IMPERIAL COLLEGE LONDON

Department of Mechanical Engineering
Advanced Mechanical Engineering Course

A Flexible and Lightweight Deep Learning Weather Forecasting Model

by Gabriel Zenkner

Master of Science, Advanced Mechanical Engineering, Imperial College London

A thesis submitted in partial fulfilment of the requirements for the Master of Science Degree of Imperial College London and the Diploma of Membership of Imperial College

9th September 2022

Abstract

Numerical weather prediction is an established weather forecasting technique in which equations describing wind, temperature, pressure and humidity are solved using the current atmospheric state as an input. Numerical weather prediction has improved steadily since the 1960s, but not without enormous investment. These simulations demand tremendous computational resources and are exceedingly expensive. Weather behaves chaotically and to reduce uncertainty, expensive ensemble modelling is used, in which a model is run many times with small differences in initial conditions. The Met Office is set to spend £1.2 billion on a supercomputer to address growing computational demands.

The use of deep learning to predict weather is investigated, which exploits patterns in historical data to make predictions as opposed to simulating the physics. A rise in open-source software for implementing complex deep learning architectures and the increasing availability of data and hardware, such as graphical processing and tensor processing units, has accelerated the use of deep learning to solve problems traditionally modelled by physics simulations. Compared to numerical weather prediction simulations run as frequently as four times a day, machine learning models can be trained once and can make rapid predictions; often within seconds.

This study examines deep learning to forecast weather given historical data from two London-based locations. Two distinct Bi-LSTM recurrent neural network models were developed in the TensorFlow deep learning framework and trained to make predictions. The Met Office publicises their 24-hour air temperature forecast accuracy as ±2°C in 92.5% of instances. Compared to this, the first neural network predicted temperature at Kew Gardens to the same accuracy in 72.9% of instances. The network was trained with a rented graphical processing unit costing £116 while the Met Office 2021 operations budget was £256 million. Therefore, a 26.9% increase in accuracy comes at approximately 2.2 million times the cost of the deep learning model.

The second network predicted 72-hour air temperature and relative humidity at Heathrow with root mean squared errors 2.26°C and 14% respectively. 79.5% of the temperature predictions were within ±3°C while 80% of relative humidity predictions were within ±20%. For comparison, the 24-hour model was trained in 78 seconds while the 72-hour model was trained in 187 seconds with temperature root mean squared errors of 1.45°C and 2.26°C respectively.

Acknowledgements

I would like to thank Dr Salvador Navarro-Martinez for his continuous help, advice and mentoring. His constant support, expert knowledge and enthusiasm made this project possible.

I would also like to thank the project's co-supervisor Dr Stelios Rigopoulos and Dr Rosella Arucucci for their valuable input.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Weather affects many aspects of human life and activities. Numerous sectors are heavily reliant on accurate weather forecasting including renewable energy production, energy consumption, agriculture, emergency services, aviation, retail, and recreation. An improved understanding of the atmosphere and exploration of novel techniques for weather prediction will not only benefit industry, but will provide better tools to respond to severe weather and develop strategies to minimise the impact of climate change.

Figure 1.1: Improving weather predictions will enable the growing renewable energy sector to optimise energy generation and inform investment decisions [1]

Wind and solar energy generation are dependant on weather conditions and would each benefit from improved wind and cloud-cover predictions. For instance, Sun et al. [16] recognise the current limitations in predicting wind behaviour due to its stochastic nature and strong local environment effects. Meenal et al. [17] highlight the planning challenges with renewable energy generation as they become more prevalent. Specifically, photovoltaic panels and wind turbines produce energy intermittently and better forecasting tools are needed to inform infrastructure and operational decisions.

There is a movement within agriculture from a productivity- to a sustainability-based approach and a need for enhanced weather understanding and forecasting to improve decision-making processes to support sustainability within the sector. This is especially true as the world's population is expected to exceed nine billion inhabitants by 2050 [18]. The ability to precisely predict rainfall, temperature and relative humidity several days into the future would enable farmers to better plan their activities such as planting and harvesting.

Severe and unpredictable weather causes significant damage every year through flooding, storm surges and hail. Fu et al. [19] report that in 2015, adverse weather resulted in damages amounting to $7.9 billion. Improvements in weather forecast accuracy would enable emergency services to proactively respond to storm surges [20]. Accurately preempting these phenomena will minimise their impact and result in reduced damage to infrastructure and loss of human life.

Figure 1.2: Flash flooding and dry spells have enormous social and economic implications [2]

Today, weather evolution is simulated using numerical weather prediction (NWP). The concept of simulating atmospheric conditions originated with Lewis Fry Richardson nearly a century ago. An improved understanding of thermodynamics and fluid mechanics enabled meteorologists to couple partial differential equations describing atmospheric behaviour, and solve them to predict weather. In modern numerical weather prediction, the domain of interest is discretised into a three-dimensional grid and the governing equations are solved [21] with the prediction accuracy increasing slowly, but steadily over the last four decades [22]. This is primarily driven by improvements in computational power, data acquisition, data assimilation and ensemble modelling. However, numerical weather simulations come at a remarkable cost with the Met Office recently reporting a £1.2 billion investment on a supercomputer [9] while the National Weather Service will spend up to $500 million on supercomputers for adverse weather prediction [23]. Additionally, simulations can take several hours to run and consume large amounts of energy. Supercomputer energy consumption accounted for the majority of the £7.63 million Met Office energy expenditure in 2021 [24].

In contrast to solving atmospheric equations, a data-driven approach seeks to use statistical or machine learning models to infer patterns from historical weather data. Weather prediction

has been attempted with computationally lightweight statistical models such as auto regression integrated moving average or traditional machine learning algorithms such as linear regression, support vector machines and random forests. However, it is neural networks that have garnered the most interest in weather forecasting because they can model non-linearity [12]. Furthermore, the accuracy of predictions has been shown to improve when using neural networks compared to traditional machine learning, provided large amounts of data are available. Roy [25] has evaluated a multilayer perceptron, a long short-term memory (LSTM) model and a hybrid convolutional neural network LSTM model and concludes that models with more complex architectures improve performance. Ravuri et al. [26] recently demonstrated their neural network model can predict precipitation more accurately in 89% of instances compared to existing weather prediction techniques. Hewage et al. [27] report that their neural network models predict weather conditions 12 hours into the future with higher accuracy than conventional weather forecasting. Bauer et al. [4] recently showed their convolutional neural network ensemble forecasting model can predict anomalies like Hurricane Irma, which occurred in 2017.

This study aims to design a neural network model to make short-term 24-hour air temperature predictions, which will demonstrate proof of concept and serve as a benchmark. Following this, a medium-term model was developed with the aim of forecasting air temperature, humidity, and wind speed over 72 hours. We study the effects of using data from multiple locations, the impact of weather parameter selection and the model setup on forecast skill. The novelty of our approach to data-driven weather forecasting is characterised by a combination of several components. Firstly, our models are incredibly lightweight, with training taking 78 and 187 seconds using publicly-available datasets from the Met Office, and are capable of generating new forecasts within seconds. The neural networks have been designed for maximum flexibility to enable predictions of any weather parameters contained within the original dataset, such as wind speed or dew point temperature. Furthermore, these models can produce forecasts of any length using any length of sequential data as an input. Perhaps most importantly, the models cost £116.62 to develop.

Chapter 2: The Background discusses the state of the art of numerical weather prediction, how it has evolved and how it is expected to evolve. Furthermore, the limitations of conventional weather prediction systems are reviewed. Existing data-driven approaches are investigated and the benefits and limitations are discussed along with a summary of candidate neural networks. **Chapter 3**: The Methodology discusses the architecture of a deep feed forward network and a recurrent neural network, both of which are used in forecasting. Then, the data is analysed and processed to ensure compatibility with these models. **Chapter 4**: Each model's ability to predict on various temporal scales is assessed in the Results and Discussions. **Chapter 5**: The key findings from the results are summarised and the areas for future work are identified in the Conclusions.

Chapter 2

Background to Weather Prediction

2.1 Numerical Weather Prediction

Initially conceived in the early 1900s, numerical weather prediction (NWP) is the gold standard for weather forecasting and first found practical application in the 1950s [22]. While numerical weather prediction accuracy has has increased gradually since the 1950s, this improvement has not come without significant financial investment.

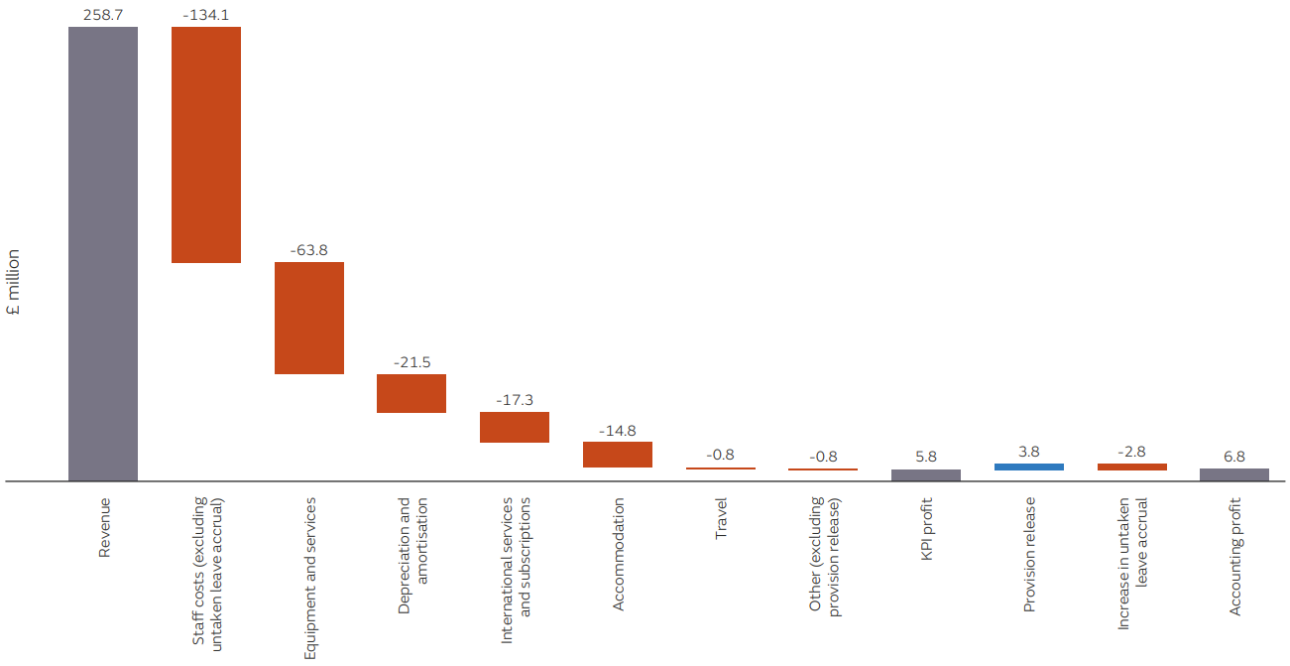

Figure 2.1: The Met Office had an annual revenue target of £258.7 million to offset their operational expenditures [3]

This is exemplified by the Met Office budgeting £1.2 billion for a supercomputer in 2020 [28]. According to their 2021 Accounts Report, the Met Office accrued £258.7 million in

annual operational expenses which does not include R&D and large infrastructure expenditures [3]. While it is difficult to quantify the cumulative expenses of a single organisation, weather prediction is a global initiative and weather agencies rely heavily on international collaboration to improve their predictions, which adds additional cost. Despite these enormous costs, weather prediction is indispensable in our modern society and the benefits far outweigh the investment.

Fully representing atmospheric conditions is challenging and is currently a bottleneck in weather predictions. It is impossible to capture every relevant detail of the atmospheric state, which is needed as input for initial and boundary conditions. The lack of a complete description of the atmospheric state leads to uncertainty in the model. In spite of these challenges, the accuracy of weather predictions for 3-, 5-, 7- and 10-day forecasting products has improved steadily since 1981 as depicted by the European Centre for Medium-range Weather Forecasts (ECMWF) in Figure 2.2. The disparity between accuracy in the Southern and Northern Hemisphere has been reduced in the last two decades due to the availability of satellite data. Three-day forecasts are now made with high confidence while confidence deteriorates significantly for 10-day forecasts.

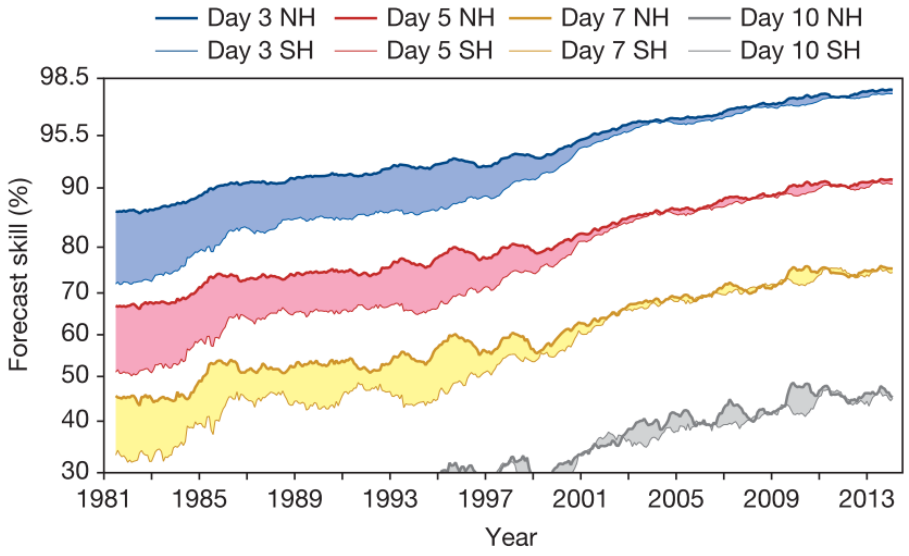

Figure 2.2: 500-hPa, roughly 5500m, forecast skill has increased gradually over the last four decades and creates a benchmark for forecasting at other altitudes [4]

2.1.1 Data Acquisition and Assimilation

Bjerknes postulated in 1904, that in order to predict the future state of the atmosphere and weather, the current state of the atmosphere needs to be fully defined [6]. He believed describing the evolution of the atmospheric state was achievable with the primitive equations, namely conservation of momentum, energy, mass, and water. At the time, only mechanical computers were available, which were too slow to produce timely results. It was acknowledged that three-dimensional data acquisition was needed to express the required boundary conditions, however, weather could only be measured at ground level at the time [29].

In modern weather forecasting, sensor data describing the state of the atmosphere are acquired by a host of diverse systems, including land and sea-based weather observation stations, radiosondes, RADAR, aircraft and satellite imagery, as depicted in Figure 2.3. The radiosonde, a sensor that communicates by radio and is attached to a weather balloon, revolutionised weather predictions as it enables weather acquisition at different vertical levels. Another milestone was data acquisition over the oceans in 1960s via satellites and buoys. Previously, meteorologists used a flux adjustment that approximated moisture and heat behaviour over the oceans. These advancements provided more information on humidity and solar flux [22].

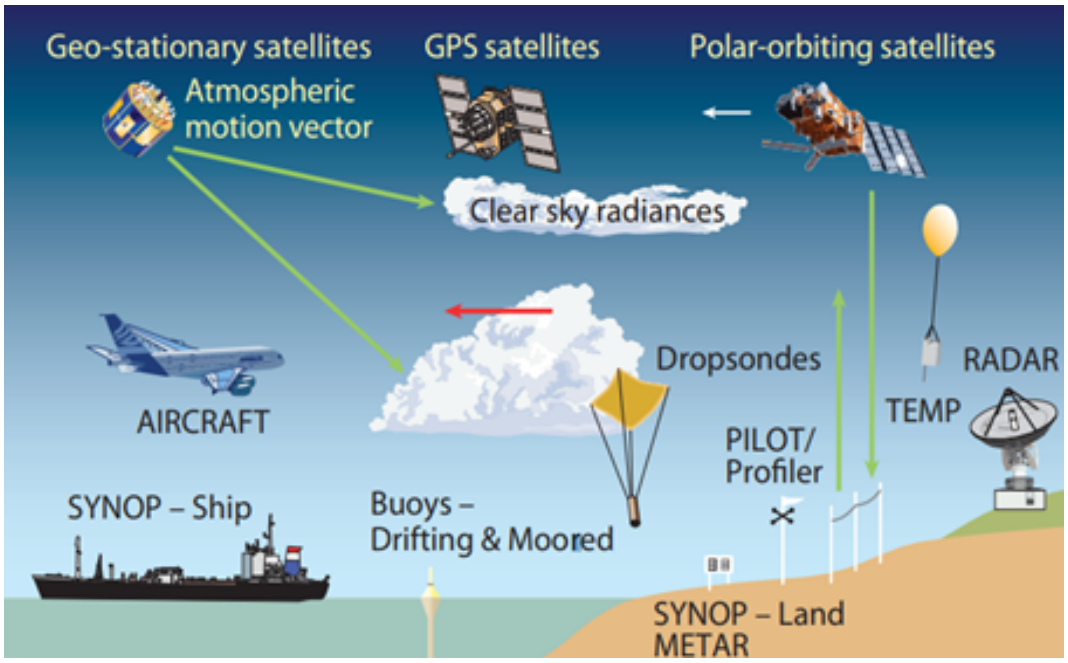

Figure 2.3: Modern weather prediction relies on numerous data sources to describe the atmospheric state [5]

In spite of our rich and diverse ecosystem of data acquisition systems and their ubiquity, we are still unable to describe the atmosphere in sufficient detail. Data assimilation is a technique used to approximate the atmospheric state using both observations and predictions as input [25], which in turn are used to define initial and boundary conditions for simulations. The objective of data assimilation is to translate these data that are distributed heterogeneously in time and space and represent them in a mathematical form a computer can interpret. The ultimate goal is to create a sufficient representation of the physical world and atmosphere from a finite number of data sources [30]. Ensemble Kalman filtering is a technique ubiquitous in data assimilation for noise processing and is used to address measurement and process uncertainty. It works by removing unnecessary information and randomly passing data in a sequence to the model. [31]. However, the algorithm relies heavily on selecting the correct parameters in order for the filter to generate meaningful results [32].

2.1.2 Physics Model

The physics and equations that describe the behaviour of the atmosphere have been known since the 1800s and are known as the primitive equations. These equations are conservation of momentum (2.1), continuity (2.2), the ideal gas law (2.3), conservation of energy (2.4) and conservation of water undergoing change of state (2.5), and are expressed in spherical coordinates. There are seven variables, which when solved for, give air velocity in three dimensions, temperature, density, pressure and humidity, These governing equations are expressed as

$$\rho \frac{D\vec{v}}{Dt} = -\vec{\nabla} p + \nabla \cdot \tau + \rho \vec{f} + \rho \vec{g} \tag{2.1}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{2.2}$$

$$p = \rho R T \tag{2.3}$$

$$Q = C_p \frac{dT}{dt} - \frac{1}{\rho} \frac{dp}{dt} \tag{2.4}$$

$$\frac{\partial (\rho q)}{\partial t} = -\nabla \cdot (\rho q \vec{v}) + \rho (E - C) \tag{2.5}$$

where $\rho \vec{f}$ is the Coriolis term, E and C are evaporation and condensation rates respectively and q is the water vapour mixing ratio [4, 6, 30].

At the onset, solving these equations may seem elementary. It is when the dynamics of weather is considered, the non-linearity, chaotic behaviour and the sheer scale of modelling, that the challenges become apparent. Turbulence is unsteady in time and makes solving the momentum equation very difficult. It is necessary to discretise these into finite-difference equations to approximate the partial differential equations [6]. This set of nonlinear differential equations must be solved numerically with up to half a billion calculations per each timestep computed. [4]. Chaotic tendencies imply that small variations in initial conditions produce very different outcomes. It is this chaotic behaviour that has a profound impact on the length and accuracy of forecasts, as well as their viability. Firstly, predictions beyond two weeks deteriorate significantly because the initial conditions are not represented adequately in the model. Secondly, ensemble models can be run to calculate an average of the ensemble thereby reducing the effects of weather's chaotic behaviour, but quickly become a limiting factor due to their enormous computational demands.

The energy cascade, described as motion and energy transfer from the largest to smallest scales, is of particular significance when considering weather behaviour. The disparity in length and timescales makes predictions more challenging and requires simplifications to be made [33]. Physical processes occur on various spatial scales and include radiative heating, heat dissipation, friction, condensation, and evaporation. As it is impossible to model these phenomena on their corresponding scales, they are generally approximated and expressed in terms of fluxes [4].

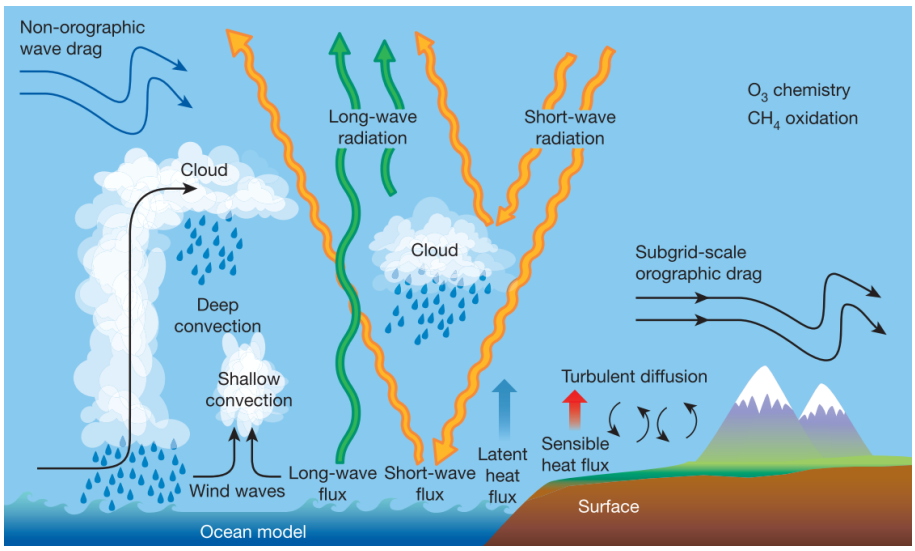

Figure 2.4: Physical phenomena such as radiation, convection and diffusion have a significant effect on weather and are introduced into the physics model via simplified flux terms [4]

Once the data have been captured and assimilated and the physics model defined, spatial and temporal discretisation over the region of interest is performed and a three-dimensional grid generated so computers can solve the coupled, non-linear equations at discrete points. Weather conditions such as humidity, temperature, satellite imagery, heat transfer and surface hydrology define the initial and boundary conditions for the meteorological model [27].

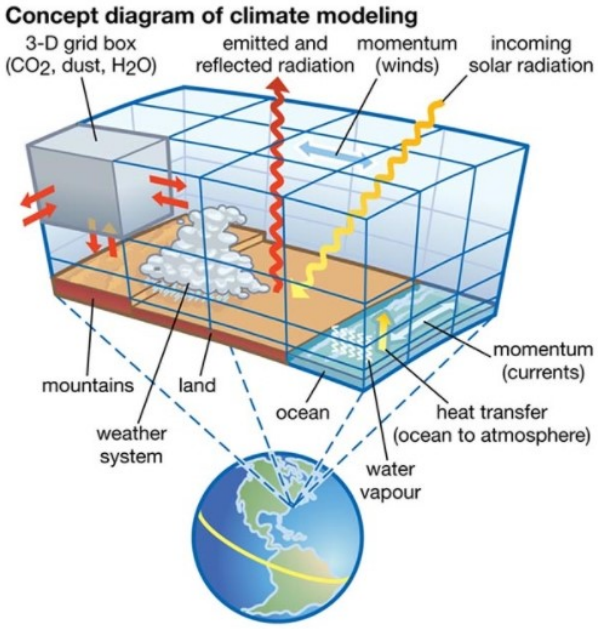

Figure 2.5: The domain is discretised into Cartesian or spherical coordinates and boundary conditions imposed at the upper and lower boundaries [6]

The size of the cells within the domain must be carefully considered. Compared to high resolution grids, large cells will introduce more errors into the modelling which may grow and become unstable. High resolution grids, such as those on the 4km and 1km scale, were

shown to improve precipitation contour predictions compared to larger grids [34]. Climate models are used to understand global climate trends and require global inputs resulting in very large models. Slingo and Palmer [10] highlight the current climate grid sizes of 100km are insufficient for modelling phenomena taking place on a smaller scale, and still result in significant uncertainty for these long-term predictions. While a higher resolution discretisation scheme would alleviate some of these issues, the limiting factor is computational time and resources.

According to Chantry et al. [35], short term forecasting is characterised by predictions up to 48 hours with this timescale corresponding to the development of a weather front or storm. Medium term forecasting is loosely defined as periods ranging from several days to several weeks while sub-seasonal forecasting is described as the period ranging from several weeks to the length of seasons. The Meteorological Office (Met Office) has developed a series of models for prediction across various temporal and spatial scales as seen in Figure 2.6 and are collectively known as the Unified Model. The Met Office uses ensemble modelling to repeatedly run simulations, each time with a small change to the initial conditions, for predictions beyond five days. Averaging the results of the predictions has been shown to improve forecast accuracy while the distribution of the ensemble indicates the uncertainty in the forecast. Ensemble modelling is necessitated due to chaotic effects, which become increasingly dominant beyond five days [9].

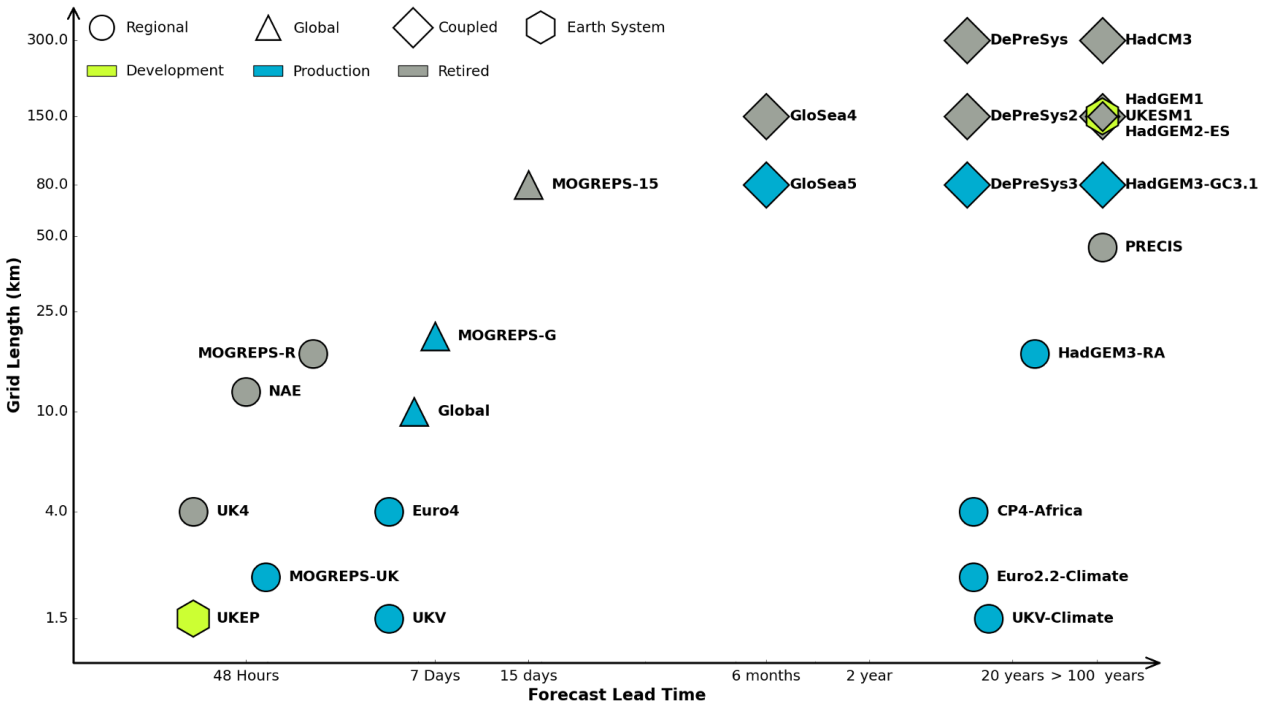

Figure 2.6: In the Unified Model, the grid length defines the spatial resolution of the model and together with the forecast period affect the accuracy and computation time [7]

One example of a model is the UKV-Climate model, which has a grid length of 1.5km and

is used to predict climate. It uses three-dimensional variational assimilation (3D-Var) and is deterministic, meaning it is not run as an ensemble, but as a single simulation [36]. Variational data assimilation is a technique used to account for uncertainty within the data used in the predictive model [4]. Another example is the MOGREPS-G ensemble model with a resolution of 25km, which is initialised every six hours and takes between 4-6 hours to complete a simulation. The initialisation is created through four-dimensional variational assimilation (4D-Var) corresponding to three-dimensional space and time [4]. 3D-Var has a point based, static assimilation window whereas 4D-Var uses a dynamic assimilation window which accounts for changes in time [30]. However, this comes at a significantly higher cost than 3D-Var. While 4D-Var has been shown to be superior to 3D-Var, there are substantial improvements that can be made to 3D-Var [37].

The Unified model is one of several major weather prediction models globally. Hewage et al. [27] report that the Weather Research Forecasting model is the most ubiquitous weather forecasting model and has seen advances in recent years driven primarily by the availability of high-performance computational resources and a wide community. It is a mesoscale model, ranging from tens of meters to thousands of kilometres, launched in 2000 by the National Centre for Atmospheric Research and is used in 162 countries [38].

The European Centre for Medium-Range Weather Forecasting (ECMWF) model uses a 25km grid with 91 vertical levels, uses 4D-Var and can make predictions from several days to several months. The ECMWF model can make one-month and six-month forecasts, both of which use the Ensemble Prediction System [22]. The ECMWF has the objective of creating global scale predictions on the 5km scale by 2025 [8]. The ECMWF also makes use of the increasing availability of satellite information and have a model capable of making a 10-day deterministic forecast; this is less computationally intensive than ensemble modelling.

2.1.3 Uncertainty and Probabilistic Forecasting

Quantifying uncertainty is a key component of weather predictions. The uncertainty arises from discrepancies in the description of the atmospheric state. Numerical models are sensitive to initial conditions arising from the chaotic behaviour of weather and turbulence. In ensemble modelling, a model is run numerous times with marginally different initial conditions resulting in various outcomes, as exemplified in Figure 2.7. A probability distribution describes the initial distribution of the input measurement, and from this, the outcomes are simulated and often exhibit highly divergent behaviours.

Ensemble modelling was first used in 1992 when it was deployed by the ECMWF [31]. The precursor to ensemble modelling was single deterministic weather predictions. It has been shown that the average of the ensembles is more accurate than any single prediction. The ECMWF developed the Ensemble Prediction System, which simulates 50 different outcomes

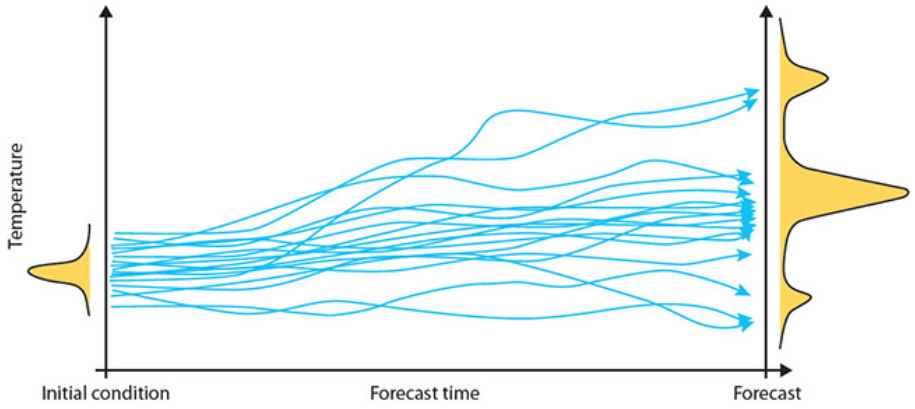

Figure 2.7: Uncertainty is quantified through ensemble modelling, whereby the initial atmospheric conditions may expressed as a distribution to describe uncertainty in the input measurements, with the effect of initial perturbations on the forecasts observed [8]

and their resultant probability distributions [39]. However, ensemble modelling comes at a significant cost in computational resources [40]. Presently, it is impossible to create a weather forecasting system that can perfectly model weather evolution as it is not feasible to observe every aspect of the initial state of the weather nor perfectly predict the chaotic and turbulent behaviour of weather [9], even if computational resources were not a limiting factor.

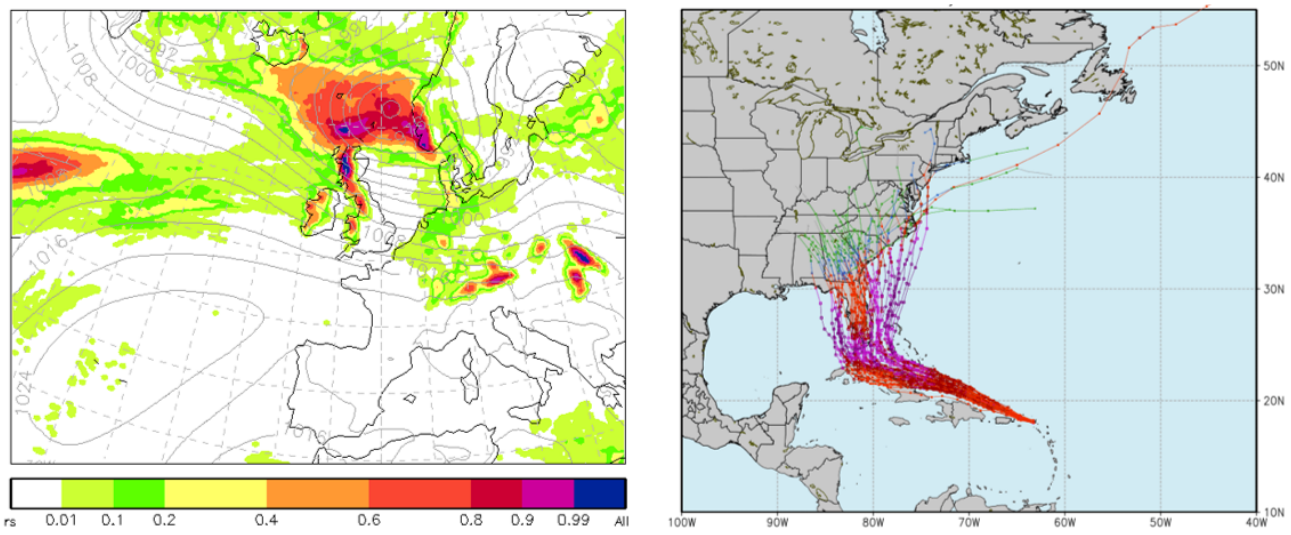

Figure 2.8: Left: Weather uncertainty is presented geographically by a colour scheme indicating the probability of 10mm of precipitation occurring within 24 hours [9]. Right: ECMWF typhoon ensemble modelling predicts dozens of possible locations where a typhoon may encounter land [8]

In the UK, the Met Office Global and Regional Ensemble Prediction System (MOGREPS) is used to calculate uncertainty. The regional system, MOGREPS-UK simulates on a 2.2km wide grid with 70 vertical levels while the global system, MOGREPS-G, simulates ensemble predictions on a global scale with a much lower resolution of 25km [9]. The simulation accuracy is limited by the available computer infrastructure and the ability to compute and produce products within reasonable amounts of time. Numerical models can take from 15 minutes to

several hours per simulation. While this does not pose an issue for long-term predictions that are significantly longer than the simulation time, these run times make short-term prediction impractical in certain situations.

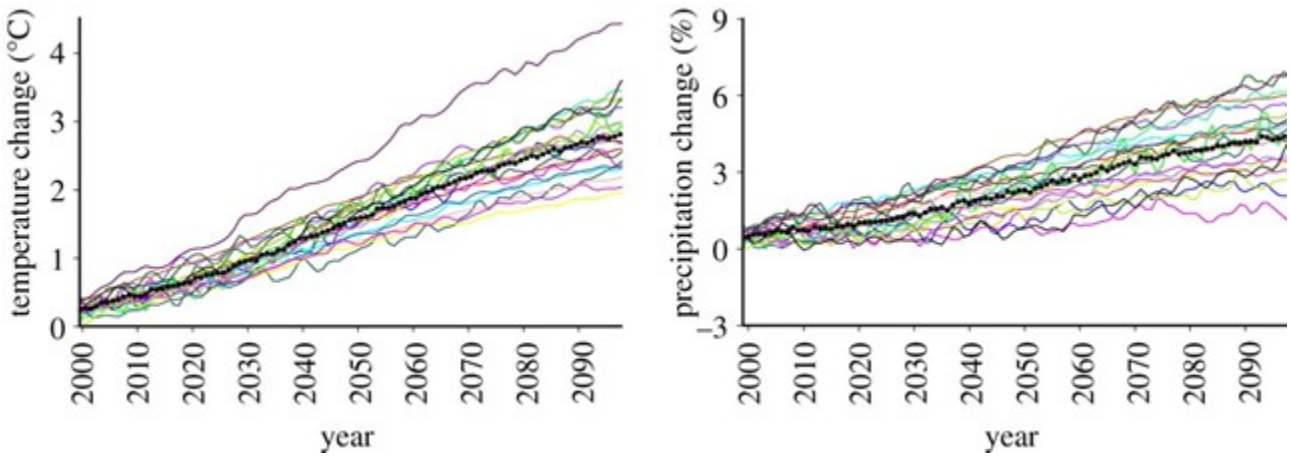

Figure 2.9: There is great uncertainty in climate change forecasts up to 100 years that arises from a misrepresentation of the atmospheric state and physics model [10]

Decadal forecasts are possible, but carry with them considerable uncertainty as seen in Figure 2.9, such as the UKV-Climate model. These represent changes to our climate over nearly a century with an uncertainty exceeding ±50%. Slingo and Palmer [10] concluded that misrepresentation of the initial conditions is not the only source of uncertainty in model predictions. The assumption that other factors, aside from the misrepresentation of the atmosphere, affect the outcome formed the foundation for the Lorenz Model and the role of chaos. Additionally, simplifications to the physics model been shown to contribute to the overall uncertainty [31].

2.1.4 Challenges in Numerical Weather Prediction

One of the biggest challenges is meeting the ever-increasing computational demands of evolving models and data assimilation. This is exemplified by the upgrade to a £1.2 billion supercomputer from the present £97 million supercomputer [28]. Chattopadhyay et al. [31] identifies calculation of the adjoint matrix in variational data assimilation as a significant challenge. This is due to the non-linearity and high dimensionality of the governing equations. The ensemble data assimilation technique negates the need to calculate the adjoint, but the ensemble approach is more expensive than the deterministic approach. Ensemble modelling has now become a mainstream technique for assessing uncertainty in models. However, it is computationally demanding requiring numerous reruns of each model with different initial conditions, often as many as 50. Rasp et al. [21] suggest that to make meaningful seasonal predictions, the number of runs should be between 100 and 200. Bauer et al. [4] note that precipitation is highly unpredictable and arguably one of the most important parameters in weather and that reducing the uncertainty of rainfall would require further ensembles.

In addition to ensemble modelling, Bauer et al. [4] highlight the process of data assimilation, whereby the initial state is approximated from observational data and predictions, as an area ripe for improvement. Furthermore, they note that physical phenomena are likely to be misrepresented by these simplifications. Physical phenomena on the molecular scale and chemical processes and the relationships describing them are simply not present in any models. Instead, observations are made on the macroscale and expressed as fluxes.

Rihan [30] cites acquisition of representative initial conditions as one of the biggest hurdles in numerical weather prediction. This is because it is impossible to characterise every aspect of the atmospheric state. This characterisation process becomes increasingly challenging in cities where the landscape drastically affects wind and temperature behaviour. This is exemplified by the Grenfell Tower Fire in 2017 where the nearest weather observation station at Kew Gardens recorded zero wind speed between 04:00 - 05:00 UTC, while visual reports of the smoke plume rising from the tower indicate a strong downwind during this period which caused the fire to spread more rapidly leading to a devastating and tragic outcome.

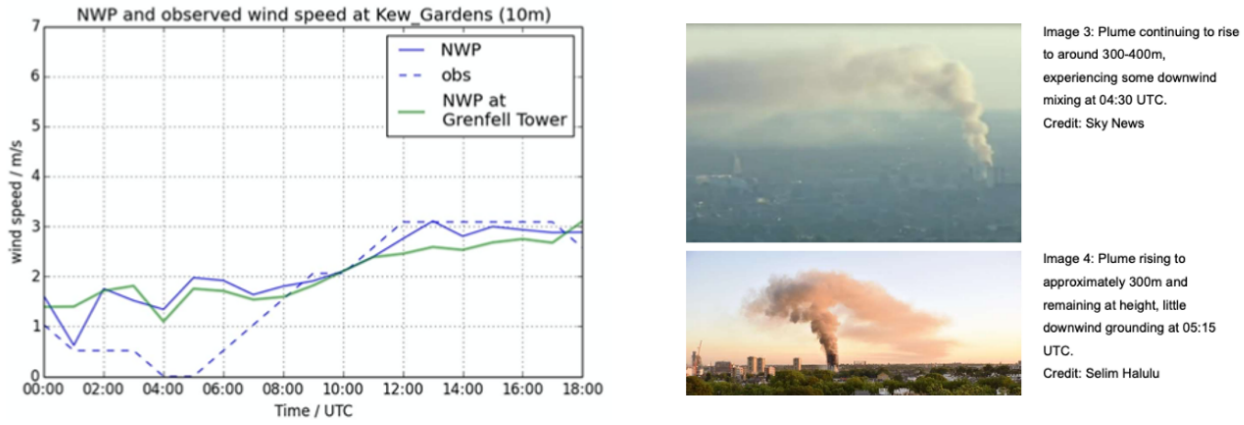

Figure 2.10: The observed wind speed between 04:00 and 05:00 UTC at nearby Kew Gardens was not aligned with the wind speed at Grenfell as indicated by the fast-moving smoke plume [11]

The amount of data captured, processed, and stored for numerical weather prediction is enormous and poses a significant challenge. Handling and manipulating these data comes at a considerable cost. Numerical weather prediction requires the most powerful computers available and a vast investment in infrastructure and maintenance running in the tens of billions of dollars [29]. Bauer et al. [4] state current data assimilation techniques are required to process approximately 10^7 observed data points per day while the ECMWF performs approximately 650 million spatial point calculations daily.

The question arises if a machine learning approach can complement existing numerical weather prediction, or perhaps even substitute it, thereby reducing the enormous computational demands and operational costs associated with numerical weather prediction.

2.2 Deep Learning for Weather Prediction

2.2.1 Deep Learning

In contrast to a physics-based approach, there is no explicit knowledge of the physical processes behind weather evolution in a data-driven approach. **Machine learning** (ML) is a technique that applies data to generate predictions and is considered a form of artificial intelligence. **Deep learning** (DL) is a subset of machine learning and is commonly performed with a **deep neural network** (DNN), that is, a **neural network** (NN) with multiple hidden layers. These additional hidden layers have been shown to enhance model accuracy by improving the model's ability to interpret increasingly complex data relationships and structures. Neural networks are mathematical models inspired by biological neurons and are comprised of nodes that, as individuals, perform simple addition or multiplication operations. However, it is harnessing the power of dozens, hundreds or even thousands of neurons and forcing them to interact in a meaningful way that results in a powerful prediction model capable of learning the subtlest relationships within a dataset.

Designing the architecture within the hidden layer is perhaps the single biggest challenge in deep learning and is highly application-specific. While neural network architectures vary considerably, they all have an input layer, output layer and one or more hidden layers as exemplified in Figure 2.11. The number of layers and nodes in each layer depends on the type and number of parameters, whether the data is sequential, the complexity of the problem and the type of output, whether regressive or categorical. A **feed forward neural network** (FNN) is one where information flows in a single direction, namely from the input to the output layer. The simplest deep neural network is a feed forward network with multiple hidden layers.

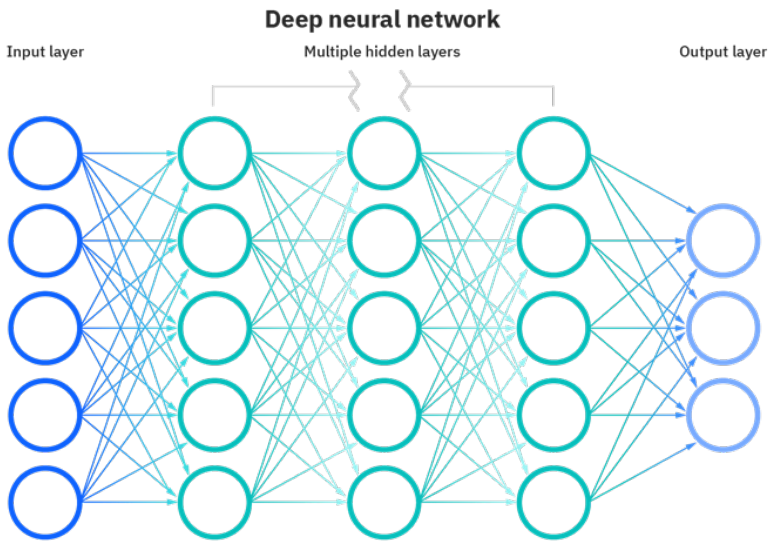

Figure 2.11: The simplest deep neural network is a feed forward neural network with multiple hidden layers which can enhance performance compared to a single layer [12]

Neural networks have existed conceptually since the 1940s [41]. However, numerous barriers stood in the way of practical application, such as efficient algorithms and the necessary computational resources to perform high dimensional matrix multiplication. The earliest instances of neural network models for weather prediction date back to 1991 when an artificial neural network was used to predict minimum air temperature [42] and later in 1995 to predict rainfall and snowfall [43]. These models were able to improve the forecasting accuracy compared to statistical models available at the time [27]. However, the limited forecast of 30-180 minutes and difficulties in obtaining solution convergence made practical application impossible [43]. There are several key developments that have rekindled interest in neural network prediction models including increases in data availability and quality, improved data manipulation techniques and the ever-increasing accessibility and performance of local and cloud-based computational resources. Perhaps the biggest motivator is the development of open-source platforms along with algorithm advancements within the last decade that have made deep learning approaches highly accessible.

Computationally lightweight prediction model alternatives to neural networks include statistical models such as Monte Carlo and ARIMA. Traditional machine learning examples include support vector machine or linear regression which are typically far less computationally demanding than neural networks and have been investigated as forecasting candidates. For example, Ma et al. [44] deployed a traditional machine learning model known as XGBoost, which is comprised of gradient boosted decision trees, to predict air temperature and humidity over a 3-hour period with resulting root mean square error (RMSE) values of 1.77℃ and 6.33 respectively. While traditional machine learning approaches show promise, there are several reasons why a deep learning approach is preferred for weather prediction.

Firstly, many traditional algorithms are unable to model nonlinearity, which is essential in predicting weather evolution [10, 45]. For instance, Shao et al. [37] report that statistical and traditional ML techniques are not well-suited for complex wind forecasting and attribute this need to the turbulent and chaotic behaviour of wind.

Secondly, deep learning leverages the growing volume and accessibility of data. While traditional machine learning models reach a point beyond which additional training data no longer improves model performance, deep learning models have been observed to benefit from the increase in data as exemplified in Figure 2.12. Wang et al. [46] recently compared prediction performance of a traditional machine learning model and a **convolutional neural network** (CNN). With a small training dataset, the accuracies were 0.86 and 0.83 for the two models respectively. However, when a larger, similar training dataset was used, the accuracies were 0.88 and 0.98 respectively.

Lastly, certain deep learning architectures can negate the need for processing steps such as feature extraction. Feature extraction identifies high impact variables or information while eliminating those with a lower impact. For instance, CNNs are particularly well-suited for con-

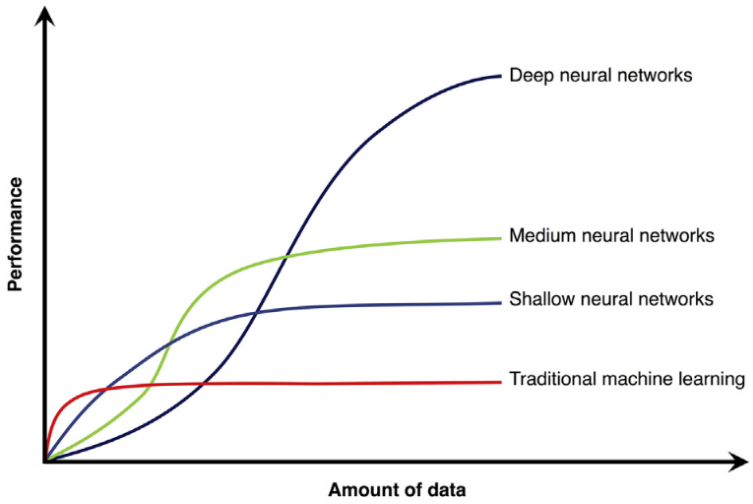

Figure 2.12: Neural networks, and deep neural networks specifically, tend to perform better in the presence of large volumes of data compared to traditional machine learning [13]

densing large, multidimensional datasets while ensuring key information is retained [47]. Signal decomposition is another processing technique used to separate trend, noise and periodicity in a time sequence and is a crucial step in statistical and traditional machine learning algorithms, but can be avoided when using a neural network [48].

Solving real-world problems with machine learning has been accelerated by advances in computing resources, such as graphical processing units. The tensor processing unit (TPU) is hardware specifically engineered to process high-dimensional data much more efficiently than a central processing unit [49]. Furthermore, data is more abundant now than ever with open-source observation station, satellite, RADAR and Internet of Things (IoT) data available [35]. Technological advancements in hardware has spurred the development of open-source deep learning frameworks, such as PyTorch and TensorFlow, which have created tremendous interest in data-driven approaches for weather prediction. Chantry et al. [35] observed the number of papers published daily referencing machine learning is of the order of 100 as of 2021. Three neural networks commonly used to make time series predictions are feed forward, convolutional, and recurrent neural networks.

2.2.2 Feed Forward Neural Network

Feed forward neural networks are the simplest of the neural network architectures and are ubiquitous in problems that have no temporal component, but are less common in time series predictions as they were not specifically created to comprehend sequential data. A FNN may consist of several hidden layers, each of which contain nodes. As data is introduced into the input layer of the model it propagates forward through to the hidden layer where it is multiplied by a bias term, summed, and then and passed through an activation function. The process is repeated once more between the hidden and output layer where the predicted result is compared

with the measurement or true value, often referred to as the ground truth. The calculated error is then used to adjust the weights through a process known as backpropagation. This full cycle is known as an **epoch** and may be repeated many times until the accuracy or loss converges. A FNN containing multiple hidden layers is considered a deep neural network.

Sun et al. [16] have described the challenges of predicting building cooling loads and the multitude of factors that affect heating. They created a FNN and observed that their model improved temperature prediction accuracy thereby reducing running costs by 12.5%. Two models were necessary for this operation, the first to predict meteorological conditions such as cloud cover, wind speed and temperature, and a second forecasting model to predict load evolution as a function of the predicted weather conditions. Karatasou et al. [50] simplified this process by using historical data available for temperature and loading thus negating the need for hourly prediction increments.

2.2.3 Convolutional Neural Networks

Convolutional neural networks are typically used in image processing and recognition and have unique convolutional and pooling layers that extract key information in a sequential order and feed the information into a FNN in the final layer. Temporal convolutional networks are designed specifically with sequential applications in mind and have been observed to perform well on sequential datasets [51]. Compared to a FNN, a CNN is typically more computationally intensive owing to the operations in the convolution layer, but is less demanding than **recurrent neural networks** (RNN) and **transformers** as these models do not perform feature extraction.

Weyn et al. [52] showed it possible to increase weather prediction skill by applying ensemble modelling. They trained 32 CNN models, each with different starting conditions and sets of weights. In this ensemble approach, the mean predictions of all models were shown to improve prediction accuracy. Evidently, running the models numerous times results in longer run times, however, the computations were noted to be inexpensive and could be performed within a reasonable amount of time.

Weyn et al. [52] identified the importance of improving seasonal forecast accuracy at timescales of up to six weeks. They predicted six weather parameters for periods ranging between two and six weeks. They used an ensemble CNN that performed comparably well to the ECMWF subseasonal forecasting model at six weeks, however, for shorter term predictions, the ECMWF model performed better. The ERA5 dataset was used with a grid resolution of 1.4°. Weyn et al. [52] acknowledges that the significantly shorter model runtimes associated with deep learning may improve probabilistic forecasting as the number of ensembles can be increased resulting in more detailed probability distributions.

2.2.4 Recurrent Neural Networks

A **recurrent neural network** (RNN) can be thought of as many feed forward networks in parallel, each processing a single timestep in a sequence and sharing the prediction with the subsequent network giving the model sequence comprehension. **Long short-term memory** (LSTM) networks are a form of RNNs designed to address the vanishing or exploding gradient problem. When the gradient vanishes, the model is no longer capable of learning as the weights become static and when the gradient explodes the model becomes unstable [53].

Chen et al. [20] states the computational burden and expertise needed to operate traditional dynamic models to predict precipitation is too great for specialised applications in agriculture where accurate and localised rainfall is to be predicted. Therefore, they have turned to a data-driven approach to predict monthly rainfall using an LSTM model. They found that a window lag of four months produced the best results at two weather stations in Turkey, with RMSE values of 2.2cm and 2.6cm. Prediction of solar irradiance is a valuable tool for photovoltaic energy generation. The ability to predict the intensity of solar irradiance over several hours to several days allows energy providers, who increasingly rely on energy generation from private consumers, to optimise power generation [54]. Four models were built and used to assess prediction performance of solar irradiance, namely persistence, linear regression, a backpropagation neural network and an LSTM. The LSTM model performed significantly better than the persistence model and the linear regression model when predicting over 18 hours [54]. Shaojie Bai [51] recently demonstrated that LSTMs can be used to accurately predict storm surges in periods of up to nine hours, which would act as an early warning system. These surges manifest themselves in the form of rapidly rising sea levels during a hurricane or storm.

2.2.5 Hybrid Machine Learning Workflows

Neural networks have been identified as being particularly promising in precipitation forecasting. A MetNet model developed at Google was shown to predict precipitation accurately over the course of 8 hours. In this hybrid approach, several models were used at different stages including LSTMs and CNNs. The F1 score demonstrated that the neural network-based approach generated forecasting products which performed better than the High-Resolution Rapid Refresh over eight hours [49]. This model is developed by the the National Oceanic and Atmospheric Administration. However, Casper et al. [49] conclude that the model will require larger volumes of data to increase the forecast length beyond eight hours.

Fu et al. [19], upon evaluating many neural network architectures, settled on a combined **Bidirectional-LSTM** (Bi-LSTM) and a one-dimensional CNN to predict 2m air temperature, 2m relative humidity and 10m wind speed over seven days. They used local weather station data from 10 weather stations in Beijing and a total of nine weather parameters. Since

the authors compare their model's prediction accuracy to numerical weather prediction results, the quantitative performance relative to the local weather observations is uncertain. It was noted that the model contained 1.225 million nodes and convergence occurred after 500 epochs suggesting the model runtime was considerable. While the results are fascinating, the generalisability of the model to new data, and therefore, the practical value, is uncertain.

2.2.6 Hybrid Numerical Weather Prediction and Machine Learning Workflows

Hybrid approaches combine numerical weather prediction and machine learning with specific objectives, such as improving accuracy or reducing model runtime. Schultz et al. [14] identify the potential in the large volumes of data available through modern data acquisition systems and the potential of a data-driven approach to not only complement existing weather prediction techniques in a hybrid approach, but also to generate end-to-end products. This has the benefit of reducing or eliminating the time-consuming data assimilation and post-processing steps as seen in Figure 2.13. However, it is acknowledged that many existing data-driven models are limited to short term forecasting up to 24 hours.

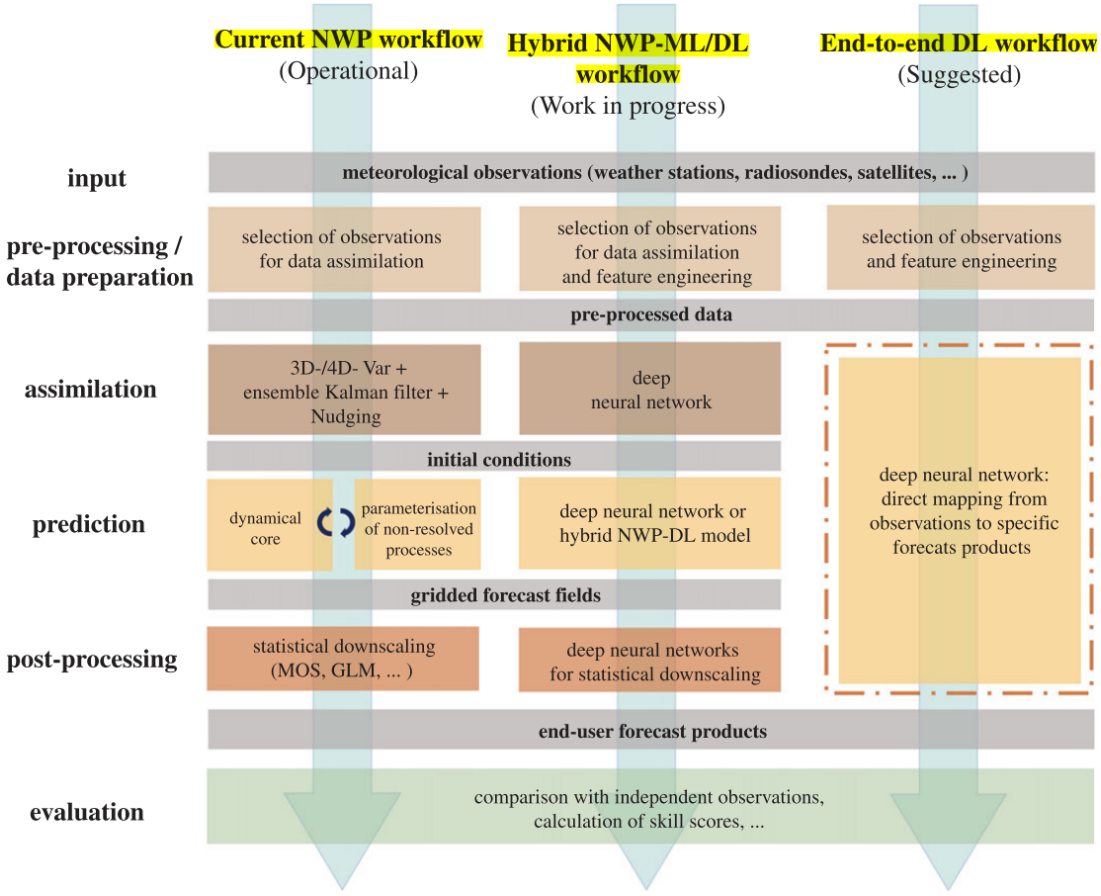

Figure 2.13: A description of NWP, hybrid and data-driven workflows indicate the reduction in number of processing steps when a data-driven approach is adopted [14]

Recently, Frnda et al. [45] demonstrated a deep learning hybrid approach could improve weather prediction accuracy. They used a deep neural network along with ECMWF weather predictions and additional parameters such as water area, the air quality index and mean effective green infrastructure to improve the forecasting accuracy locally. The forecast error was reduced by 13% for 2m air temperature and 45% for daily precipitation over a three-day period. Furthermore, traditional machine learning methods such as Gaussian process regression, linear regression and support vector machine were trialled with Gaussian process regression producing the highest accuracy for 2m air temperature and linear regression producing the best result for daily precipitation. However, the deep neural network had the best performance of all machine learning models.

Chattopadhyay et al. [31] highlight the growing number of deep learning models being developed for weather prediction and proposes techniques to support commonly used deep learning models and improve forecast skill. Amongst the techniques, significant attention is brought to data assimilation, and specifically the use of the sigma-point Kalman Filter algorithm to reduce the impact of noisy observations; this is a random approach while conventional Kalman Filtering is typically deterministic. Data reanalysis is a technique whereby short term predictions are combined with observations to create a complete picture of historical data and are referred to as "maps without gaps". Chattopadhyay et al. [31] believe when deep learning models are trained on this type of assimilated data, they will not encounter the same limitations as traditional forecasting models.

2.2.7 Challenges in Deep Learning

As with numerical weather predictions, it is difficult to overstate the importance of data selection and processing in machine learning. A model's ability to make accurate and meaningful predictions depends on the quantity and quality of data available. Data engineering is the process of extracting key parameters and combining them to form new parameters the model can process more easily and is a skill that must be refined. Data processing is almost always the most critical and time-consuming procedure in any machine learning problem and is very often the limiting factor when applying machine learning to solve problems.

Rasp et al. [21] have identified a key challenge in establishing benchmark datasets to enable the comparison of model performance. Only once a baseline has been established and a consistent set of metrics introduced can model performances be assessed in a meaningful way. They also note that high quality reanalysis data is only available from the last forty years, which corresponds to 350,000 training samples for datasets with high sampling rates of one hour.

Interpretation of neural networks has long been a point of contention. Methods are being developed to better understand why models make the decisions they do [55]. According to Schultz et al. [14], one of the biggest barriers to adoption of deep learning techniques within

the scientific and meteorological community is the lack of explainability and constraints that obey fundamental physical laws. Raissi et al. [56] discuss physics-informed neural networks and the addition of physical constraints in the form of partial differential equations which may help to accelerate adoption. As with numerical weather prediction, understanding the uncertainty in a machine learning approach is crucial to its adoption.

One important challenge specific to deep learning weather prediction is the forecast length. One approach is to train the model to predict a single timestep, normally one hour, and loop to generate the full forecast window with the number of loops determining the forecast length. The alternative approach is to train the model on a predefined number of timesteps corresponding to the full prediction window. Ultimately, the flexibility to adjust the forecast length is lost in this approach. The disadvantage of using a single timestep approach is the absence of a robust correlation between single timestep performance and performance over multiple timesteps.

2.3 Summary

Numerical weather prediction is the gold standard for predicting the weather and the forecasting skill has increased steadily over the last few decades. Partial differential equations that describe the atmospheric state must be solved. Before this is possible, data must be acquired and assimilated. Data assimilation is the process of creating a complete description of the atmospheric state from a finite number of measurements and defines the initial and boundary conditions. Many simulations, in the form of ensemble modelling, are run on supercomputers that are expensive and runs can take many hours. These simulations are repeated for every weather prediction update. There is significant investment in research, infrastructure, and maintenance of existing numerical weather prediction systems. Turbulence, non-linearity of the system and the chaotic tendencies of weather are some challenges which must be overcome.

Deep learning is a field within machine learning and is a a technique used to solve many different real world problems. Specifically, the objective is to predict the future state of the weather using historical data. Deep learning has been shown to be a powerful prediction tool in natural language processing, image recognition and anomaly detection. The literature suggests that deep learning is already capable of high-accuracy short term weather forecasting. In the long term, the hope is that deep learning will be able to substitute numerical weather predictions and reduce the financial burden associated with this traditional approach. The benefits of a data-driven approach are evident. Deep learning models are trained once and can rapidly update predictions as new weather data becomes available. It is possible to train deep learning models on widely available and low-cost hardware such as central and graphical processing units. The training time may range from minutes to hours and once the model is trained, predictions can often be made within seconds. While it is unlikely that a deep learning approach would eliminate the process of data assimilation, it has the potential to simplify it.

However, like numerical weather prediction, deep learning faces challenges. Neural networks are difficult to interpret, the process for optimising them is highly iterative and time consuming. A benchmark for weather prediction testing must be established so performance can be quantified in a structured and repeatable way. Processing and preparing high quality data that a model can learn from is time consuming and is a step for which the criticality can not be overstated.

Chapter 3

Methodology

The neural network architectures most widely cited in the machine learning weather prediction studies reviewed, along with their distinguishing capabilities, are listed in Table 3.1. LSTMs are applied frequently in sequential problems as they address the issue of loss of long-term memory [57]. The Bi-LSTM recurrent neural network builds upon the LSTM structure. In a Bi-LSTM model a duplicate layer is produced. Sequential information flows in chronological order through the first layer while the duplicate layer is used for the same sequential information, but this time the order is reversed. This provides the model with far more context as key information at both the start and end of the sequence is available.

It was determined through an iterative approach that the Bi-LSTM performed better than a conventional LSTM. In Chapter 4.1, a standalone Bi-LSTM is used to make 24-hour predictions. For 72-hour predictions in Chapter 4.2, a feed forward network was coupled to a Bi-LSTM model as this was found to improve accuracy.

Capabilities	**FNN**	**CNN**	**RNN**	**LSTM**	**Bi-LSTM**
Deep learning	Yes	Yes	Yes	Yes	Yes
Non-linearity	Yes	Yes	Yes	Yes	Yes
Sequence comprehension	No	Yes	Yes	Yes	Yes
Short & long-term dependencies comprehension	No	No	No	Yes	Yes
Past and future sequence comprehension	No	No	No	No	Yes

Table 3.1: Evaluation matrix for neural network architectures ubiquitous in time series prediction with a description of their capabilities

3.1 Feed Forward Neural Network

To develop an understanding of recurrent neural networks, it is necessary to first understand how basic neural networks function. A feed forward neural network is one of simplest neu-

ral network architectures and lays the foundation for deep neural networks and Bi-LSTMs. Structurally, it contains a input layer, followed by a single hidden layer and an output layer.

For data with a temporal component, the number of input nodes in the input layer corresponds to the number of weather parameters multiplied by the input sequence length or **context**. The context is the window of time steps immediately leading up to a prediction and used as a direct input to make that prediction. The context must be carefully selected for a specific forecast length. The number of hidden layer nodes required will depend on the complexity of the data and number of input and output layers. Finally, the number of nodes in the output layer is determined by the number of weather parameters and batch size. Batches are segments of the full dataset created to enable the data to fit into available memory or improve speed.

In a feed forward network, each weather observation, x_i, commonly referred to as an **index**, is multiplied by the corresponding weight, w_{ji}^1, and a common bias term, w_0^1, in the hidden layer. This process is repeated between the output and hidden layer with weights w_{kj}^2, and the second bias term, w_0^2, as seen in Figure 3.1.

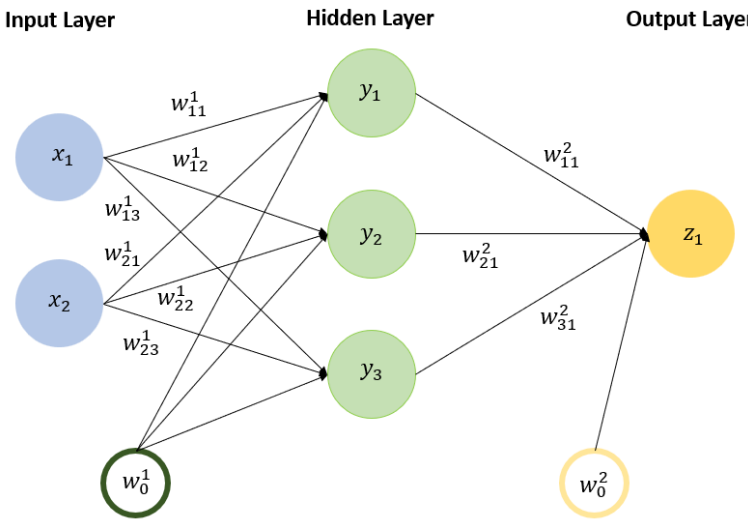

Figure 3.1: The architecture of a single variable feed forward neural network with a single continuous output

Mathematically, the local value at each node in the hidden layer is defined as

$$y_j = f\left[\sum_{i=0}^{n} w_{ij}^1 x_i + w_0^1\right] \tag{3.1}$$

where x_i is the i^{th} index or observation and w_{ij} is the weight, w_0 is the bias term and f an activation function [58].

Activation functions are used to constrain the output or to ensure it does not vanish or explode, as previously discussed. A rectified linear unit (ReLU) is an activation function and is highly efficient to compute. It is a piecewise function and is defined as

$$f(x) = \begin{cases} y = 0 & \text{if } x \leq 0 \\ y = x & \text{if } x > 0 \end{cases} \tag{3.2}$$

The bias term is the threshold beyond which the sum of the weights and inputs becomes significant and has considerable influence on the model. The bias term is capable of scaling all values in a layer concurrently.

Applying the same procedure to the output layer, the predicted output z_k is calculated as

$$z_k = f\left[\sum_{j=0}^{n} w_{jk}^2 \left[\sum_{i=0}^{n} w_{ij}^1 x_i + w_0^1\right] + w_0^2\right] \tag{3.3}$$

After computing z_k, it is compared to the true value t_k from the training dataset and the error is computed to quantify how well the model performed. Mean square error is used to compute the cost function J and is defined as

$$J = \frac{1}{2}\sum_{k=0}^{M}(t_k - z_k)^2 \tag{3.4}$$

Backpropagation is the process of iteratively adjusting the weights, in this instance moving from the output to the input layer, with the objective of minimising the error. This is achieved by computing the derivative of the cost function with respect to the weights in the hidden and output layers. The derivatives to compute are

$$\frac{\partial J}{\partial w_{jk}^2} = \sum_{k=0}^{K}(z_k - t_k)\frac{\partial z_k}{\partial w_{jk}^2} \tag{3.5}$$

$$\frac{\partial J}{\partial w_{ij}^1} = \sum_{k=0}^{K}(z_k - t_k)\frac{\partial z_k}{\partial w_{ij}^1} \tag{3.6}$$

Each cycle of data being fed forward and backpropagation is known as an epoch. We wish to calculate the updates to the weights, Δw_{ij}^1 and Δw_{jk}^2 , such that equation 3.4 is minimised [58].

$$\Delta w_{kj}^2 = \eta(z_k - t_k)f'\left[\sum_{j'=0}^{nh} w_{kj}^2 y_{j'}\right] y_j \tag{3.7}$$

$$\Delta w_{ji}^1 = \eta f'\left[\sum_{m=0}^{M} w_{ji}^1 x_i\right] x_i \sum_{m=0}^{M}(z_k - t_k)f'\left[\sum_{j'=0}^{nh} w_{kj}^2 y_{j'}\right] w_{mj}^2 \tag{3.8}$$

3.2 Bidirectional-LSTM

There is much overlap between a recurrent neural network and feed forward network and in the case of the RNN, the output recurs as an input. Simply, this means the recurrent network uses the current input and past state of the sequence to predict the future state, making them more suitable for use in time series predictions.

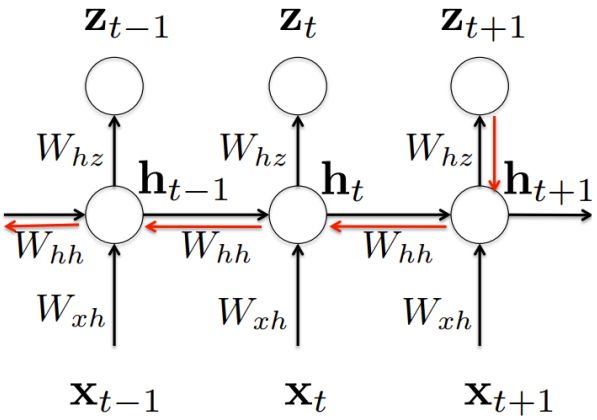

Figure 3.2: A recurrent neural network is a series of feed forward networks operating in parallel and use observations and the past state from the previous network as inputs. The red arrows denote the backpropagation route [15]

On a model level, the hidden state, $\mathbf{h}_t$, and the prediction, z_t, are defined as

$$\mathbf{h}_t = tanh(W_{hh}\mathbf{h}_{t-1} + W_{xh}\mathbf{x}_t + \mathbf{b}_h) \tag{3.9}$$

$$z_t = W_{hz}\mathbf{h}_t + \mathbf{b}_z \tag{3.10}$$

where W_{hh}, W_{xh} and W_{hz} are weights to be computed during the training process and $\mathbf{b}_h$ and $\mathbf{b}_z$ are bias terms [15].

These three weight matrices are shared across all time steps, regardless of the context length. The long short-term memory model is a recurrent neural network variant and overcomes issues with the vanishing and exploding gradient seen in a "vanilla" recurrent neural network. For a large number of input timesteps, the gradient may vanish or tend to zero when the weight is less than unity. It may explode or tend to infinity if the weight is greater than unity. An LSTM model contains four gates, namely a forget gate, an input gate, a state gate and an output gate.

The Sigmoid and hyperbolic tangent functions ensure the prediction value is within -1 and +1. Without these functions, it is possible that the output will become unstable. The Sigmoid activation function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.11}$$

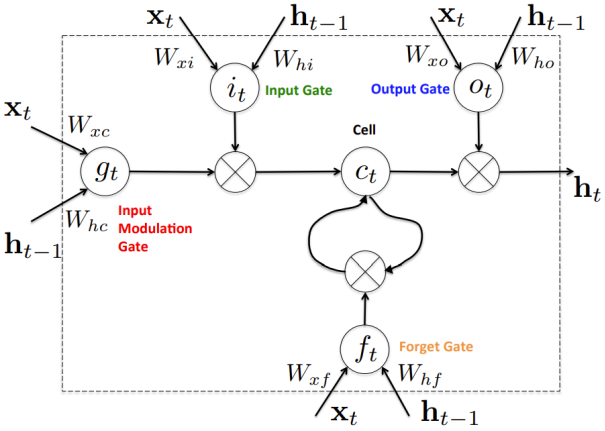

Figure 3.3: An LSTM cell has an input, forget, input modulation and output gate which take input from the hidden state $\mathbf{h}_{t-1}$ and the current observation $\mathbf{x}_t$. While omitted for consistency with Figure 3.2, $\mathbf{c}_{t-1}$ is the output from the previous cell and used to calculate $\mathbf{c}_t$ [15]

The forget, the input, input modulation and the output states are respectively defined as

$$\mathbf{f}_t = \sigma(W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + b_f) \tag{3.12}$$

$$\mathbf{i}_t = \sigma(W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + b_i) \tag{3.13}$$

$$\mathbf{g}_t = tanh(W_{xc}\mathbf{x}_t + W_{hc}\mathbf{h}_{t-1} + b_c) \tag{3.14}$$

$$\mathbf{o}_t = \sigma(W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + b_o) \tag{3.15}$$

where W_{xf}, W_{xi}, W_{xc}, W_{xo}, W_{xh}, W_{hf}, W_{hi}, W_{hc} and W_{ho} are weight vectors [15].

On a cell level, the hidden state, $\mathbf{h}_t$, and the memory cell state, $\mathbf{c}_t$, are defined as

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t \tag{3.16}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ tanh(\mathbf{c}_t) \tag{3.17}$$

where $\circ$ is the composition of two functions [15].

A Bi-LSTM takes this one step further with the flow of information in the forward direction as discussed above, and adds a second layer. The second layer in this network is identical in construction to the first, however, the input sequence is fed in a reverse order. The first layer gives context of the sequence in chronological order while the second layer works from the back to the front of the sequence giving the model context of the future state. The model can make more informed decisions with this additional information.

To calculate the gradients and adjust weights, recurrent neural networks use the Backpropagation Through Time methodology [59], with the process being similar to that of a feed forward neural network. The difference is the weights are now vectors and calculation of the derivative takes place with respect to each timestep [59].

3.3 Exploratory Data Analysis

The data is made available by the Met Office with data from two London weather observation stations Kew Gardens (51.482, -0.294) and Heathrow (51.479, -0.451). The data was manually extracted from the Centre for Environmental Data Analysis website and contains weather information from 2015-2021 with dozens of hourly weather parameters, hereinafter referred to as **features** for consistency. However, not all features are available for all weather stations and so the selection was limited to six unique features (see Figure A.1 in Appendix A). The three features of particular interest are air temperature, relative humidity and wind speed at both Heathrow and Kew Gardens, and are plotted in Figure 3.4.

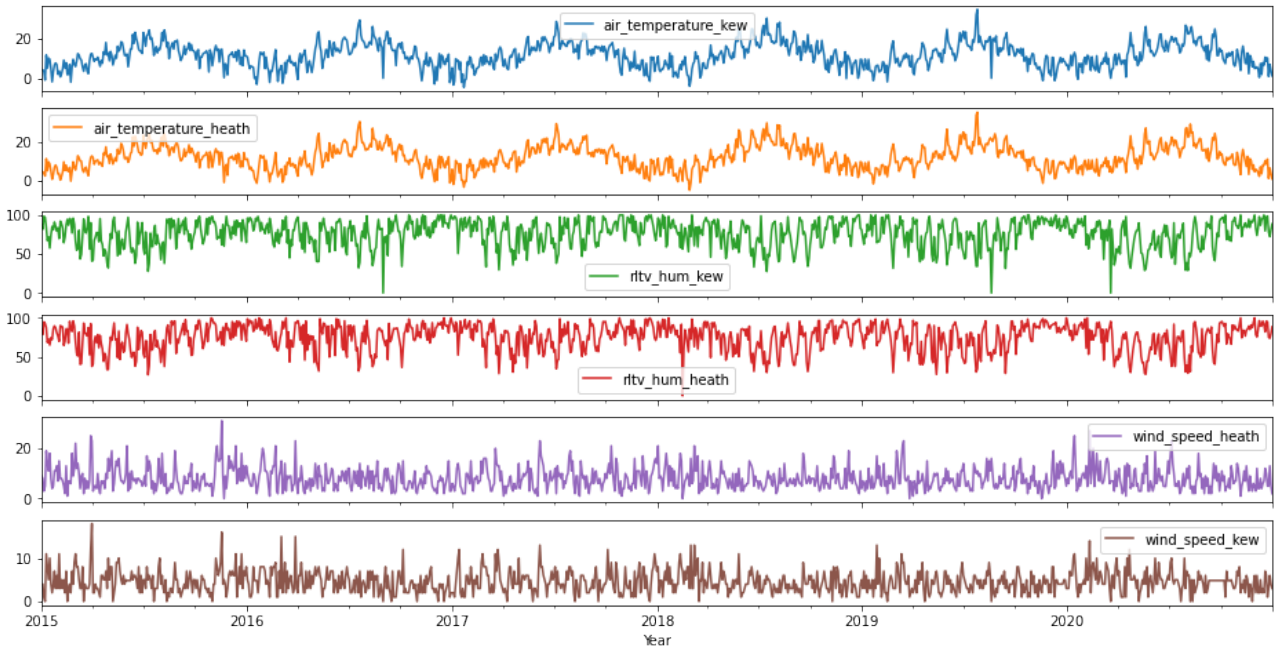

Figure 3.4: Six years of air temperature, wind speed and relative humidity data from Kew Gardens and Heathrow demonstrate periodicity and correlation between the features with every 50^{th} datapoint used for illustrative purposes

Next, probability distributions for air temperature, relative humidity, wind speed and wind direction were generated and are illustrated as diagonal plots in Figure 3.5. The non-diagonal plots are two-dimensional probability distributions using each feature combination. From these figures, it is evident that neither air temperature, relative humidity nor wind direction follow a normal distribution. Instead, wind direction has strong bimodal tendencies while air temperature and relative humidity exhibit similar, less pronounced characteristics. This bimodality could be explained by prevailing winds present during the day and night. The diagonal probability distributions generally indicate good agreement between locations. However, wind speed is clearly an exception with the shape of the distributions varying significantly in the two locations. Kew Gardens is near the River Thames in a more built up area while Heathrow Airport is more isolated with fewer obstacles in its vicinity.

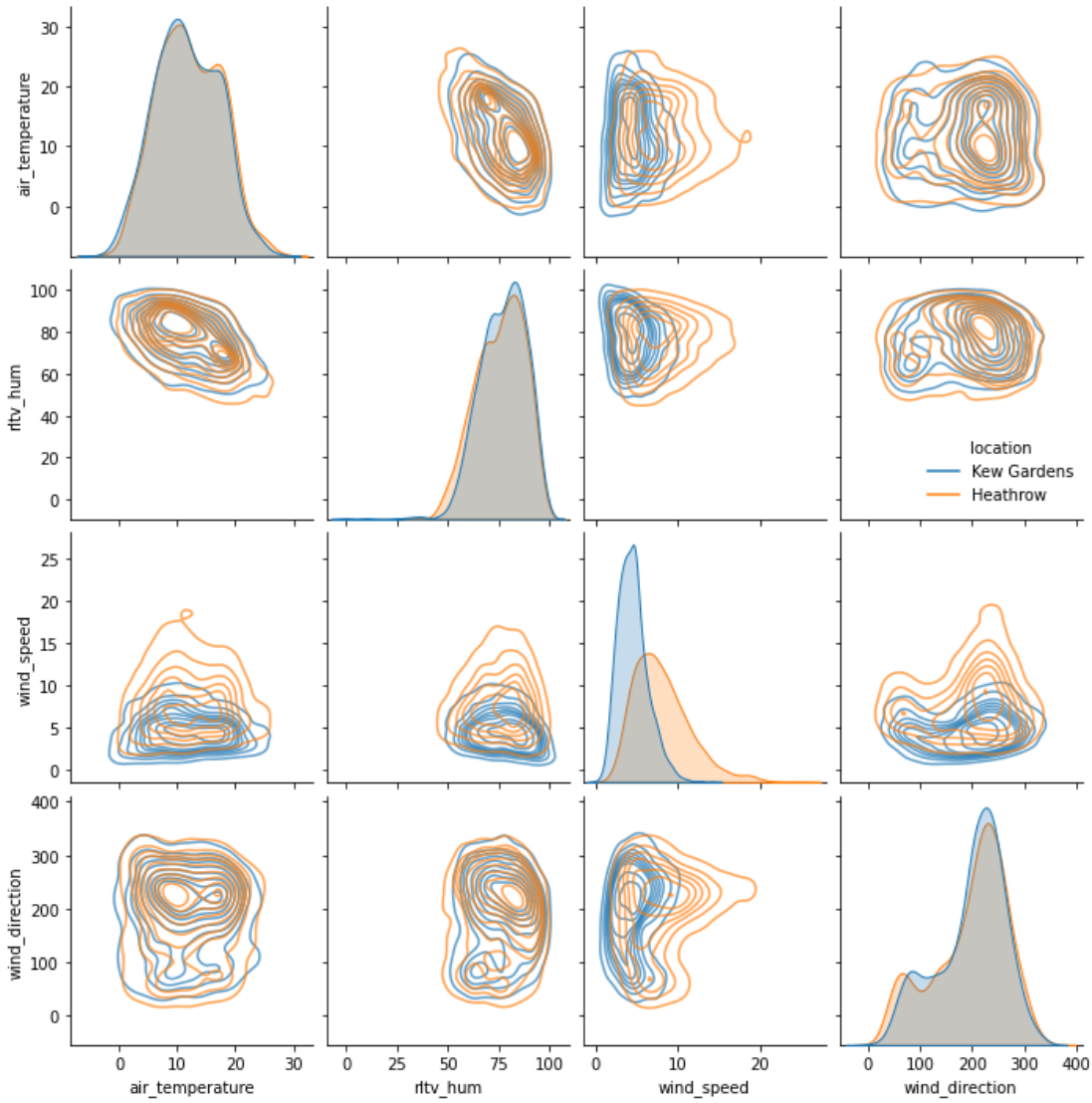

Figure 3.5: Comparing the wind speed diagonal plot, there are significant differences between Heathrow and Kew Gardens. Furthermore, all of the two-dimensional wind speed combination plots have notable location-based differences

Several hundred missing values were detected within the dataset. As the model is unable to process missing values, it is conventional to impute missing data. The objective of data imputation is to substitute missing data with information inferred from the same or other feature columns, such that the impact of imputed data on the model is minimised. This is most commonly achieved by calculating the average of each feature column and imputing missing rows with the average, thereby ensuring key statistical properties such as the mean remain unaffected. An alternative approach is to predict missing values using the other features within that row. This approached was explored but was abandoned due to time constraints within the project.

The data exploration phase is used to understand the data and identify key trends and features, such as periodicity. This understanding will determine which information is important and should be considered in the data processing phase and which information can be neglected.

3.4 Data Processing

Weather data collected by sensors is often comprised of trend, noise, seasonality, non-linearity and chaotic components [20]. While data for the resolution and accuracy of sensors used by the Met Office is not readily available, commercially available weather transmitters such as the ATMOS41 has a resolution of 0.1°C and accuracy of ±0.6°C while the WXT520 has a resolution of 0.1°C and accuracy of ±0.1°C [60]. Noise is always present in the signal and is the result of the A/D board, electromagnetic interference, and radio frequency interference. Techniques for noise reduction include filtering, signal conditioners and signal smoothing using averaging. One commonly used technique in noise reduction is Kalman Filtering, however, the noise variance parameter must be manually selected, which often proves challenging [32]. Time-series datasets can be decomposed into a seasonal, trend and noise component using moving averages or filters such Baxter-King or Hodrick-Prescott [61]. While it is acknowledged that signal filtering is an important step in statistical and traditional machine learning time-series forecasting as well as numerical weather predictions [4, 62], exploring the effects of filtering in the case of a neural network are not within the scope of this project.

Data must be processed with the algorithm and features in mind. The objective is to process the data such that a recurrent neural network can interpret it and do so in the most efficient way possible. Feature engineering is the process by which features are combined, modified, or removed to provide the model with the most relevant information. Neural networks are better able to interpret wind speed and direction when they are decomposed into their longitudinal and lateral components. This decomposition step improves the interpretability, not just for the algorithm, but improves the visual interpretability for the human brain.

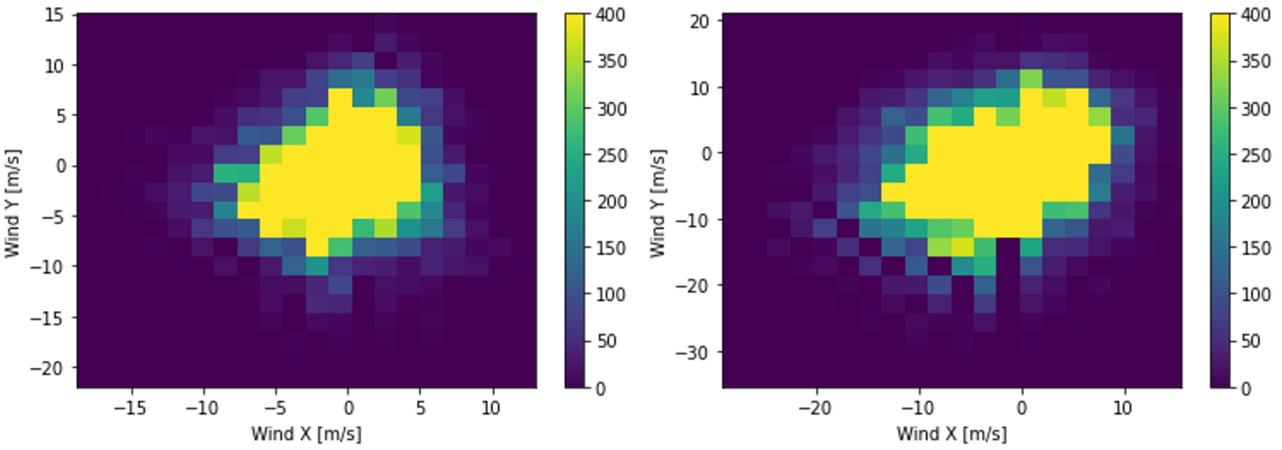

Figure 3.6: Two-dimensional histograms of longitudinal and lateral wind speed at Kew Gardens (left) and Heathrow (right) suggest the wind behaviour is slightly more erratic at Heathrow

We can see from the two-dimensional histograms in Figure 3.6 that the longitudinal and lateral wind speed vectors form a circular distribution at Kew Gardens with no significant skewing. In contrast, the distribution at Heathrow is more elongated and contains more discontinuities.

These discontinuities may be the result of human activities as they appear to be more erratic. The elongation indicates a stronger correlation between high longitudinal and lateral wind speeds and points to preferential air currents between the southwest and northeast. The features present in the original dataset are air temperature, dew point temperature, wet bulb temperature, relative humidity, and wind speed and direction. The wet bulb temperature combines humidity and temperature to describe how efficiently the body can be cooled.

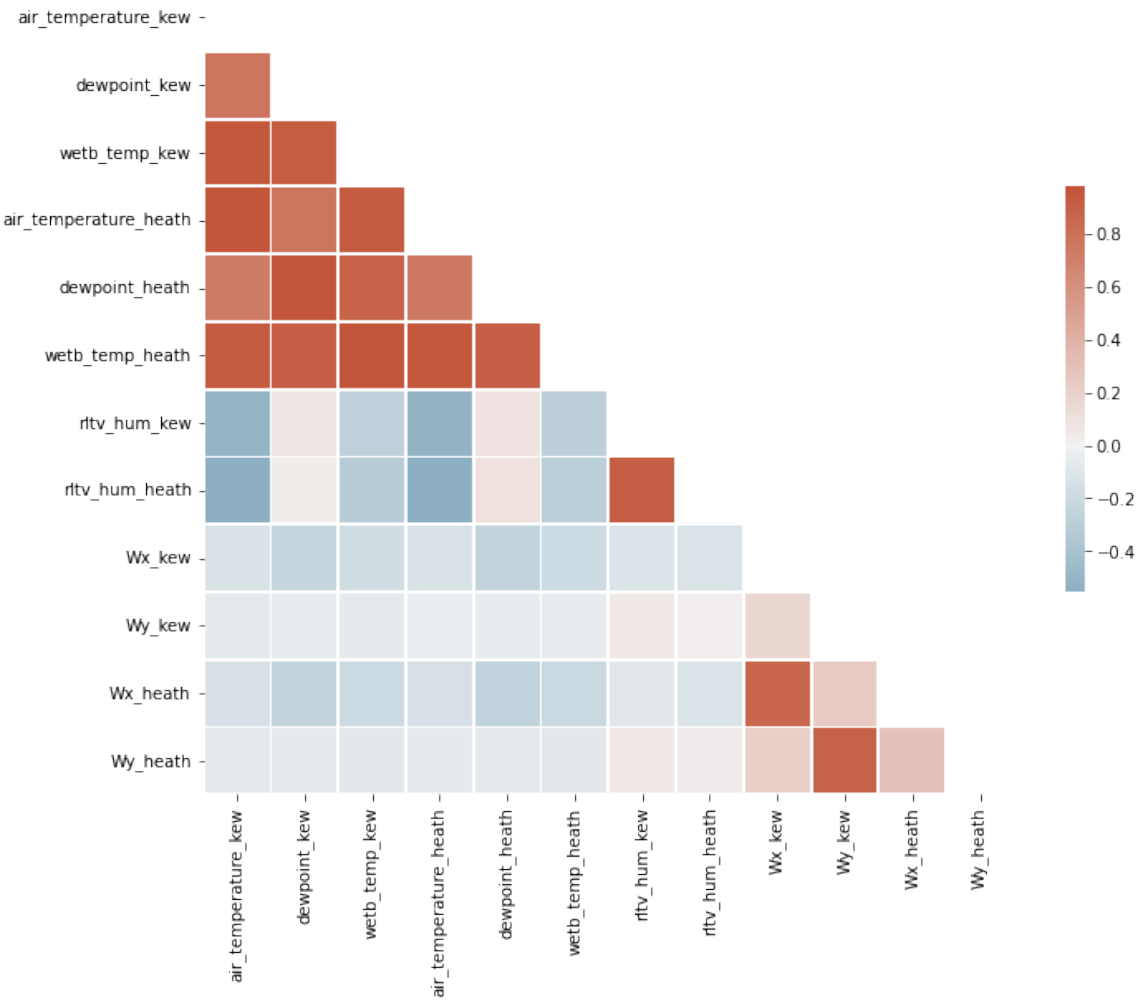

Figure 3.7: A Pearson correlation plot with the processed features indicates high correlation between wet bulb, air and dew point temperatures with minimal correlation between the remaining features (see Figure A.1 for unprocessed features)

A Pearson correlation plot is used to identify trends between the weather features. The Pearson correlation coefficient in equation (3.18) is a metric used to identify **linear** correlation between two or more variables [63] and is defined as

$$r_{xy} = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_1^n (x_i - \bar{x})^2 (y_i - \bar{y})^2} \tag{3.18}$$

where x_i and y_i are the i^{th} observations in each feature set, and $\bar{x}$ and $\bar{y}$ are their means.

However, the relationship between air temperature, humidity and wind speed seen in equations (2.1) - (2.5) are not always linear. This metric is commonly used in exploratory data analysis as it is fast to compute and can highlight simple relationships between variables and improves data visualisation.

With the features selected and processed, the next step is to split the dataset. The dataset is split into a training set, a validation set and, finally, a test set. The training dataset is directly used in the calculation and adjustment of the weight vectors during the training process. It is the primary source of information used in training, while the validation dataset is used to check performance of the model on data during training. The validation data do indirectly influence the training process. The test dataset assesses the model's performance on unseen data and does not influence training. Next, the dataset is normalised. This is performed by calculating the mean and standard deviation for each feature column and subtracting the mean from each observation and dividing by the standard deviation. Normalisation is essential when training neural networks as amplitude affects the model with larger values exerting greater influence on the model. It is essential to compute the mean and standard deviation from the training dataset as including data from the validation and test datasets to calculate the mean and standard deviation would reveal information to the model during training that should be withheld and may result in overfitting [63].

A key parameter in timeseries forecasting is the context or length of the input window. The required length of the context will depend heavily on the length of the forecast among other factors. There are two distinct ways in which the input sequence is provided to the LSTM.

In the **multi-timestep** approach, the forecast length must be defined before the model is trained and it is not possible to alter the forecast length after training. The entire forecast is made from the input or context, which is comprised entirely of measured data and does not contain any predicted inputs.

In the **single-timestep** approach, there is more flexibility and forecasts of any length can be defined after model training is complete. In the single-timestep approach, the context is initially comprised of measurement data only. If the the forecast length is equal to the context length, then the final context will be comprised entirely of predicted values by the final prediction. The context window moves forward one timestep after each prediction is made and that predicted value is used for the next prediction and the value at the front of the sequence is dropped to maintain a consistent context length. The single-timestep approach was chosen as it provides the most flexibility during the trial and error phase and is vital in establishing the capabilities and limitations of the model.

Finally, LSTM neural networks require three-dimensional data as an input. The first and second dimensions are the observation and feature vectors respectively. The third dimension is the context and it is this additional dimension in the form of a time sequence that gives recurrent neural networks memory and the ability to interpret temporal sequence and context.

3.5 Parameter Evaluation and Optimisation

The process of optimising a neural network is highly iterative and the feature and hyperparameter selection is based on trial and error. This subsection reviews the impact of hyperparameter and feature selection on the prediction accuracy and runtime. The model performance is quantified by means of commonly used error metrics, namely mean absolute error (E_1), root mean squared error (E_2) and maximum error (E_∞).

$$E_1 = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{t_i - z_i}{t_i}\right| \tag{3.19}$$

$$E_2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - z_i)^2} \tag{3.20}$$

$$E_\infty = max\,|t_i - z_i| \tag{3.21}$$

where t_i is the measured value and z_i is the predicted value.

With the error metrics established, it is now possible to quantify and compare the performance of each model setup and the associated training time. The epoch number is the number of times the model sees the entire dataset and has a significant impact on the training time. Conventionally, the training and validation losses indicate when training should be stopped, i.e., the number of epochs. Extensive training results in overfitting while too few epochs results in underfitting. When overfitting occurs, the model will often perform poorly on new data, but will give the impression of strong performance on the test dataset. This stage is so critical that TensorFlow has developed dedicated software, TensorBoard, to track and visual the training process. Two models were developed, the first capable of making 24-hour forecasts and the second, capable of making 72-hour forecasts. The methodology and and selection of parameters for the one day and three day models are discussed in Sections 3.5.1 and 3.5.2.

3.5.1 Model A: One Day Forecast

When designing the first model, the objective was to build a benchmark model capable of predicting air temperature at Kew Gardens over a 24-hour period. The entire dataset spans six years of weather data with a total of 52,608 samples and six features. The first task was feature selection. From the Pearson Correlation in Figure 3.7, three features stood out due to their high cross-correlation, namely air, wet bulb and dew point temperature as seen in Figure 3.8.

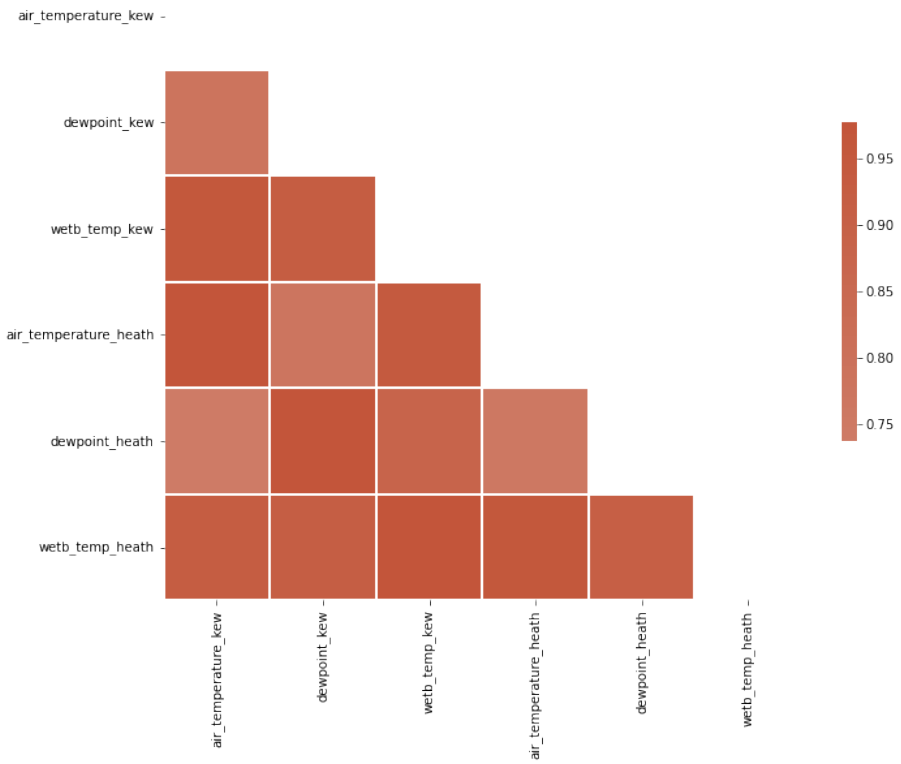

Figure 3.8: Air, wet bulb and dew point temperature measurements from Heathrow and Kew Gardens are used to predict air temperature at Kew Gardens with the Pearson plot demonstrating excellent correlation between all six features

Model reproducibility is essential to make meaningful comparisons between different models. As neural networks are stochastic, one method of reproducing results is to use a random seed generator. A seed is a number that ensures the model starts with the same initial conditions and results are repeatable. A seed of two was arbitrarily chosen and used consistently throughout the modelling.

The training, validation and test datasets are split up in fractions of 0.7, 0.15 and 0.15 respectively with the chronological sequence of the data maintained. This corresponds to a sample size of 36,825, 7,891 and 7,892 observations respectively. A context length of 120 is subtracted from the dataset resulting in a final test size of 7,772 samples. Feature selection and hyperparameter optimisation are highly iterative processes.

Conventionally, a framework such a *GridSearchCV* would be used to automate parameter optimisation. The parameters of interest are specified and the model is trained with each combination and the respective performance recorded. Because of the difficulties in implementing automation, optimisation was performed manually until the RMSE value fell below 2°C in all instances. The model was initially trained with three features and the performance noted. More features were added throughout the iteration process with the view of assessing if they increased performance. Figure 3.9 illustrates the input and output of the model and how the forecast is compared to the measured data.

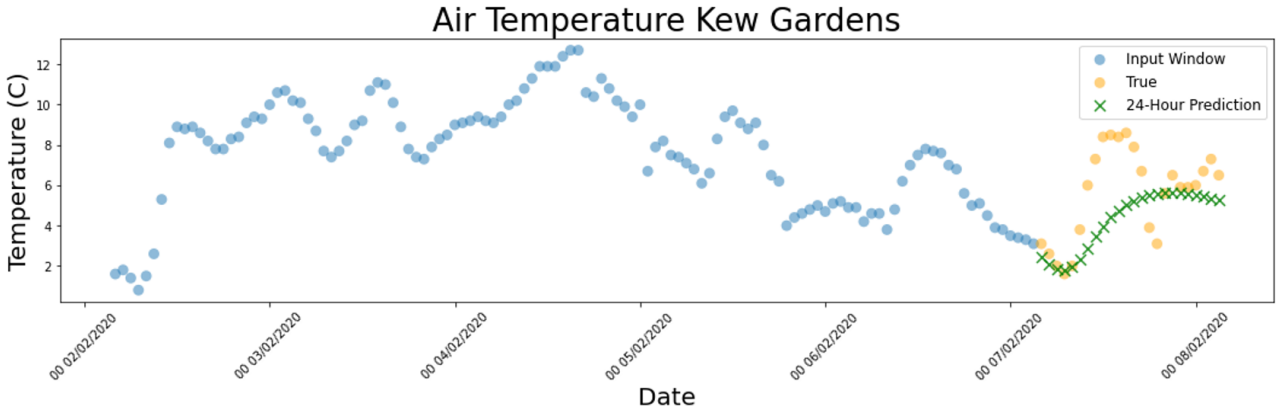

Figure 3.9: A context length of 120 hours is used to make the first single hour prediction and the process is repeated 23 times with the context being updated with each new prediction to generate a 24-hour forecast and, finally, the error is computed

After the model was trained with the training and validation data, the portion of test data set aside was used to check generalisability. The entire test dataset corresponds to roughly one year of data in 2020 and is plotted in Figure 3.10. The model uses 120 measured timesteps to make single timestep predictions. This process is repeated across the entire test dataset and 7,772 single-hour predictions are generated. The root mean, mean absolute and maximum errors were 0.89°C, 0.62°C and 12.81°C respectively. The air temperature is seen to closely follow the true air temperature at Kew Gardens with several exceptions around timestep 700. A linear regression model is generated from the actual temperature with the r-squared value of 0.979 quantifying performance of the predicted values with respect to the linear model in Figure 3.10.

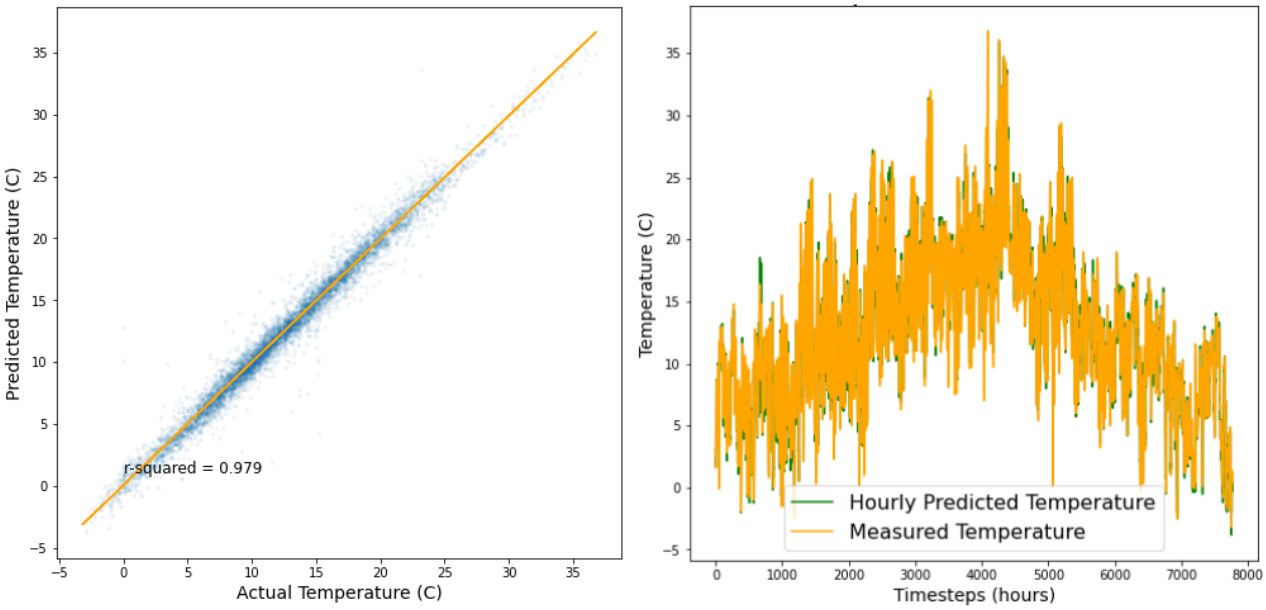

Figure 3.10: With a forecast length of one hour and a context length of 120 hours, the model is able to generalise well to an entire year of test data and provides the confidence needed to extend the model to make 24-hour forecasts over different seasons

In addition to the dataset size and model complexity, the training time is dictated by the batch size and number of epochs. Batching is the process of segmenting the input data with the objectives of, firstly, ensuring the data fit within the available computer memory and, secondly, reducing the training time as fewer parallel mathematical operations are performed on smaller batches. The trade-off of smaller batches is that fewer data are used to adjust the weights. In contradiction to expectations, Radiuk [64] states more batches result in better performance. For the neural network, the optimal number of epochs was found to be two, meaning the dataset was introduced to the model on two separate occasions. Running two epochs independently produced different results to running two epochs sequentially, which can be explained by the stochasticity of the algorithm and the seed number. Running more than two epochs further increased the R-squared value while reducing the RMSE. However, this improvement was not reflected in the 24-hour forecast, whereby a considerable drop in performance of both metrics was observed. It was recognised, that the only way to circumnavigate this issue, would be to train the model with respect to the entire forecast sequence. However, developing new models with a fixed length forecast and optimising them were deemed outside of scope of the project.

Conventionally, a training and validation loss graph, as seen in Figure A.2 in Appendix A, is an excellent metric to determine the number of epochs that produce the highest performance and generalisability. Typically, the training loss will continue to decrease with the number of epochs while the validation loss decreases initially, but tends to increase beyond a certain point indicating that the model is overfitting to the training data. Our initial methodology relied heavily on this approach to assess performance in the early stages of the project. In a traditional setting, a loss graph provides excellent insight into the training characteristics of the model, however, the information gained in this instance is limited to the general trajectory of the loss curves. Ultimately, the only metric that could be relied upon was the performance of the 24-hour forecast. The strategy shifted to running each set of model conditions with 1 - 6 epochs and recording the performance to identify the right number of epochs.

As is expected with neural networks, the model performed worse when it was trained on a single year of data instead of six years. Compared to the results from earlier, the model makes predictions less adept at capturing details, such as peaks and troughs. The effects of relative humidity was observed by using it to train the model. While performance was improved in some instances, the addition of relative humidity resulted in a loss in generalisability. The average performance across seasons was reduced suggesting overfitting in some regions. A total of four epochs were run with three epochs giving the best performance. Following this, wind speed was assessed. A total of two epochs were run and the performance of one and two epochs were compared to the reference model with three features. The performance across the four forecasts was not improved with the addition of wind speed. Ultimately, Model A was trained with the parameters in Figure 3.8. It was decided that Model A would be built using three features to predict air temperature at Kew Gardens and that the non-linear relationships would be explored in Model B.

3.5.2 Model B: Three Day Forecast

Initially, the objective was to assess the viability of predicting air temperature, relative humidity, and wind velocity on a 240-hour timescale. Instead, it became evident that much work needed to be done to transition from a 24-hour to a 240-hour model and the time constraints of the project made this infeasible. Therefore, the focus was shifted to 72-hour forecasts and obtaining accuracy that is comparable to existing numerical weather predictions systems, such as Met Office forecasts. Much of the process of fine-tuning Model B is identical to that of Model A, therefore, the description is limited to the steps that distinguish Model B from Model A.

A deeper model was used in this instance, which was comprised of a Bi-LSTM with more cells and an additional FNN in the second hidden layer. When selecting features, it was identified that using all available features as seen in Figure 3.11 produced the highest accuracy. The original dataset had a sampling frequency of one hour. Initially, the dataset was resampled to six hours using an averaging technique. However, reverting to the original, hourly sampling rate resulted in better performance. Model B was trained on the same dataset with the same split ratio for training, validation, and testing. A batch size of four was selected and one epoch was performed.

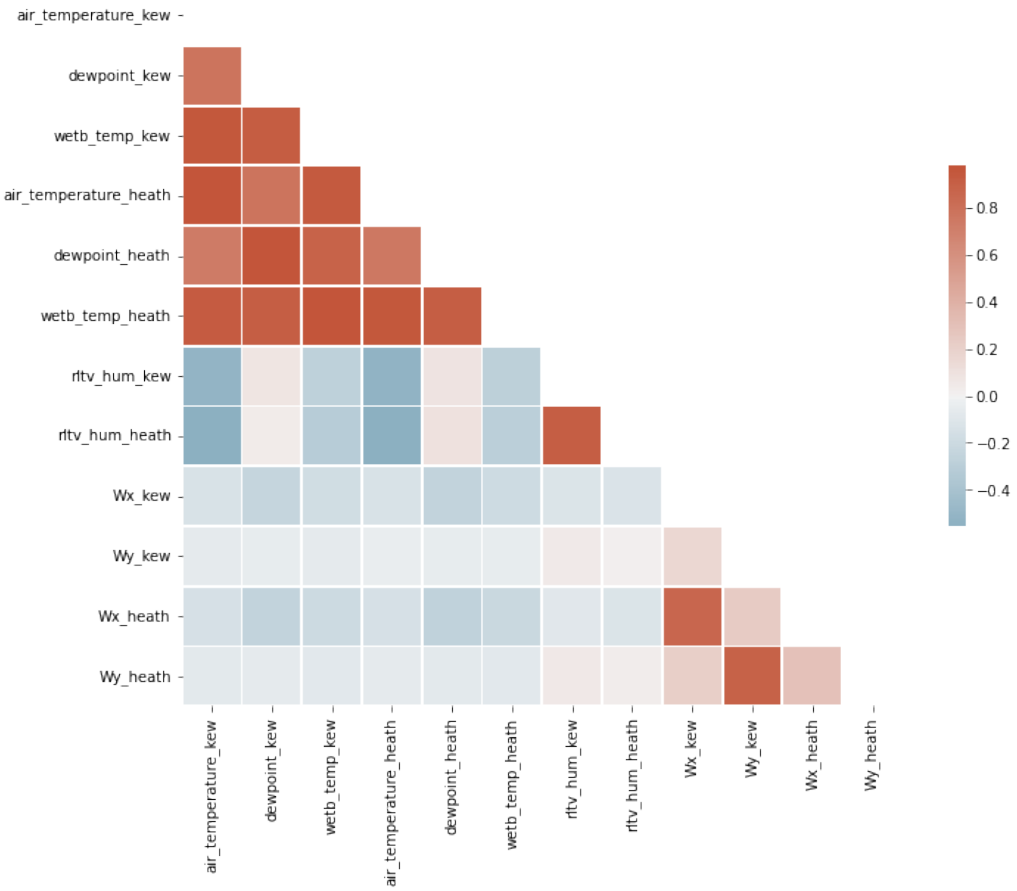

Figure 3.11: In addition to the features from Figure 3.8, wind speed, direction and relative humidity are also used in Model B

Chapter 4

Results and Discussions

4.1 One Day Temperature Forecast

The architecture of Model A is characterised in Table 4.1 and determines the number of calculations performed. The input layer shape is defined by the length of the context and the number of features. The hidden layer shape is defined by the batch size and number of Bi-LSTM units; 256 forward and 256 backward units. A batch size of 32 results in 1,151 observations per batch from a total of 36,825 training observations with any difference subtracted from the final batch. Finally, the output layer shape is defined by the number of features and batch size. The total number of parameters to be trained in the model is the sum of those in the hidden layer and output layer, totalling 541,702. The four gate equations 3.12 - 3.15 must be computed in each cell. Equations A.1 and A.2 in Appendix A describe how the number of parameters can be calculated.

Layer	**Type**	**Value**	**Shape**	**Parameters**
Input	-	-	(120 x 6)	0
Hidden	Bi-LSTM	Tanh activation function	(32 x 512)	538,624
Hidden	Dropout	0.25	(32 x 512)	0
Output	Linear	-	(32 x 6)	3,078
Total				**541,702**

Table 4.1: Architecture of the Bi-LSTM used in Model A, which includes the number and type of layers and the number of nodes in each layer

While the validation data is not explicitly used to train the model, it influences the weight adjustment as it is used to quantify the accuracy during training. A dropout layer is included to minimise the impact of overfitting by randomly setting the weight of 25% of the units in the hidden layer to zero. Dropout is a well-established technique in neural network modelling to overcome overfitting and is considered a more practical approach than regularisation, which is a common approach to reduce overfitting in traditional machine learning problems [29].

Parameter	Value
Context Length	120 hours
Gradient Optimisation	Adaptive moment estimation (ADAM)
Learning rate	0.001
Model optimised metric	Mean squared error
Performance metric	Root mean squared error
Epochs	2
Batch size	32
Runtime size	78 seconds
Train, validate, test ratios	0.7, 0.15 and 0.15

Table 4.2: Parameters used in Model A including number of epochs and optimiser settings

The training process was performed using Jupyter Notebook within a Google Colaboratory environment and a Tesla T4 graphical processing unit. The complete runtime was 78 second after which predictions could be made within 10 seconds. The maximum memory usage during training was 15.84 GB. The gradient descent optimiser used is the ADAM optimiser, which is known to generalise well and reduce training time. The learning rate defines the step size that the model is allowed to take when minimising the loss function with small learning rates taking long but providing better results. It is analogous to the step size, η, in equation 3.8.

It was concluded that our model is capable of making single-hour predictions that generalise well to new seasons as seen in Figure 3.10. Next, a comparison was made between the hourly single timestep and multi-timestep prediction models to assess the impact of error propagation as seen in Figure 4.1. The root mean squared error (E_2) is particularly insightful as it describes how well the entire prediction population performs with respect to the measured data while penalising outliers, unlike the mean absolute error (E_1), which is limited to expressing the cumulative error and averaging it. The maximum error (E_∞) highlights the single largest outlier but does not convey the frequency of occurrence.

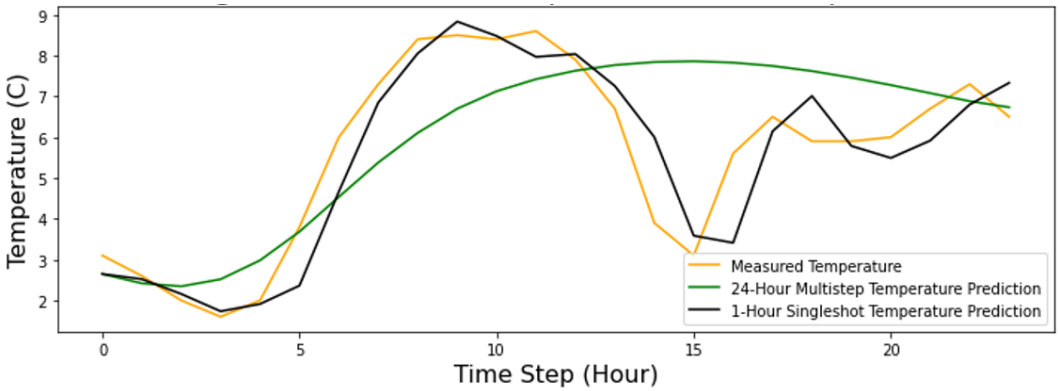

Figure 4.1: Hourly temperature predictions show excellent agreement with the measured temperature while the 24-hour prediction captures trend reasonably well, but fails to express the details

While both models have a context of 120 hours, the single timestep forecast uses measured data only while the multi-timestep forecast uses a combination of both. Initially, 120 measured and 0

	RMSE [$°C$]	**MAE** [$°C$]	**Max. Error** [$°C$]
Single timestep	0.86	0.63	2.19
Multi-timestep	1.74	1.33	4.76

Table 4.3: A comparison of performance between hourly and 24-hour predictions in Figure 4.1

predicted samples are used and by the final timestep, 97 measured and 23 predicted samples are used. The multi-timestep model prediction error according to all three metrics is approximately twice as large as the single timestep error as seen in Table 4.8. As expected, a longer forecast produces greater uncertainty, and the model performance deteriorates. A benefit of having a context length greater than the forecast length is that some measured data will always be used in making the prediction. However, the returns are diminished as the temporal gap between the measured data and forecast increases. A model with a larger context of 240 hours was observed to capture trend but failed to express the peaks and troughs accurately. The root mean, mean absolute and maximum errors across the entire single-hour training dataset were 0.89°C, 0.62°C and 12.81°C respectively.

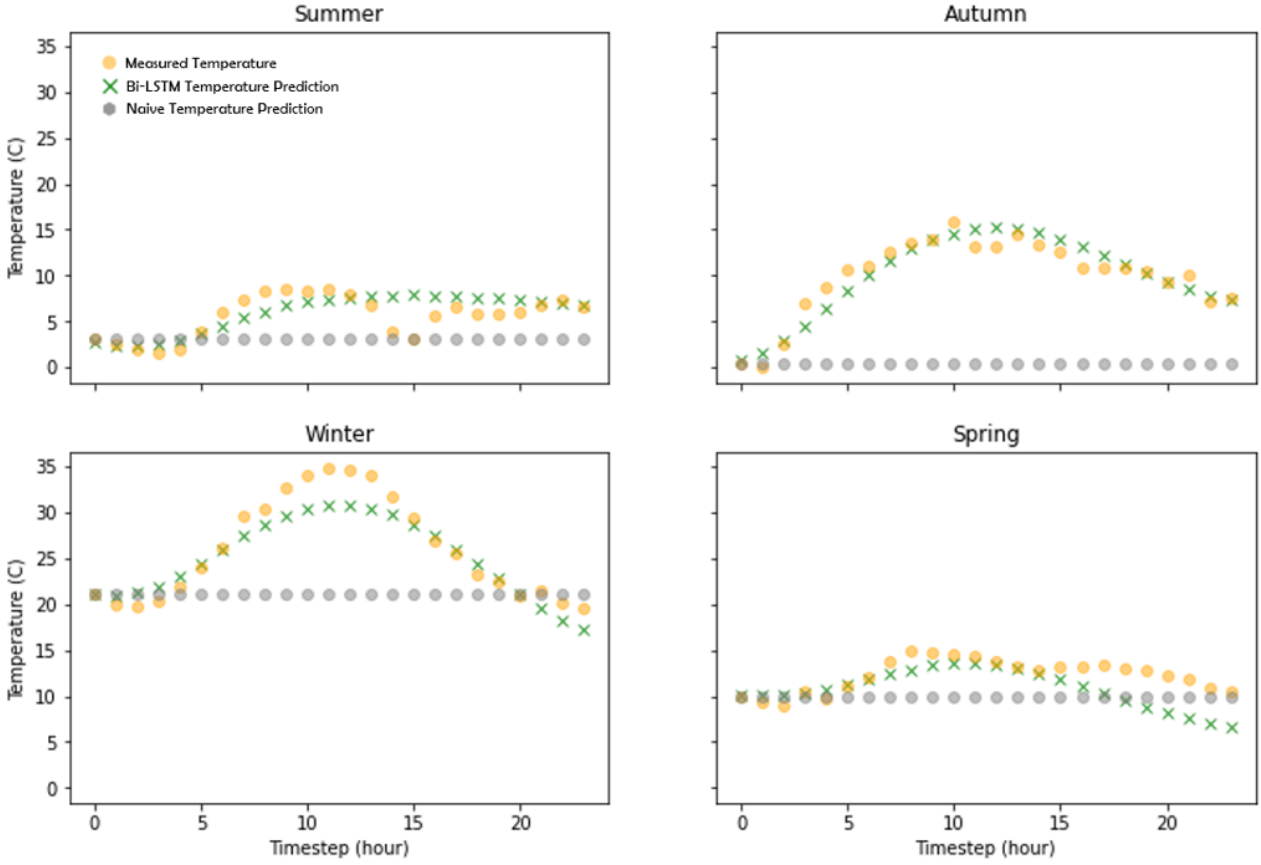

Figure 4.2: A 24-hour forecast of the air temperature at Kew Gardens, which illustrates the performance of the neural network model and naïve model across all four seasons

To quantify how well our 24-hour model generalises to different time periods and seasons, four prediction windows spaced 90 days apart are illustrated in Figure 4.2. In machine learning,

it is considered best practice to benchmark model performance against a naïve model. The naïve model uses the last measured temperature timestep for the entire 24-hour forecast, which simulates a case where only the current weather state is known, but nothing is known about the prior state and temperature is assumed constant. The naïve model is useful as it is system that assumes nothing about the future state and is completely uninformed. This creates a useful benchmark to assess performance against. The root mean squared errors confirm the neural network performs significantly better than the naïve model in all instances (Table 4.4) with an average error of 1.45°C and 6.00°C for the neural network and naïve forecast respectively.

	RMSE $[°C]$	**RMSE Naïve** $[°C]$	**MAE** $[°C]$	**Max. Error** $[°C]$
Summer	1.33 (0.91)	3.30 (2.27)	1.74 (1.20)	4.76 (3.27)
Autumn	1.12 (0.77)	10.4 (7.15)	1.36 (0.93)	2.39 (1.64)
Winter	1.64 (1.13)	7.30 (5.02)	2.03 (1.40)	4.01 (2.76)
Spring	1.73 (1.19)	3.00 (2.06)	2.25 (1.55)	4.24 (2.91)
Mean	1.45	6.00	1.84	3.85
Std. dev.	0.244	3.06	0.333	0.886

Table 4.4: Despite significant differences in the weather patterns over different seasons, the neural network is able to predict 24 hours with an average RMSE accuracy of 1.45°C compared to 6.00°C for the naïve model (Figure 4.2), with the values in parentheses normalised RMSE

To contextualise the performance in a practical way, the neural network was compared to performance metrics from the Met Office. The 24-hour predictions produced by the neural network were in 72.9% of all instances accurate to ±2°C. By comparison, the Met Office states 92.5% of its 24-hour temperature predictions are accurate to ±2°C while 92% of 24-hour wind speed predictions are within 5 knots [65]. Assuming the Met Office measurements used in this study were acquired with sensors of similar or superior resolution to those discussed in the methodology section, which have a resolution of ± 0.1 °C, the effect of resolution is expected to have little impact on our results in Table 4.4.

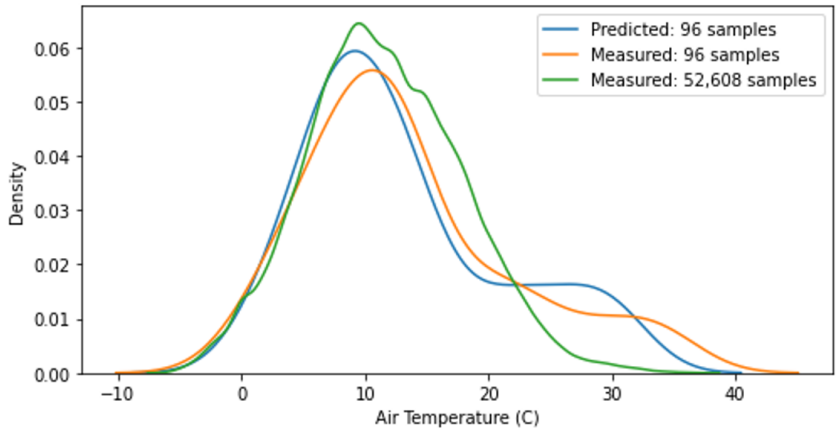

Figure 4.3: Temperature probability density functions at Kew Gardens. In addition to creating a benchmark distribution from the full temperature dataset of 52,608 samples, 96 predicted samples are compared to 96 measured samples

Three probability density functions were generated to compare the predicted and measured data. The 96 individual forecasts are derived from the four windows in Figure 4.2. These 96 points were used to compute a distribution function and are compared to a distribution using the measured temperature for the same period. The entire temperature dataset of 52,608 timesteps was used to create a benchmark. The 96-sample measured temperature peak is wider than the predicted peak indicating that predictions are conservative with both curves demonstrating bimodal behaviour. This bimodal behaviour is not present in the distribution from the full sample set, however, it is expected that the distribution function will change with the time of year and and the sample length used to create it.

The length of forecast and error were considered next with the aim of understanding how forecast skill deteriorates over multiple forecast timescales without adapting the model and parameters. The RMSE was calculated for 10 different forecast lengths ranging from one hour to seven days. Model A was used to generate all predictions in Figure 4.4 with all the initial parameters used consistently throughout. Each prediction length contains four windows with predictions for each season as per Figure 4.2. The root mean squared error was calculated for each of the four sets of predictions for each of the 10 forecast lengths.

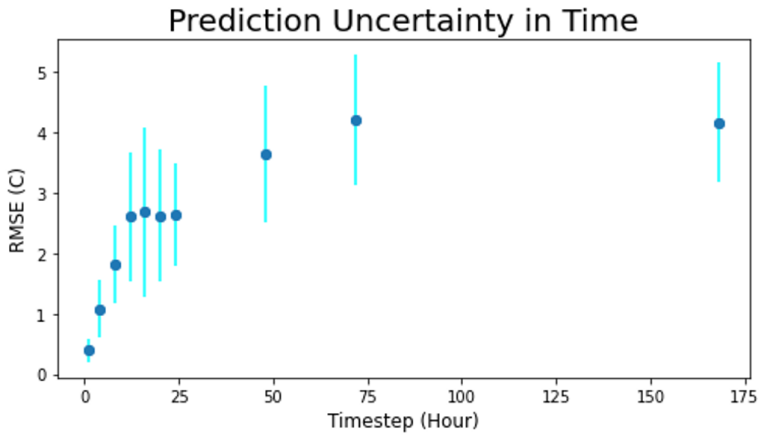

Figure 4.4: The forecast RMSE and variation is examined for 1, 2, 4, 8, 12, 16, 20, 24, 72 and 168 hour forecast lengths and is seen to increase rapidly beyond single hour predictions before stabilising around 24 hours

The RMSE mean and standard deviation are plotted against forecast length in Figure 4.4 to indicate uncertainty for increasing forecast lengths. For consistency, each prediction was run with a single epoch rather than attempting to optimise performance by identifying the most suitable number of epochs for each forecast length. The single hour prediction has the smallest mean and standard deviation, both of which increase as the forecast length increases, but become more stable after 24 hours. 1-24 hour predictions have a mean error less than 3°C. Beyond 24 hours, the prediction uncertainty continues to increase before rapidly converging around 4°C and stabilising. While there are many caveats to this information, it indicates that the model should not be used for predictions exceeding 24 hours and tells us much about the steady state value of predictions. Predictions must be well below 6.0°C RMSE to be meaningful.

4.2 Three Day Temperature, Relative Humidity and Wind Velocity Forecasts

After the 24-hour forecast optimisation process, a 240-hour forecast was attempted. As a baseline, the architecture from Model A was used, but with a longer context and more features. Initially, the data was resampled from one hour to six hours using mean resampling, which resulted in the one-sixth as many observations for the model to learn from. This led to an immediate degradation in the model accuracy due to the loss of training data and the data was reverted to the original number of samples to avoid information loss.

The forecast in Figure 4.5 was created using an unoptimised 240-hour model with features from Figure 3.11. It is evident that the model identifies some characteristics such as periodicity, but fails to make meaningful predictions. While the model would have undeniably benefited substantially from further optimisation, a decision was made to pursue optimisation of a 72-hour forecast instead of a 240-hour forecast to ensure the work remained within the scope of the project. In this approach, the architecture built upon that of Model A, but with the addition of a hidden linear layer with a ReLU activation function following the Bi-LSTM layer as it was shown to improve accuracy.

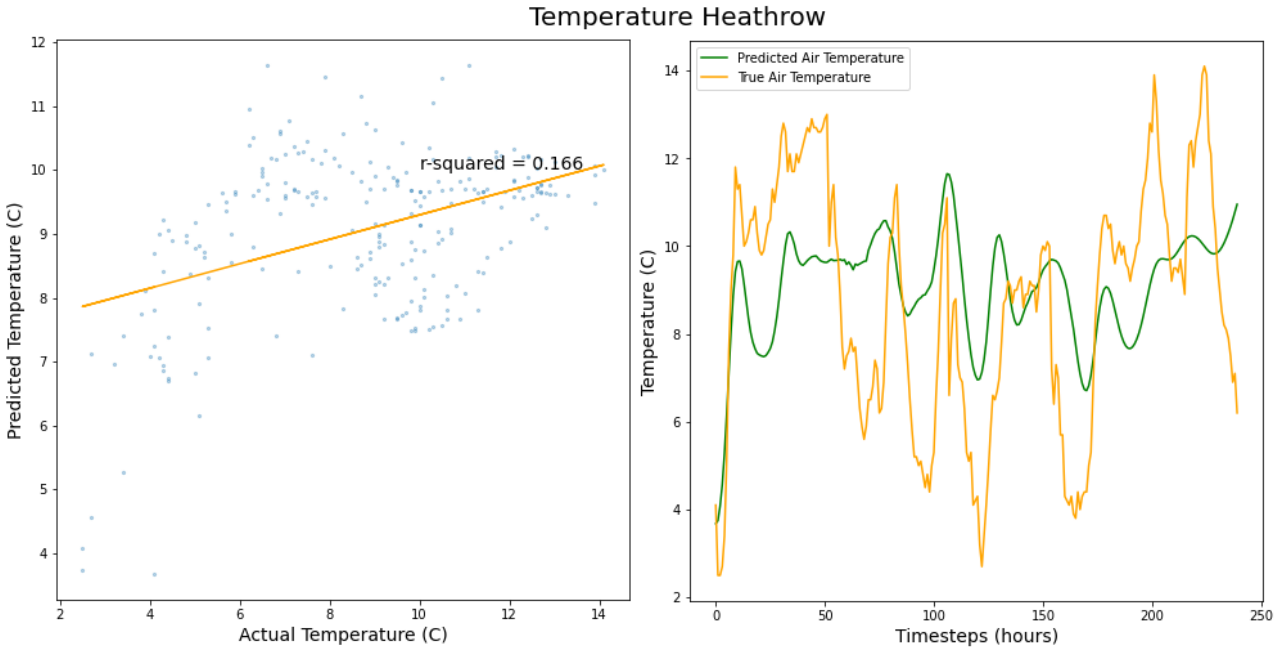

Figure 4.5: Temperature scatter plot (left) and line plot (right). With the confidence from Model A, predict of 240-hour air temperature was attempted

As with Model A, once a suitable 72-hour forecast architecture had been identified, the layer setup was recorded in Table 4.5. This table includes the full network including the number of parameters to be computed. The main difference is the addition of a linear layer within the hidden layer and a reduction in the dropout percentage to 10%. Furthermore, the hyperparameters used in the optimised model are recorded in Table 4.6.

Layer	Type	Value	Shape	Parameters
Input	-	-	(168 x 12)	0
Hidden	Bi-LSTM	Tanh activation function	(4 x 640)	852,480
Hidden	Linear	ReLU activation function	(4 x 256)	164,096
Hidden	Dropout	0.10	(4 x 256)	0
Output	Linear	-	(4 x 12)	3,048
Total				**1,019,660**

Table 4.5: Architecture of Bi-LSTM model, Model B, which includes the number and type of layers and the number of nodes in each layer

As before, an increase in the number of epochs resulted in a reduction in the root mean squared error and increase of the r-square value. However, there was no direct correlation between optimisation of these two parameters and how the 72-hour forecast performed over different time periods. Therefore, once a capable architecture was identified, a similar trial-and-error approach began to optimise the hyperparameters and context length with the mean RMSE value from the four windows used to make decisions. One single epoch was run with additional epochs reducing the accuracy. Initially, 120 hours were used for the context length but later changed to 168 hours as this gave optimal performance. After upwards of twenty iterations with different conditions, the hyperparameters listed in Table 4.6 resulted in the best performance.

As expected, once the model was trained it was possible to make new predictions very rapidly, in this case it was possible to predict up to 240 hours within 15 seconds. The single-timestep hourly prediction RMSE was 0.94 °C, MAE 0.68°C and maximum error 14.94°C when calculated over the entire test dataset. While these numbers are quite comparable to the single-hour predictions generated in Model A, the model did not perform quite as well over three days as one day. Naturally, this is to be expected as the forecast window is three times longer and the likelihood of error propagation is much higher. Furthermore, as the forecast length increases, the greater the chaotic tendencies will be and higher the impact of turbulence and uncertainty.

Parameter	Value
Context Length	168 hours
Gradient Optimisation	Adaptive Moment Estimation (ADAM)
Learning rate	0.001
Loss: model training	Mean squared error
Metrics: test data evaluation	Root mean squared error
Epochs	1
Batch size	4
Run time	187 seconds
Train, validate and test split	0.7, 0.15 and 0.15

Table 4.6: The finalised hyperparameters used to train Model B including the number of epochs and optimiser settings

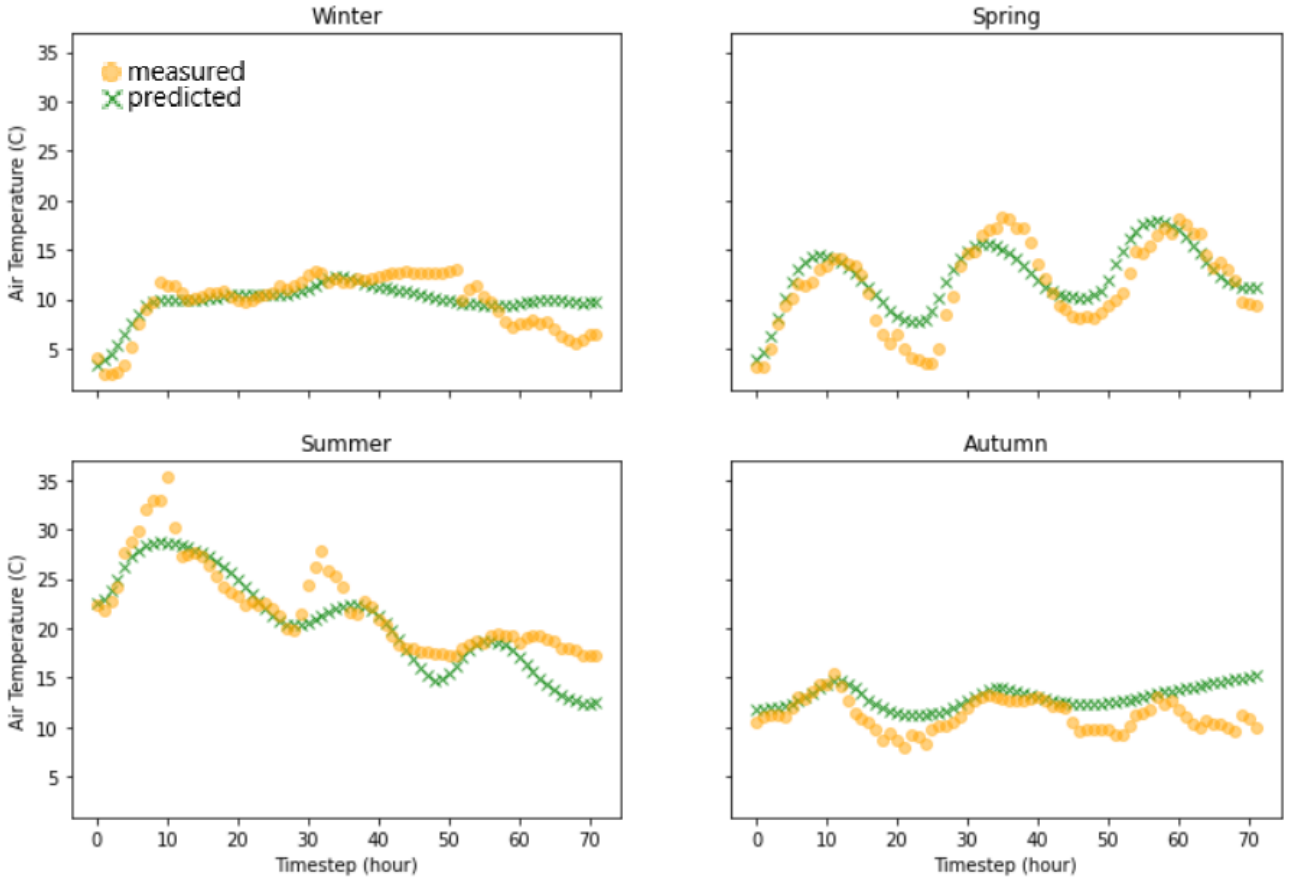

Figure 4.6: A three day forecast of the air temperature at Heathrow, which illustrates the performance of the neural network model across all four seasons

The four windows in Figure 4.6 illustrate how the Bi-LSTM and linear model is highly capable of making predictions with excellent generalisability across different periods and seasons. Since the Met Office does not publish three day forecast accuracy for ±2°C, it is not possible to make a good comparison as with Model A. Therefore, we are forced to draw comparisons between our three day forecast and one day forecast and their corresponding root mean square errors. The single day forecast resulted in an RMSE mean and standard deviation of 1.45°C and 0.244°C respectively, while the three day forecast resulted 2.26°C and 0.316°C respectively. 79.5% of the temperature forecasts are within ±3°C when making a 72-hour forecast.

	RMSE $[°C]$		**MAE** $[°C]$		**Max. Error** $[°C]$	
	Kew G.	**Heathrow**	**Kew G.**	**Heathrow**	**Kew G.**	**Heathrow**
Winter	2.22 (0.78)	1.80 (0.63)	1.79 (0.63)	1.44 (0.51)	6.25 (2.2)	4.11 (1.45)
Autumn	3.02 (1.06)	2.27 (0.80)	2.51 (0.88)	1.89 (0.67)	7.64 (2.70)	5.24 (1.85)
Summer	3.41 (1.20)	2.69 (0.95)	2.80 (0.99)	2.03 (0.72)	7.10 (2.50)	6.66 (2.35)
Spring	2.70 (0.95)	2.31 (0.82)	2.12 (0.75)	1.87 (0.66)	5.25 (1.85)	5.31 (1.87)
Mean	2.83	2.26	2.31	1.81	6.56	5.33
Std. dev.	0.436	0.316	0.383	0.221	0.910	0.904

Table 4.7: The neural network is able to create 72-hour forecasts with with an average RMSE accuracy of 2.26°C at Heathrow as seen in Figure 4.6, with normalised RMSE in parentheses

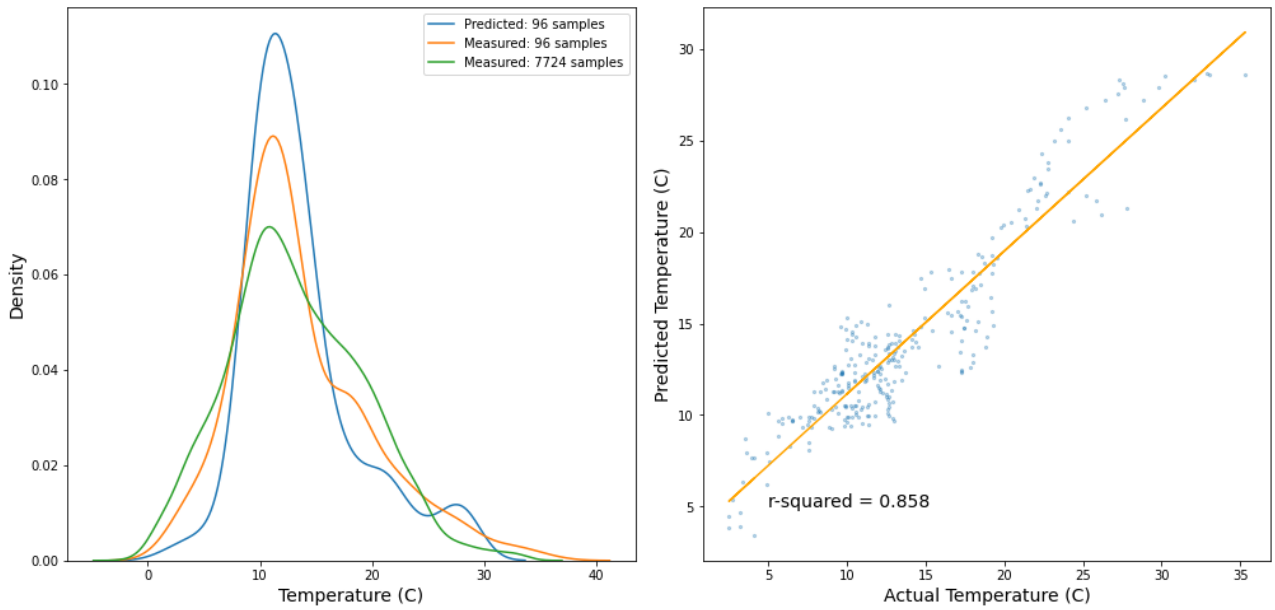

Figure 4.7: Temperature probability density functions (left) and scatter plot (right) at Heathrow. While both distributions exhibit good agreement, the predicted function has a taller peak and narrower base indicating extreme temperatures are being underestimated. The r-squared value quantifies the variation between measured and predicted values

Next, the relative humidity was predicted using the same set of features. The model takes in all features from both locations resulting in six unique features and 12 features in total. As before, it is possible to generate a prediction for any one of the 12 features introduced to the model in training. The objective was to predict relative humidity without retraining the model or the need to make feature-specific adjustments. While the model does take all 12 inputs into consideration during training and seeks to minimise the loss function with respect to all 12 features, the performance arising from this approach does not necessarily translate into good generalisability across all timescales. When training the model, TensorFlow uses the weighted sum of all 12 features when minimising the loss, which means the algorithm assigns different levels of importance to each feature. 237 relative humidity data points were noted as missing from Kew Gardens and 80 from Heathrow. While these are relatively small compared to the entire population, these missing data had to be imputed.

The model optimises the loss function based on the feedback from a single timestep and not the full forecast window of 72 hours. A single-timestep prediction with a low loss does not necessarily result in high accuracy over 72 predictions. Since, during the training of Model B, the objective was to optimise the 72-hour air temperature predictions, there was no guarantee that this performance would translate into comparable performance for relative humidity. The accuracy of the results in Figure 4.8 are simply a byproduct of the process to optimise the air temperature. There is a strong expectation, that, had the relative humidity been the focus of the optimisation, that forecast skill would have seen considerable improvement; possibly with minimal changes to the training strategy while avoiding modification of the model architecture.

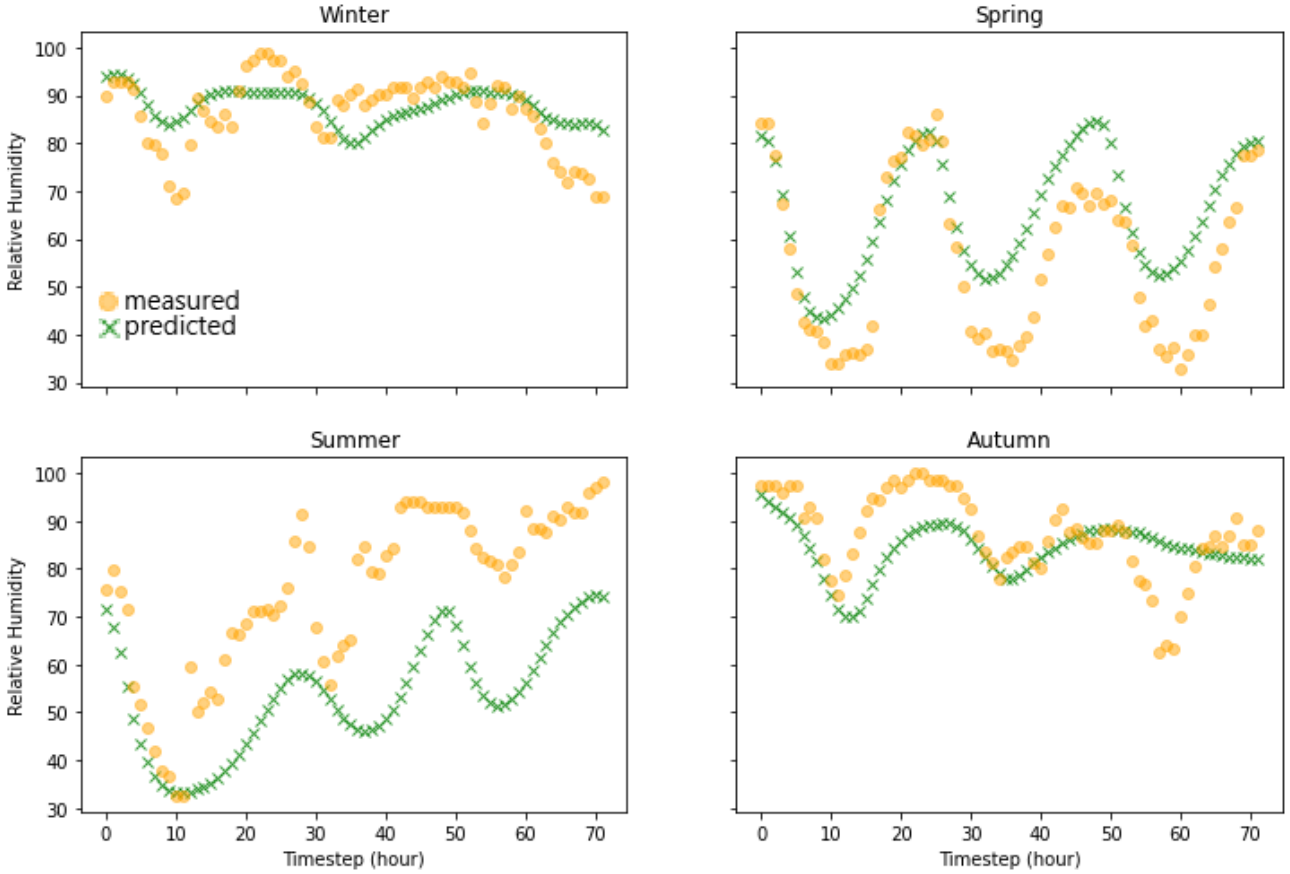

Figure 4.8: The first, second and fourth windows are able to capture the trends in relative humidity at Heathrow while there is noticeable degradation of performance in window three

	RMSE [%]		**MAE [%]**		**Max. Error [%]**	
	Kew G.	**Heathrow**	**Kew G.**	**Heathrow**	**Kew G.**	**Heathrow**
Winter	8.78	7.46	6.33	5.4	22.9	20.6
Autumn	9.48	8.25	7.30	6.21	21.9	19.6
Summer	28.0	29.1	22.2	23.43	58.6	61.6
Spring	11.9	11.5	8.43	8.51	36.2	33.4
Total	58.1	56.3	44.3	43.6	139.6	135.2
Average	14.5	14.0	11.1	10.9	34.9	33.8

Table 4.8: The neural network is able to create 72-hour relative humidity forecasts with an average RMSE accuracy of 14% at Heathrow

Finally, longitudinal wind velocity was predicted using the same features and hyperparameters as before. It must be noted that 2072 wind speed and direction data points were missing at Kew Gardens and 665 of each at Heathrow. Many of these occurred subsequently suggesting the sensors were out of operation for several days at a time. Since the data was imputed with the mean of each feature column, there were longer periods where the wind speed is constant. Naturally, this is extremely problematic for our model as it tries to learn this non-representative behaviour. This flatness is reflected in the non-responsive predictions seen in Figure 4.9.

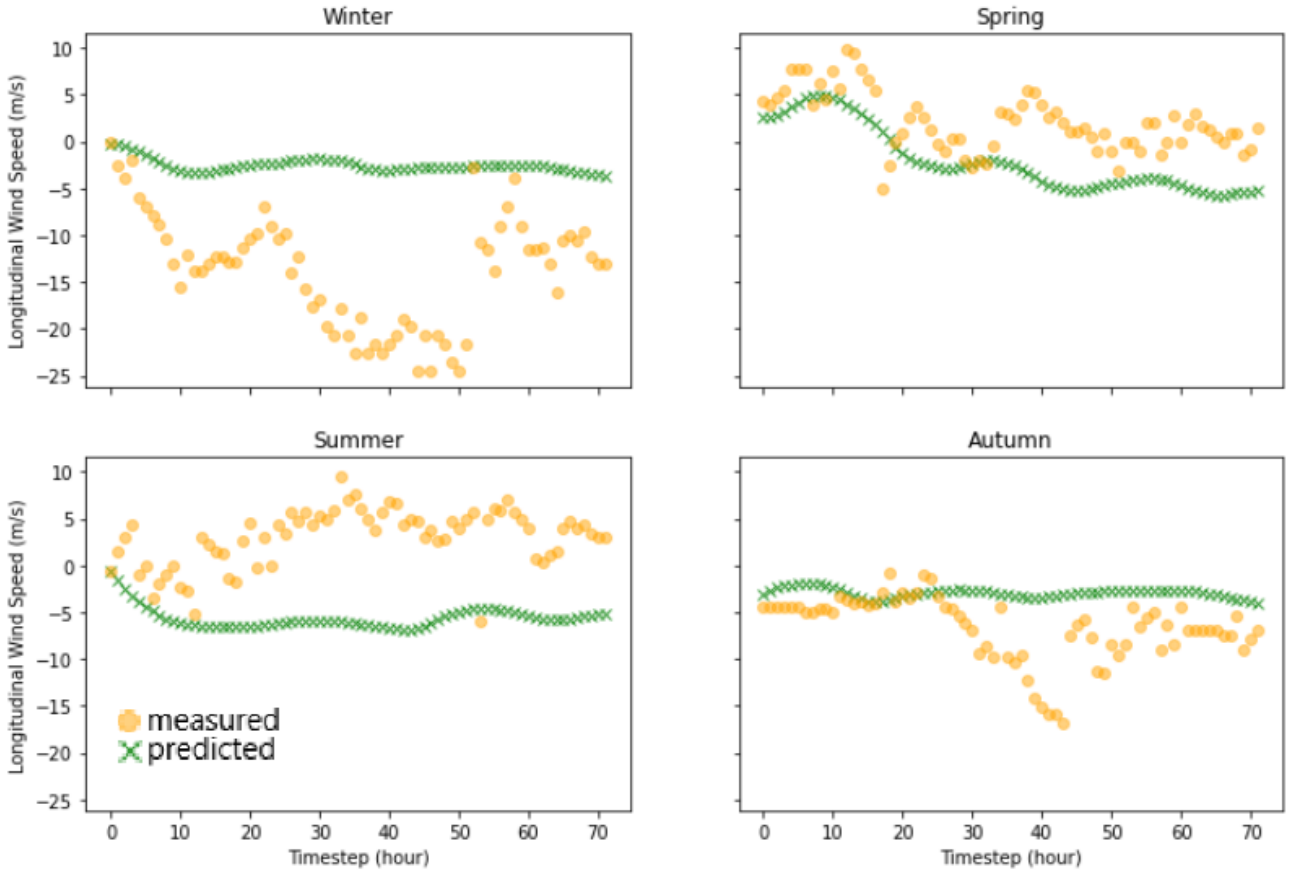

Figure 4.9: Longitudinal wind speed is predicted at Heathrow but fails to capture the complexity of the turbulence and unpredictability of wind

As is the case with relative humidity, the model was trained with the optimisation of air temperature performance in mind. As such, the wind speed was never considered in the 72-hour forecast optimisation process. Therefore, it can be concluded that a better imputation method must be used or a different dataset with fewer missing values. Only then can we accurately assess how capable our model is of making predictions on wind behaviour.

Chapter 5

Conclusions

At the start of the research project in Autumn 2021, the primary objective was to assess the feasibility of machine learning to predict the complex relationships that govern weather evolution. Over the course of the research project, it became apparent that data-driven weather forecasting was gaining significant traction within the meteorological community, as well as within industry, as numerous citations in this research project were published after the start of the project. The focus gradually shifted from demonstrating data-driven weather prediction is possible, to designing a lightweight and flexible model. A computationally efficient model has enormous value in the context of ensemble modelling, whereby numerous predictions are run to improve model performance. Ensemble modelling is currently a limiting factor in numerical weather predictions as the models are run up to 50 times to achieve desired performance. In the proof-of-concept stages of deep learning weather forecasting, flexibility is critical. The model was designed with maximum flexibility in mind and is capable of rapidly forecasting any weather parameter at any location within the training dataset, and is able to generate a forecast of any duration. Another novelty of this approach is the ability of the model to make spatio-temporal predictions using readily available and open source data to predict air temperature and relative humidity in a purely data-driven approach. In contrast, many earlier studies required varying degrees of data assimilation or used a hybrid model.

A total of two models were trained and used to predict air temperature, relative humidity and wind speed. The dataset used to train the models contained six years of historical weather observations from Kew Garden and Heathrow weather observation stations in London. The objective of having multiple locations is to create a spatial, geographical and topographical representation for the model to learn from. As the two weather observation stations are positioned 11km apart, it is expected that they would share similar weather characteristics. Many of the differences in wind speed and humidity could be explained by local land features and artificial structures. Kew Gardens is positioned near the river Thames in a built-up area while the nearest body of water to Heathrow is several kilometres away. Furthermore, the Heathrow observation station is situated within the airport boundaries with few obstructions.

Model A is a 24-hour prediction network designed to predict air temperature. This model was intended to demonstrate proof of concept and was trained with wet bulb, air and dew point temperatures. The model training time was 78 seconds making it possible to iterate through many different feature and hyperparameter combinations and identify the best combinations that describe 24-hour air temperature behaviour. Model A achieved its objective of establishing a baseline for further predictions. It successfully demonstrated that air temperature could be predicted with reasonable accuracy compared to the Met Office and created a benchmark for building the more versatile model, Model B. Model A predicted the air temperature within a range of 2°C in 72.9% of instances with a maximum error of 3.85°C.

Model B is a 72-hour prediction network that attempted to predict air temperature, relative humidity and wind speed. Despite a three-fold increase in the forecast length, the model was able to accurately predict air temperature with an RMSE of 2.26°C at Heathrow and was able to predict the temperature accurately to within ±3°C in 79.5% of instance. It was able to predict the relative humidity in the same location with an RMSE of 14%. However, this must be caveated by stating that several hundred humidity records were missing and that Model B was optimised with respect to air temperature. Equally, the model assigns various degrees of importance to each feature. The impact of feature weighting on prediction performance was not explored in this project as it was out-of-scope. We attempted to forecast wind velocity but concluded that in the presence of thousands of missing data points, it was not possible to assess performance in a meaningful way. The impact of missing wind speed data in our 72-hour temperature and humidity forecasts has not been thoroughly investigated.

Numerical weather prediction, on the other hand, has been in development for decades. It requires enormous amounts of data and data assimilation, has significantly greater spatial coverage and must be run on prohibitively expensive supercomputers. While the models did not exhibit the same forecast skill as the Met Office model, they present many benefits. The single biggest consideration is accuracy-to-cost ratio. Comparing the neural network temperature model to the Met Office temperature model, the relative accuracy is 0.79. The Met Office spent £256.7m in 2021, which are the bare minimum operating costs and do not account for numerous infrastructure and additional costs. The neural network, by comparison, was trained on a cloud-based GPU at a monthly subscription cost of £9.72 and a total cost of £116.64. It is clear that costs associated with traditional weather modelling are orders of magnitude greater.

The single biggest drawback to the neural network is its high sensitivity to errors, and when the single-timestep prediction is looped to create longer forecasts, error propagation occurs resulting in unpredictable behaviour. To compensate for this phenomenon, the approach was to optimise the long-term forecasts iteratively rather than relying on the traditional stopping criteria, such as training and validation loss metrics. Simply, it is important to minimise the loss function, but the lowest training and validation losses are certainly not guaranteed to result in the best long-term forecast after making predictions with multiple loops. Presently, it is not possible to fully explain the neural network's decisions and addressing explainability is crucial.

5.1 Future Work

The neural network models have shown enormous promise considering this was a machine learning approach with no understanding of the physics of weather. Naturally, there is still significant opportunities for improvement. Four avenues are discussed as potential future work, which are believed to have the highest return on investment and are listed in the perceived order of impact.

In addition to the flexible models that can make predictions of any length, the next step would be to develop multi-timestep models for predetermined lengths and train the model with respect to each forecast length. This will overcome the numerous obstacles faced in this research project associated with the trail-and-error approach. It is acknowledged that such a model will surely have its own unique challenges and might cause it to run into issues with, for example, generalisability or convergence.

It is clear from the neural network models that, despite making excellent air temperature predictions with limited geospatial information, the description of the atmospheric state must be expanded considerably. This is especially important in the context of loss of data with only two weather stations. It is anticipated that incorporating weather data from many more locations within several kilometres to several hundred kilometres would improvement forecast skill. Naturally, with many locations comes the challenge that traditionally weather forecasting faces, namely, the need to assimilate data.

While it is clear that numerical weather forecasting is sensitive to the quality of the assimilated data, the same is not necessarily true for deep learning forecasting. With a vast representation of the atmospheric state, the next step would be to investigate how ensemble modelling affects performance. Adding more spatial awareness to the model will naturally result in much longer runtimes. Therefore, it is of enormous benefit that our models can be trained quickly and bearing in mind that our models were trained on a single GPU. Parallel computing is the obvious solution for scaling computational power and still negates the need for prohibitively expensive supercomputers.

Finally, investigating the effects of additional weather parameters on forecast skill should be considered as it was shown that more information can improve model accuracy. This is especially true for deep learning models. However, there are often hardware limitations that limit data acquisition and these must be considered. Finally, it would be invaluable to extend the prediction outputs to include cloud-cover and precipitation.

Bibliography

[1] Lora Shinn. Renewable energy: The clean facts, 2022. URL https://www.nrdc.org/stories/renewable-energy-clean-facts. [Accessed 05/05/2022].

[2] EU Science Hub. Droughts in europe in july 2022: Almost half of the eu + uk territory at risk, 2022. URL https://joint-research-centre.ec.europa.eu/jrc-news/droughts-europe-july-2022-almost-half-eu-uk-territory-risk-2022-07-18_en. [Accessed 04/07/2022].

[3] Met Office. Annual report and accounts. Report, 2021. URL https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/library-and-archive/library/publications/corporate/annual_report_2021_optimised.pdf.

[4] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[5] European Centre for Medium-Range Weather Forecasts. Data assimilation and observations. Report, 2022. URL https://www.ecmwf.int/en/research/data-assimilation/observations. [Accessed 01/08/2022].

[6] Zhaoxia Pu Kalnay and Eugenia. Numerical weather prediction basics: Models, numerical methods, and data assimilation. *Handbook of Hydrometeorological Ensemble Forecasting. Springer, Berlin, Heidelberg*, 2018.

[7] Meteorological Office. Unified model, 1992. URL https://www.metoffice.gov.uk/research/approach/modelling-systems/unified-model/index. [Accessed 17/11/2021].

[8] ECMWF. How to pin down uncertainty in weather forecasting, 2017. URL https://www.ecmwf.int/en/about/media-centre/news/2017/how-pin-down-uncertainty-weather-forecasting. [Accessed 18/02/2022].

[9] Met Office. The met office ensemble system, 2022. URL https://www.metoffice.gov.uk/research/weather/ensemble-forecasting/mogreps. [Accessed 05/05/2022].

[10] Julia Slingo and Tim Palmer. Uncertainty in weather and climate prediction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1956):4751–4767, 2011.

[11] C. S. Witham M. C. Hort E. L. Kendall, S. J. Leadbetter. Grenfell tower fire: modelling deposition of smoke particulates using name. Report, 2019.

[12] IBM. Recurrent neural networks, 2020. URL https://www.ibm.com/cloud/learn/recurrent-neural-networks. [Accessed 05/02/2022].

[13] An Tang, Roger Tam, Alexandre Cadrin-Chênevert, Will Guest, Jaron Chong, Joseph Barfett, Leonid Chepelev, Robyn Cairns, J. Ross Mitchell, Mark D. Cicero, Manuel Gaudreau Poudrette, Jacob L. Jaremko, Caroline Reinhold, Benoit Gallix, Bruce Gray, Raym Geis, Timothy O'Connell, Paul Babyn, David Koff, Darren Ferguson, Sheldon Derkatch, Alexander Bilbily, and Wael Shabana. Canadian association of radiologists white paper on artificial intelligence in radiology. *Canadian Association of Radiologists Journal*, 69(2):120–135, 2018.

[14] M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadtler. Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200097, 2021.

[15] Gang Chen. A gentle tutorial of recurrent neural network with error backpropagation. Report, Cornell University, 2018. URL https://arxiv.org/pdf/1610.02583.pdf.

[16] Yuying Sun, Wei Wang, Yaohua Zhao, and Song Pan. Predicting cooling loads for the next 24 hours based on general regression neural network: Methods and results. *Advances in Mechanical Engineering*, 5(0):954185, 2013.

[17] R. Meenal, D. Binu, K. C. Ramya, Prawin Angel Michael, K. Vinoth Kumar, E. Rajasekaran, and B. Sangeetha. Weather forecasting for renewable energy system: A review. *Archives of Computational Methods in Engineering*, 29(5):2875–2891, 2022.

[18] Rayda Ben Ayed and Mohsen Hanana. Artificial intelligence to improve the food and agriculture sector. *Journal of Food Quality*, 2021:1–7, 2021.

[19] Qi Fu, Dan Niu, Zengliang Zang, Junhao Huang, and Li Diao. Multi-stations' weather prediction based on hybrid model using 1d cnn and bi-lstm. IEEE.

[20] Chengcheng Chen, Qian Zhang, Mahsa H. Kashani, Changhyun Jun, Sayed M. Bateni, Shahab S. Band, Sonam Sandeep Dash, and Kwok-Wing Chau. Forecast of rainfall distribution based on fixed sliding window long short-term memory. *Engineering Applications of Computational Fluid Mechanics*, 16(1):248–261, 2022.

[21] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11), 2020.

[22] Peter Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431–3444, 2008.

[23] Casey Crownhart. How two new superscomputers will improve weather forecasts, 2021. URL `https://www.technologyreview.com/2021/10/27/1036815/supercomputers-national-weather-service-forecasts/`. [Accessed 02/07/2022].

[24] Press Office. The cray xc40 supercomputing system, 2020. URL `https://www.metoffice.gov.uk/about-us/press-office/news/corporate/2020/supercomputer-funding-2020`. [Accessed 01/12/2021].

[25] Debneil Saha Roy. Forecasting the air temperature at a weather station using deep neural networks. *Procedia Computer Science*, 178:38–46, 2020.

[26] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, R. Prudden, A. Mandhane, A. Clark, A. Brock, K. Simonyan, R. Hadsell, N. Robinson, E. Clancy, A. Arribas, and S. Mohamed. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

[27] Pradeep Hewage, Marcello Trovati, Ella Pereira, and Ardhendu Behera. Deep learning-based effective fine-grained weather forecasting model. *Pattern Analysis and Applications*, 24(1):343–366, 2021.

[28] Press Office. Up to £1.2billion for weather and climate supercomputer, 2020. URL `https://www.metoffice.gov.uk/about-us/press-office/news/corporate/2020/supercomputer-funding-2020`. [Accessed 05/05/2022].

[29] Cliff Mass. The mathematics of weather forecasting. [Accessed 12/12/2021], 2020. URL `https://sites.math.washington.edu/~mathcircle/mathhour/talks_2020/MMH4_Mass_slides.pdf`.

[30] Chris G; Roulstone Ian Rihan, Fathalla A; Collier. Four-dimensional variational data assimilation for doppler radar wind data. *Journal of Computational and Applied Mathematics*, 176(1):15–34, 2005.

[31] Ashesh Chattopadhyay, Mustafa Mustafa, Pedram Hassanzadeh, Eviatar Bach, and Karthik Kashinath. Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based u-net in a case study with era5. *Geoscientific Model Development*, 15(5):2221–2237, 2022.

[32] Sebin Park, Myeong-Seon Gil, Hyeonseung Im, and Yang-Sae Moon. Measurement noise recommendation for efficient kalman filtering over a large amount of sensor data. *Sensors*, 19(5):1168, 2019.

[33] Sylvie Malardel and Nils P. Wedi. How does subgrid-scale parametrization influence nonlinear spectral energy fluxes in global nwp models? *Journal of Geophysical Research: Atmospheres*, 121(10):5395–5410, 2016.

[34] Humphrey W. Lean, Peter A. Clark, Mark Dixon, Nigel M. Roberts, Anna Fitch, Richard Forbes, and Carol Halliwell. Characteristics of high-resolution versions of the met office unified model for forecasting convection over the united kingdom. *Monthly Weather Review*, 136(9):3408–3424, 2008.

[35] Matthew Chantry, Hannah Christensen, Peter Dueben, and Tim Palmer. Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft ai. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200083, 2021.

[36] Andrew I Barrett, Suzanne L Gray, Daniel J Kirshbaum, Nigel M Roberts, David M Schultz, and Jonathan G Fairman. The utility of convection-permitting ensembles for the prediction of stationary convective bands. *Monthly Weather Review*, 144(3):1093–1114, 2016.

[37] Bilin Shao, Dan Song, Genqing Bian, and Yu Zhao. Wind speed forecast based on the lstm neural network optimized by the firework algorithm. *Advances in Materials Science and Engineering*, 2021:1–13, 2021.

[38] Jordan G. Powers, Joseph B. Klemp, William C. Skamarock, Christopher A. Davis, Jimy Dudhia, David O. Gill, Janice L. Coen, David J. Gochis, Ravan Ahmadov, Steven E. Peckham, Georg A. Grell, John Michalakes, Samuel Trahan, Stanley G. Benjamin, Curtis R. Alexander, Geoffrey J. Dimego, Wei Wang, Craig S. Schwartz, Glen S. Romine, Zhiquan Liu, Chris Snyder, Fei Chen, Michael J. Barlage, Wei Yu, and Michael G. Duda. The weather research and forecasting model: Overview, system efforts, and future directions. *Bulletin of the American Meteorological Society*, 98(8):1717–1737, 2017.

[39] Massimo Bonavita Simon Lang, Elias Holm and Yannick Tremolet. A 50-member ensemble of data assimilations. (158):27–29, 2019. URL `https://www.ecmwf.int/node/18883`.

[40] Sebastian Scher and Gabriele Messori. Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, 144(717): 2830–2841, 2018.

[41] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[42] Charles Lin, Slavko Vasić, Alamelu Kilambi, Barry Turner, and Isztar Zawadzki. Precipitation forecast skill of numerical weather prediction models and radar nowcasts. *Geophysical Research Letters*, 32(14), 2005.

[43] K. Ochiai, H. Suzuki, K. Shinozawa, M. Fujii, and N. Sonehara. Snowfall and rainfall forecasting from weather radar images with artificial neural networks. IEEE, 1995.

[44] Xiaoming Ma, Cong Fang, and Junping Ji. Prediction of outdoor air temperature and humidity using xgboost. *IOP Conference Series: Earth and Environmental Science*, 427 (1):012013, 2020.

[45] Jaroslav Frnda, Marek Durica, Jan Rozhon, Maria Vojtekova, Jan Nedoma, and Radek Martinek. Ecmwf short-term prediction accuracy improvement by deep learning. *Scientific Reports*, 12(1), 2022.

[46] Pin Wang, En Fan, and Peng Wang. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, 141:61–67, 2021.

[47] Alberto Gasparin, Slobodan Lukovic, and Cesare Alippi. Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, 7(1): 1–25, 2022.

[48] C. Narendra Babu and B. Eswara Reddy. Predictive data mining on average global temperature using variants of arima models. In *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, pages 256–260.

[49] Casper, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *arXiv pre-print server*, 2020. URL `https://arxiv.org/abs/2003.12140`.

[50] S. Karatasou, M. Santamouris, and V. Geros. Modeling and predicting building's energy use with artificial neural networks: Methods and results. *Energy and Buildings*, 38(8): 949–958, 2006.

[51] Vladlen Koltun Shaojie Bai, J. Zico Kolter. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv*, abs/1803.01271, 2018. URL `https://dx.doi.org/10.48550/arxiv.1803.01271`.

[52] Jonathan A. Weyn, Dale R. Durran, Rich Caruana, and Nathaniel Cresswell-Clay. Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, 13(7), 2021.

[53] Jürgen Schmidhuber Sepp Hochreiter. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

[54] C V Akhila. A survey on collaborative learning approach for speech and speaker recognition. AIJR Publisher.

[55] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3681–3688, 2019.

[56] 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*, 378:686–707, 2019.

[57] Jürgen Schmidhuber Gers, Felix A and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[58] Peter Huthwaite. Me4 machine learning. 2020.

[59] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[60] Olga Dombrowski, Harrie-Jan Hendricks Franssen, Cosimo Brogi, and Heye Reemt Bogena. Performance of the atmos41 all-in-one weather station for weather monitoring. *Sensors*, 21(3):741, 2021.

[61] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. SciPy. URL `https://dx.doi.org/10.25080/majora-92bf1922-011`.

[62] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3): e0194889, 2018.

[63] Jian Qi Wang, Yu Du, and Jing Wang. Lstm based long-term energy consumption prediction with periodicity. *Energy*, 197:117197, 2020.

[64] Pavlo M. Radiuk. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. 2017.

[65] Met Office. How accurate are our public forecasts?, 2022. URL `https://www.metoffice.gov.uk/about-us/what/accuracy-and-trust/how-accurate-are-our-public-forecasts`. [Accessed 30/08/2022].

Appendices

Appendix A

Code available at: https://github.com/gzenkner

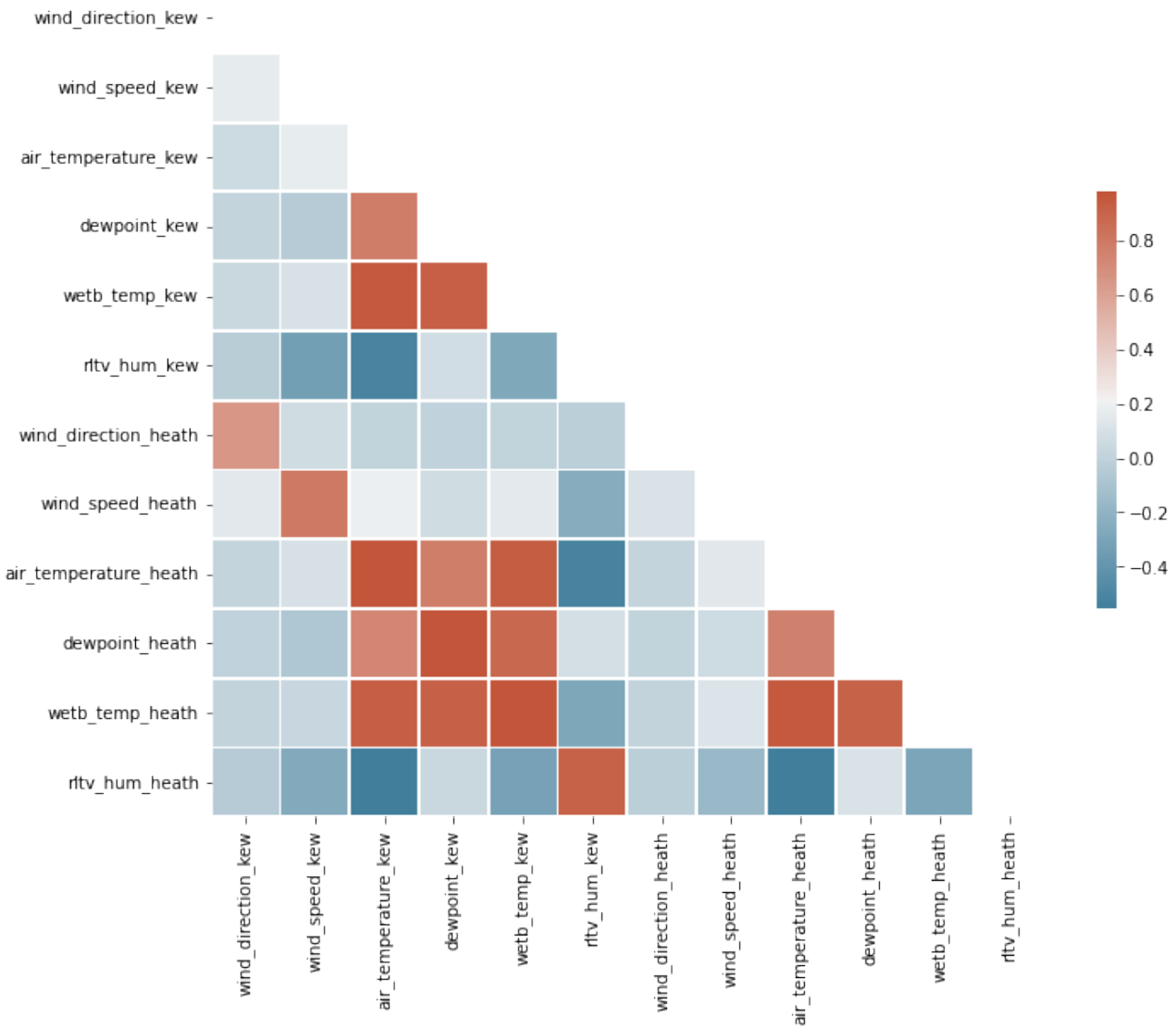

Figure A.1: A Pearson correlation plot containing all the original weather parameters before data processing

.

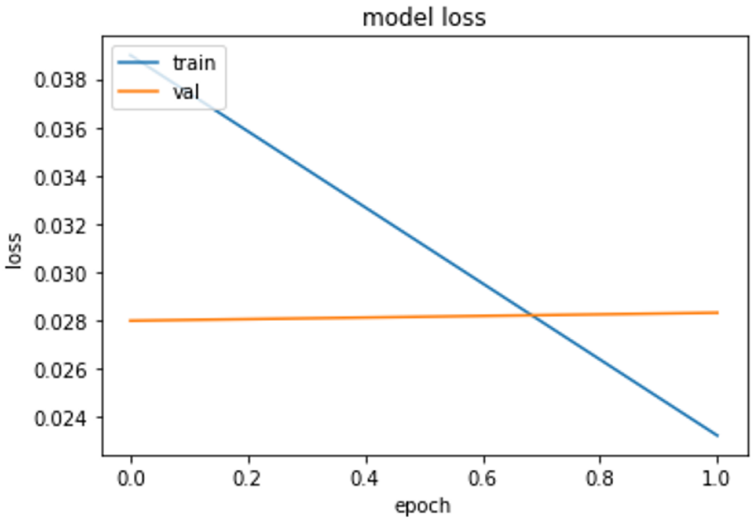

Figure A.2: While it is not possible to infer the specific trend of a curve with just two epochs, it is evident that the training loss is expected to decrease with further epochs and has dropped below the validation loss indicating that further training of the model would likely result in overfitting

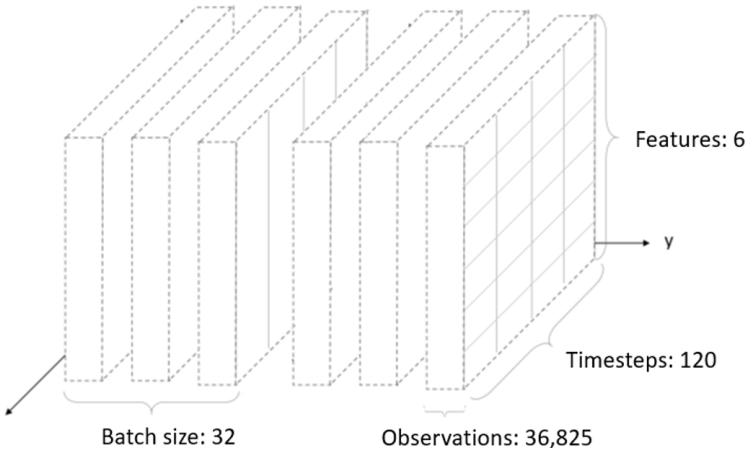

Figure A.3: The shape of the data for Model A, including the batch size and number of observations, number of features and context length, before it is seen by the Bi-LSTM

.

$$N_{parameters}^{hidden} = 2 \times 4 \times ((N_{features} + N_{nodes}) \times N_{nodes} + N_{features}) \tag{A.1}$$

$$N_{parameters}^{output} = N_{features} + N_{features} \times N_{parameters} \tag{A.2}$$