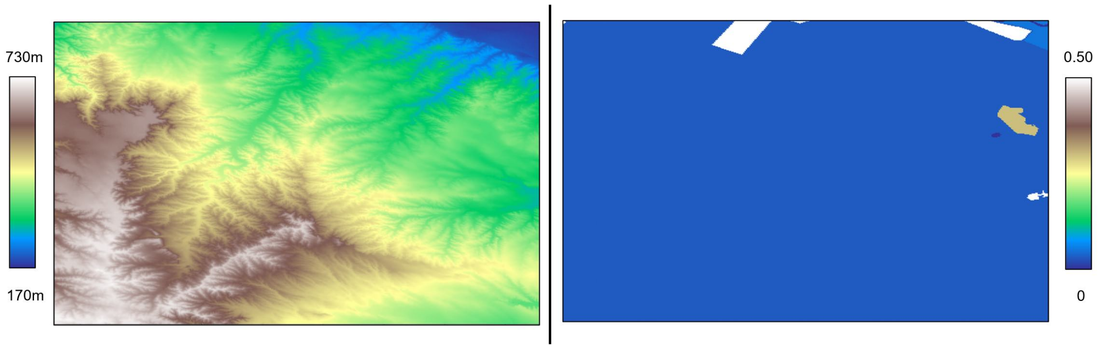


FIGURE A.12. Site 12 elevation (L) and roughness (R) maps.

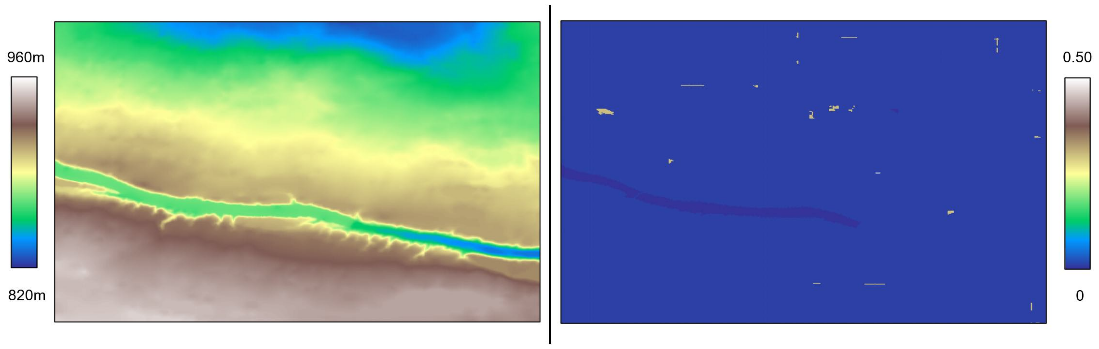


FIGURE A.13. Site 13 elevation (L) and roughness (R) maps.

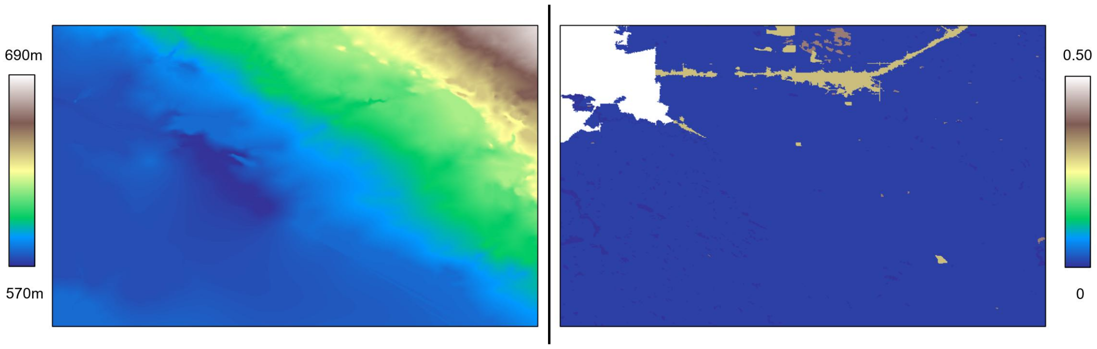


FIGURE A.14. Site 14 elevation (L) and roughness (R) maps.

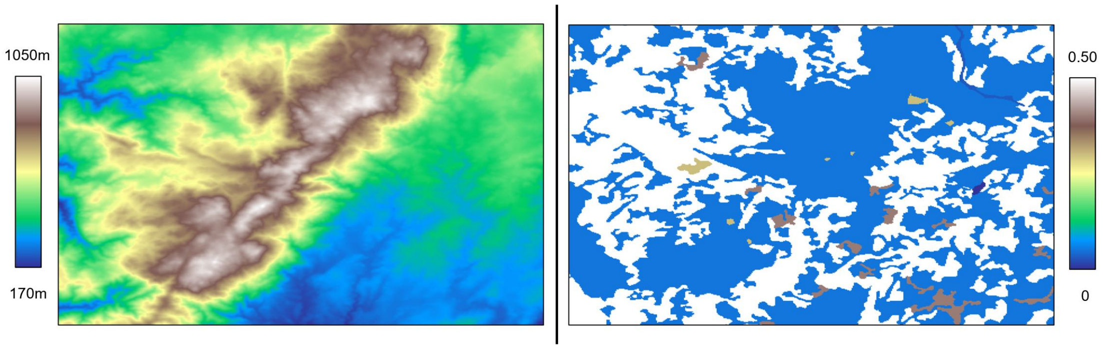


FIGURE A.15. Site 15 elevation (L) and roughness (R) maps.

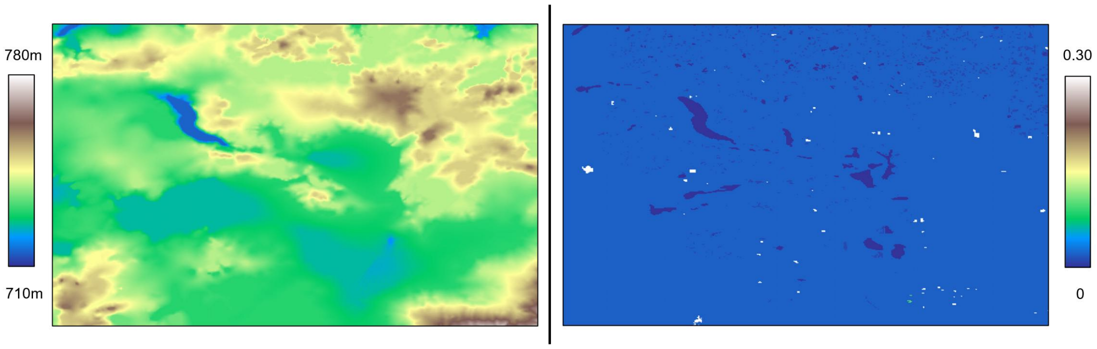


FIGURE A.16. Site 16 elevation (L) and roughness (R) maps.

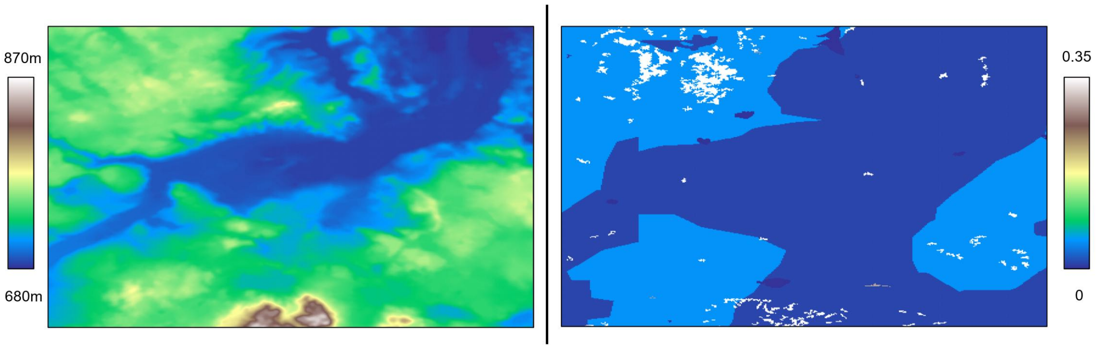


FIGURE A.17. Site 17 elevation (L) and roughness (R) maps.

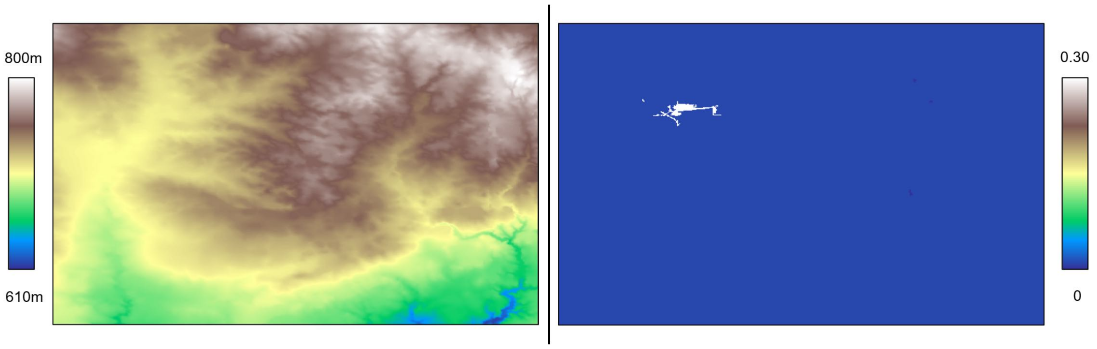


FIGURE A.18. Site 18 elevation (L) and roughness (R) maps.

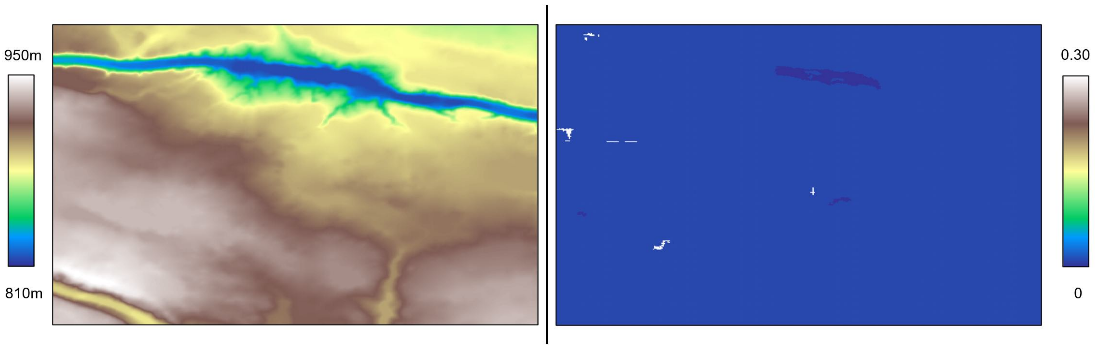


FIGURE A.19. Site 19 elevation (L) and roughness (R) maps.

Appendix B

Appendix B

This Appendix gives tables of performance metrics for all runs described in this thesis. Note that for the metrics:

- "Std. Dev." means standard deviation;
- "Min." means minimum;
- "Max." means maximum;

And for the inputs: Note that for the inputs:

- "MM Elev." is min-max normalised elevation;
- "ZS Elev." is z-scale normalised elevation;
- "Elev. Grad." is elevation gradient parallel to wind direction;
- "Elev. Grad. @ X°" is elevation gradient at angle X clockwise from the wind direction;
- "Ro" is roughness;
- "MR" is meso-roughness;
- "RC" is roughness changes.

For Convolutional Neural Networks:

- conv2d(k=K,c=C) is a 2-D convolutional layer with filter (kernel) size $K * K$, and C number of output channels (if different from number of input channels); zero padding applied and stride length 1 unless otherwise specified;

- convtr2d(k=K,c=C) is a 2-D transposed convolutional layer with filter (kernel) size $K * K$, and C number of output channels; zero padding applied and stride length 1 unless otherwise specified;
- SiLU and ReLU are SiLU and ReLU activations;
- batchnorm2d is a 2-D batch normalisation layer.

For Grid Neural Networks:

- XL YN defines a DNN with X hidden layers, each with Y neurons;
- unless specified, all DNNs have 1-D batch normalisation followed by ReLU activation after each hidden layer.

B.1 Orographic Speedup

B.1.1 Convolutional Neural Networks

The tables of performance metrics for the validation data from each orographic speedup CNN surrogate model described in Chapter 5 are given here.

Run ID	Inputs	Layers	MAE	Error Std. Dev.	Min. Error	Max. Error
OS-C-1	Elev. Grad @ 0°, Elev. Grad @ 180°	conv2d(k=5,c=6) SiLU batchnorm2d conv2d(k=5) SiLU batchnorm2d conv2d(k=3) SiLU batchnorm2d convtr2d(k=3) SiLU batchnorm2d convtr2d(k=5) SiLU batchnorm2d convtr2d(k=5,c=1)	2.02	3.40	-50.4	27.1
OS-C-2	Elev. Grad @ 0°, Elev. Grad @ 180°	conv2d(k=5,c=10) SiLU batchnorm2d conv2d(k=3) SiLU batchnorm2d conv2d(k=1) SiLU batchnorm2d convtr2d(k=1,c=1)	1.89	3.67	-50.9	32.8
OS-C-3	Elev. Grad @ 0°, Elev. Grad @ 180°	U-Net	0.897	1.52	-19.3	15.7

Table B.1: Performance metrics for orographic speedup Convolutional Neural Networks. All at 100m AGL; numbers of epochs ran and input grid size vary; all trained with MSE loss; input and output grids not rotated.

B.1.2 Grid Neural Networks

The tables of performance metrics for the validation data from each orographic speedup Grid NN surrogate model described in Chapter 5 are given here.

Run ID	**Site**	**Input**	**Sub-Grid Size (km)**	**No. Input Pts**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
OS-S-1	1	MM Elev.	Single Point	1	1.43	1.98	-10.1	13.4
OS-G-1	1	MM Elev.	0.2	9	1.40	1.95	-9.24	13.0
OS-G-2	1	MM Elev.	1.2	9	1.31	1.84	-8.51	10.3
OS-G-3	1	MM Elev.	2.7	9	1.36	1.89	-7.18	9.75
OS-S-2	1	Elev. Grad.	Single Point	1	1.67	2.52	-13.7	16.4
OS-G-4	1	Elev. Grad.	0.2	9	1.68	2.54	-13.7	14.3
OS-G-5	1	Elev. Grad.	1.2	9	1.56	2.29	-14.9	12.3
OS-G-6	1	Elev. Grad.	2.7	9	2.24	2.67	-14.3	11.9
OS-G-7	2	MM Elev.	0.2	17	1.03	1.51	-14.67	9.13
OS-G-8	2	MM Elev.	1.2	17	0.966	1.44	-15.6	7.60
OS-G-9	2	MM Elev.	2.7	17	0.954	1.41	-9.62	9.54
OS-G-10	2	Elev. Grad.	0.2	17	1.53	1.96	-18.7	11.5
OS-G-11	2	Elev. Grad.	1.2	17	1.43	1.97	-20.8	14.9
OS-G-12	2	Elev. Grad.	2.7	17	1.44	2.07	-22.0	9.57

Table B.2: Performance metrics for orographic speedup sub-grid size investigations in % speedup. All at 100m AGL; normalised direction as an input variable; consistent DNN configurations.

Run ID	Train Sites	Inputs	Architecture	Epochs	MAE	Error Std. Dev.	Min. Error	Max. Error
OS-G-13	1	Elev. Grad.	5L 10N	50	1.36	1.86	-10.8	10.7
OS-G-14	2	Elev. Grad.	5L 10N	50	1.00	1.34	-11.9	13.7
OS-G-15	3	Elev. Grad.	5L 10N	50	1.10	1.14	-8.91	7.69
OS-G-16	1	Elev. Grad., MM Elev., Rough	5L 10N	50	1.09	1.51	-21.9	7.97
OS-G-17	2	Elev. Grad., MM Elev., Rough	5L 10N	50	1.33	1.49	-8.36	9.33
OS-G-18	3	Elev. Grad., MM Elev., Rough	5L 10N	50	0.457	0.705	-10.5	9.08
OS-G-19	1-4	Elev. Grad., MM Elev., Rough	5L 10N	50	0.790	1.09	-10.5	8.06
OS-G-20	1-4	Elev. Grad., ZS Elev., Rough	5L 10N	50	0.763	1.12	-15.5	9.71
OS-G-21	1-4	Elev. Grad., MM Elev., Rough	5L 10N	100	0.856	1.14	-11.5	10.9
OS-G-22	1-4	Elev. Grad., MM Elev., Rough	Autoencoder	50	0.644	0.911	-10.5	8.20
OS-G-23	1-4	Elev. Grad., MM Elev., Rough	10L 50N	50	0.692	0.928	-8.77	7.13
OS-G-24	1-11	Elev. Grad., MM Elev., Rough	10L 50N	50	0.691	1.11	-15.9	17.7

Table B.3: Performance metrics for orographic speedup model improvements in % speedup. All at 100m AGL; input sub-grids of 1.2km square; single output point; consistent input sub-grid patterns; normalised direction as an input variable.

Run ID	Architecture	Rotate?	Balancing	Batch Norm.	MAE	Error Std. Dev.	Min. Error	Max. Error
OS-G-24	10L 50N	No	-	Yes	0.691	1.11	-15.9	17.7
OS-G-25	Autoencoder	No	-	Yes	0.721	1.17	-16.3	17.7
OS-G-26	5L 10N	No	-	Yes	0.926	1.43	-15.6	18.9
OS-G-27	10L 50N	Yes	-	Yes	0.793	1.43	-24.4	31.3
OS-G-28	10L 50N	No	-	No	1.17	2.17	-24.4	29.0
OS-G-29	10L 50N	No	Equal	Yes	0.904	1.41	-17.4	22.0
OS-G-30	10L 50N	No	Norm. Dist.	Yes	0.905	1.34	-19.7	16.5

Table B.4: Performance metrics for orographic speedup investigations in % speedup. All at 100m AGL; input sub-grids of 1.2km square; single output point; consistent input sub-grid patterns; all using MM Elev., Elev. Grad. and Ro as inputs; all trained with data from 11 sites for 50 epochs.

Run ID	Architecture	Rotate?	Balancing	Batch Norm.	MAE	Error Std. Dev.	Min. Error	Max. Error
OS-G-31	10L 50N	No	-	Yes	1.89	3.09	-53.4	60.1
OS-G-32	Autoencoder	No	-	Yes	1.92	3.16	-53.1	60.4
OS-G-33	5L 10N	No	-	Yes	2.41	3.27	-69.6	55.3
OS-G-34	10L 50N	Yes	-	Yes	1.95	3.25	-73.8	90.3
OS-G-35	10L 50N	No	-	No	2.83	6.05	-130	101

Table B.5: Performance metrics for orographic speedup investigations in % speedup. All at 10m AGL; input sub-grids of 1.2km square; single output point; consistent input sub-grid patterns; all using MM Elev., Elev. Grad. and Ro as inputs; all trained with data from 11 sites for 50 epochs.

Run ID	Height AGL (m)	Validation / Test	MAE	Error Std. Dev.	Min. Error	Max. Error
OS-G-36	10	Validation	1.90	2.79	-51.3	48.4
OS-G-36	10	Test (New Sites)	3.68	7.04	-103	111
OS-G-37	100	Validation	0.652	1.04	-14.7	14.6
OS-G-37	100	Test (New Sites)	1.64	3.01	-31.1	23.8

Table B.6: Performance metrics for final Grid NN orographic speedup models in % speedup. All with sub-grids of 1.2km square; consistent input sub-grid pattern; single point output; inputs of MM Elev., Elev. Grad. & Ro; trained with data from 11 sites for between 50 and 60 epochs.

B.2 Orographic Turn

B.2.1 Grid Neural Networks

The tables of performance metrics for the validation data from each orographic turn Grid NN surrogate model described in Chapter 6 are given here.

Run ID	**Height AGL (m)**	**Site**	**Sub-Grid Side Length (km)**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
OT-G-1	10	1	Single point	1.58	2.29	-22.9	22.9
OT-G-2	10	1	0.8	1.27	1.77	-16.8	14.4
OT-G-3	10	1	1.2	1.19	1.66	-15.6	12.6
OT-G-4	10	1	1.6	1.36	1.78	-13.6	14.1
OT-G-5	10	1	3.2	1.21	1.78	-14.8	14.9
OT-G-6	10	2	Single point	1.60	2.26	-23.8	19.4
OT-G-7	10	2	0.8	1.11	1.61	-17.2	26.5
OT-G-8	10	2	1.2	1.29	1.71	-18.9	19.9
OT-G-9	10	2	1.6	1.49	1.99	-15.4	24.3
OT-G-10	10	2	3.2	1.30	1.77	-15.7	15.3
OT-G-11	100	1	Single point	0.672	0.963	-6.63	6.53
OT-G-12	100	1	0.8	0.592	0.756	-4.68	4.06
OT-G-13	100	1	1.2	0.485	0.668	-3.68	4.04
OT-G-14	100	1	1.6	0.481	0.692	-4.09	4.31
OT-G-15	100	1	3.2	0.454	0.693	-3.81	4.05
OT-G-16	100	2	Single point	0.506	0.734	-6.28	5.28
OT-G-17	100	2	0.8	0.458	0.597	-3.88	4.54
OT-G-18	100	2	1.2	0.385	0.545	-5.01	5.71
OT-G-19	100	2	1.6	0.430	0.598	-4.36	5.86
OT-G-20	100	2	3.2	0.443	0.5s86	-3.70	4.47

Table B.7: Performance metrics for orographic turn sub-grid size investigations in °. All using MM Elev., Elev. Grad., Ro. and normalised direction as an input variables; consistent DNN configurations.

Run ID	Height AGL (m)	Input Elev. Grad. @90°	Unit	Sub-Grid Side Length (km)	MAE	Error Std. Dev.	Min. Error	Max. Error
OT-G-21	10	No	°	1.2	0.993	2.88	-258	268
OT-G-22	10	No	°	1.6	0.955	3.03	-277	294
OT-G-23	10	Yes	°	1.2	1.48	3.44	-270	191
OT-G-24	10	No	rad	1.2	0.968	2.82	-286	262
OT-G-25	10	No	-	1.2	1.15	3.17	-185	182
OT-G-26	100	No	°	1.2	0.381	0.603	-8.85	13.8
OT-G-27	100	No	°	1.6	0.406	0.616	-9.44	8.82
OT-G-28	100	Yes	°	1.2	0.483	0.715	-10.1	8.42
OT-G-29	100	No	rad	1.2	0.372	0.624	-13.95	10.1
OT-G-30	100	No	-	1.2	0.401	0.619	-14.5	8.62

Table B.8: Performance metrics for orographic turn input and output investigations in °. Runs with "-" as the Unit predict $\frac{turn}{180}$. All using MM Elev., Elev. Grad., Ro. and normalised direction as inputs; consistent DNN configurations.

Run ID	Height AGL (m)	Validation / Test	MAE	Error Std. Dev.	Min. Error	Max. Error
OT-G-31	10	Validation	0.901	2.66	-281	155
OT-G-31	10	Test (New Sites)	1.39	3.11	-288	261
OT-G-32	100	Validation	0.397	0.635	-11.9	10.3
OT-G-32	100	Test (New Sites)	0.446	0.786	-9.77	9.50

Table B.9: Performance metrics for the final orographic turn Grid NN models in °. All using MM Elev., Elev. Grad., Ro. and normalised direction as an input variables; consistent DNN configurations; run for between 50 and 60 epochs.

B.3 Roughness Speedup

B.3.1 Convolutional Neural Networks

The tables of performance metrics for the validation data from each roughness speedup CNN surrogate model described in Chapter 7 are given here. In these CNNs, the input and output grids are rotated such that the wind direction is always from 0°.

Run ID	**Height AGL (m)**	**Inputs**	**Layers**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
RS-C-1	100	ln(1/Ro), ln(1/MR), RC	conv2d(k=5,c=10) batchnorm2d ReLU conv2d(k=3) batchnorm2d ReLU convtr2d(k=3) batchnorm2d ReLU convtr2d(k=5,c=1)	1.06	1.43	-7.51	11.7

Table B.10: Performance metrics for the initial roughness speedup CNN in % speedup (note that this model was trained with MAE loss).

Run ID	**Height AGL (m)**	**Prev. Run**	**Inputs**	**Layers**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**

RS-C-2	10	-	Ro,
			ln(Ro),
			ln(1/Ro),
			MR,
			ln(MR),
			ln(1/MR),
			Ro-MR,
			RC
				conv2d(k=5,c=8)
				ReLU
				conv2d(k=3)
				ReLU
				convtr2d(k=3)
				ReLU
				convtr2d(k=5,c=1)
					1.11	2.37	-29.9	20.8

RS-C-3	25	RS-C-2	As RS-C-2	As RS-C-2	0.923	1.96	-29.4	16.2
RS-C-4	100	RS-C-2	As RS-C-2	As RS-C-2	0.862	1.53	-29.0	16.2

Table B.11: Performance metrics for roughness speedup transfer learning investigations in % speedup. Using data from Site 1 only.

Run ID	Height AGL (m)	Inputs	Layers	MAE	Error Std. Dev.	Min. Error	Max. Error

RS-C-5	10	Ro,
		ln(Ro),
		ln(1/Ro),
		MR,
		ln(MR),
		ln(1/MR),
		Ro-MR,
		RC
			conv2d(k=5,c=8)
			ReLU
			conv2d(k=3)
			ReLU
			convtr2d(k=3)
			ReLU
			convtr2d(k=5,c=1)
				2.92	4.98	-30.3	41.1

RS-C-6	10	As RS-C-5	conv2d(k=5,c=8) batchnorm2d ReLU conv2d(k=3) batchnorm2d ReLU convtr2d(k=3) batchnorm2d ReLU convtr2d(k=5,c=1)	1.72	2.98	-33.7	33.3

Table B.12: Performance metrics for roughness speedup batch normalisation CNN test, in % speedup. Using data from all 19 sites.

Run ID	**Added Filter**	**Inputs**	**Layers**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
RS-C-6	-	Ro, ln(Ro), ln(1/Ro), MR, ln(MR), ln(1/MR), Ro-MR, RC	conv2d(k=5,c=8) batchnorm2d ReLU conv2d(k=3) batchnorm2d ReLU convtr2d(k=3) batchnorm2d ReLU convtr2d(k=5,c=1)	1.72	2.98	-33.7	33.3
RS-C-7	15x15	As RS-C-6	conv2d(k=15,c=8) batchnorm2d ReLU conv2d(k=5) batchnorm2d ReLU conv2d(k=3) batchnorm2d ReLU convtr2d(k=3) batchnorm2d ReLU convtr2d(k=5,c=1)	1.82	2.54	-31.3	34.0
RS-C-8	51x51	As RS-C-6	conv2d(k=51,c=8) batchnorm2d ReLU conv2d(k=5) batchnorm2d ReLU conv2d(k=3) batchnorm2d ReLU convtr2d(k=3) batchnorm2d ReLU convtr2d(k=5,c=1)	1.53	2.54	-26.1	39.9

Table B.13: Performance metrics for roughness speedup CNNs with additional filters of varying sizes, in % speedup. Using data from all 19 sites.

B.3.2 Grid Neural Networks

The tables of performance metrics for the validation data from each roughness speedup Grid NN surrogate model described in Chapter 7 are given here.

Run ID	**Input**	**Sub-Grid Size (km)**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
RS-G-1	Ro.	2.2	2.06	2.03	-6.24	1.00
RS-G-2	MR	2.2	0.592	0.653	-2.49	2.23
RS-G-3	RC	1.8	0.134	0.265	-2.07	2.09

Table B.14: Performance metrics for roughness speedup Grid NN sub-grid size investigations with single variable inputs, in % speedup. All for Site 1 at 100m AGL; normalised direction input; consistent sub-grid patterns; consistent DNN configurations.

Run ID	**Height (m AGL)**	**Input Sub-Grid Size (km)**	**Output Sub-Grid Size (km)**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
RS-G-4	10	1.2	Single Point	0.437	0.781	-10.8	6.84
RS-G-5	10	2.2	Single Point	0.960	1.35	-6.81	10.4
RS-G-6	10	3.2	Single Point	1.45	1.71	-12.8	4.13
RS-G-7	10	1.2	0.2	0.482	0.681	-11.3	8.11
RS-G-8	10	1.2	0.6	0.438	1.09	-12.6	10.2
RS-G-9	100	1.2	Single Point	0.543	0.255	-2.71	2.08
RS-G-10	100	2.2	Single Point	0.550	0.576	-1.08	3.21
RS-G-11	100	3.2	Single Point	0.279	0.323	-1.23	0.170
RS-G-12	100	3.2	0.2	0.256	0.200	-1.80	0.545
RS-G-13	100	3.2	1.6	0.241	0.354	-1.16	1.60

Table B.15: Performance metrics for roughness speedup Grid NN input and output sub-grid configuration investigation, in % speedup. All with ln(1/Ro), ln(1/MR), ln(1/Ro)-ln(1/MR), RC and normalised direction as inputs; consistent DNN configurations.

Run ID	Architecture	MAE	Error Std. Dev.	Min. Error	Max. Error
RS-G-14	5L 10N	0.418	0.842	-9.19	10.3
RS-G-15	10L 50N	0.369	0.844	-11.1	9.24
RS-G-16	10L 100N	0.339	0.776	-10.6	9.61

Table B.16: Performance metrics for roughness speedup Grid NN network size tests, in % speedup. All at 100m AGL; with ln(1/Ro), ln(1/MR), ln(1/Ro)-ln(1/MR), RC and normalised direction as inputs; input sub-grid of 3.2km square, consistent sub-grid pattern; single point output.

Run ID	Dropout Probability	Weight Decay Factor	MAE	Error Std. Dev.	Min. Error	Max. Error
RS-G-15	0%	0.0	0.369	0.844	-11.1	9.24
RS-G-17	10%	0.0	0.517	0.912	-11.6	9.26
RS-G-18	25%	0.0	0.467	0.820	-8.65	9.33
RS-G-19	0%	0.1	4.71	1.54	-10.1	7.23
RS-G-20	0%	0.25	11.0	1.54	-16.5	0.877

Table B.17: Performance metrics for roughness speedup Grid NN anti-overfitting technique tests, in % speedup. All at 100m AGL; with ln(1/Ro), ln(1/MR), ln(1/Ro)-ln(1/MR), RC and normalised direction as inputs; input sub-grid of 3.2km square, consistent sub-grid pattern; single point output.

Run ID	Description	Inputs	Layers	MAE	Error Std. Dev.	Min. Error	Max. Error

RS-G-15	Baseline Grid NN	ln(1/Ro)
		ln(1/MR)
		ln(1/Ro)-ln(1/MR)
		RC
			10L 50N	0.369	0.844	-11.1	9.24

RS-G-21	Grid NN Input to CNN (2.4km Input Sub-Grid)	ln(1/Ro)
		ln(1/MR)
		RC
			10L 50N	0.330	0.721	-10.2	9.88

RS-C-9	CNN	RS Ro. MS Ro-MR RC	conv2d(k=5,c=5) batchnorm2d ReLU conv2d(k=3) batchnorm2d ReLU convtr2d(k=3) batchnorm2d ReLU convtr2d(k=5,c=1)	1.47	2.01	-19.8	22.2

Table B.18: Performance metrics for roughness speedup Grid NN-CNN model tests, in % speedup. "RS" represents the roughness speedup from the Grid NN RS-G-15 input to run RS-C-9. All at 100m AGL; the input and output grids for the CNN were not rotated; the Grid NN models used normalised direction as an input.

Run ID	**Height AGL (m)**	**Validation / Test**	**MAE**	**Error Std. Dev.**	**Min. Error**	**Max. Error**
RS-G-22	10	Validation	0.768	1.45	-23.0	24.1
RS-G-22	10	Test (New Sites)	0.472	1.16	-18.2	25.2
RS-G-23	100	Validation	0.517	0.980	-10.8	8.80
RS-G-23	100	Test (New Sites)	0.384	0.953	-9.91	11.7

Table B.19: Performance metrics for the final roughness speedup Grid NN models in % speedup. All using ln(1/Ro), ln(1/MR), ln(1/Ro)-ln(1/MR), RC and normalised direction as an input variable; consistent DNN configurations.