APPLICATION OF MACHINE LEARNING TECHNIQUES TO PREDICT THE FUTURE WEATHER AND CLIMATE USING THE GLOBAL HISTORICAL CLIMATOLOGY NETWORK

Candidate Number: REDACTED

University College London

Table of Contents

I. INTRODUCTION

The urgency to accurately predict weather and climate patterns has become more significant with the ever-intensifying issues of global warming and climate change. These challenges require new tools and techniques to improve the accuracy of predictions. Machine learning has emerged as a powerful tool to address this need by enhancing the accuracy and timeliness of weather and climate predictions [1]. This project focuses on the application of deep learning, a subset of machine learning, in order to predict climate variables such as temperature, precipitation, and snowfall for real-world weather stations. The primary objective of this report is to evaluate the effectiveness of deep learning techniques in predicting the daily weather patterns and their monthly averages, defined as the climate, and to explore the potential of deep learning in providing accurate long-term forecasts.

The GHCN

The Global Historical Climatology Network (GHCN) is a dataset that consists of the daily measurements of observed weather patterns all over the globe. It contains readings for a multitude of weather metrics from 991 worldwide stations, where the most up-to-date recordings were taken in 2020. In this project, only the five 'key' weather metrics are investigated, these are: Maximum Temperature (Code: *"TMAX"*), Minimum Temperature (Code: *"TMIN"*), Precipitation (Code: *"PRCP"*), Snow Depth (Code: *"SNWD"*) and Snowfall (Code: *"SNOW"*). These metrics are not only the most recorded for each station, but they provide some of the best indications of the impact of climate change.

Neural Networks

A neural network is a type of machine learning algorithm that takes inspiration from the structure and operation of the human brain. It consists of linked layers of simple processing units called "neurons" [2], between which exist numerical weights and biases that influence the output of the connected neuron according to the following formula:

$$y = f(b + \sum_{i=1}^{n} w_i x_i) \tag{1}$$

where y is the output of a neuron, f is the applied activation function (tanh, relu, sigmoid etc.), b is the bias, x_i is the input of a previous (connected) node and w_i is its corresponding weight. Using this, a network may be able to take some fixed-size input, update its weights and biases between neurons using stochastic gradient descent in a process known as 'training' [3], and produce some desired fixed-size output, analogous to a learning process. In this project, the desired output is ultimately a prediction of the future weather/climate, and so a type of network that can adapt to the patterns shown in the data is required. A Recurrent Neural Network (RNN) introduces loops within the network that allow for the output of a previous time step to be inputted to the next one and so forth, essentially allowing the network to maintain a "memory" of previous inputs via the recurrence relation:

$$h_t = f_W(h_{t-1}, x_t) \tag{2}$$

where h_t is the hidden state at time t, h_{t-1} is the hidden state from the previous time step, f_W is a non-linear function and x_t is the input at time t. This type of network would be able to adjust to the sequential weather data successfully, however, the 'exploding' or 'vanishing gradient' problem may be encountered during training. This occurs during backpropagation, a process where the loss function gradients are calculated in order to update the network's weights and biases via gradient descent, causing the gradients to become either very large or very small. This results in either a poor network performance/failure to converge or difficulty in successfully training the network [4]. For this reason, a specialised type of RNN will be utilised known as Long Short-Term Memory (LSTM), a form of gated cell that can make binary choices regarding the flow of information, allowing for information to be read, written, or deleted. An exemplary network of three LSTM cells is shown below.

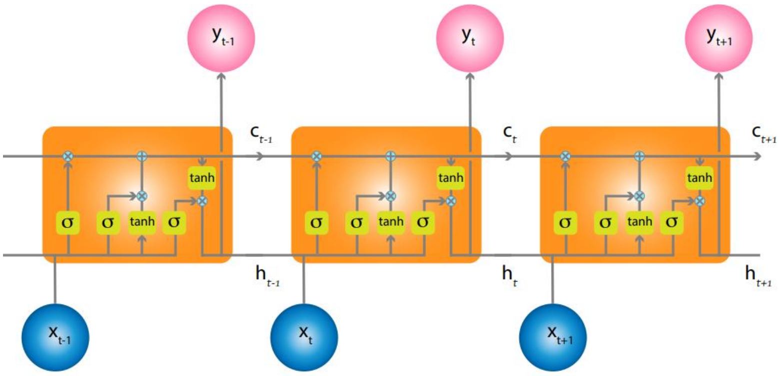


Figure 1: Three connected LSTM cells. Illustration from UCL PHAS0056: LSTMs [5]

Here, c_t is the memory and h_t is the output value. The leftmost section of each cell dictates the 'forget' function, where c_t is either multiplied by 1 (keep) or 0 (delete). The centre section of the cell represents the input function, where the stored value c_t is selectively updated with new values from the input or updated by the previous output h_{t-1}. The rightmost section of a cell corresponds to the output function, where the stored value is selectively outputted based on the current cell's input or the previous cell's output. Due to the nature of each processes' dependence on x_t and h_{t-1}, both the input and previous outputs can influence future deletion, writing or storing. Also, backpropagation is much simpler through time, rectifying the exploding and vanishing gradient problems found in standard RNNs [5].

The way in which an LSTM predicts the future of a time-ordered array involves the 'inspection' of a window of data in order to predict a target value some offset steps ahead. The window is then shifted along by 1, overlapping the previous one, in order to predict the consecutive target value and so on. This is outlined as an example on a number line in Fig. 2 below.

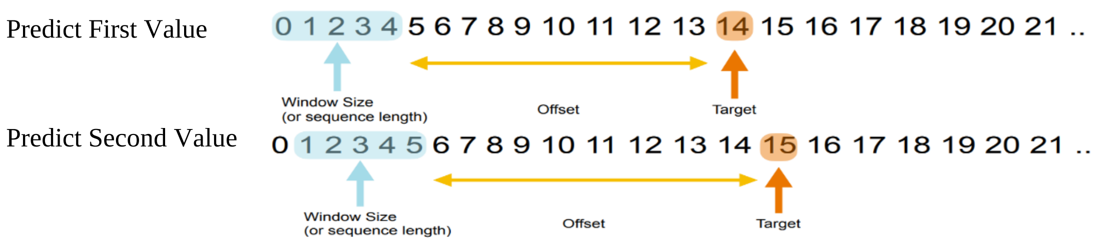


Figure 2: Illustration of future prediction method used by an LSTM on a number line, with a window size of 5 and offset of 10. From UCL PHAS0056: Week 6 Exercise Hints [6]

In this example, data within a window size of 5 is processed to predict a value with an offset of 10 from the end of the window. Note that although the first returned value from the network will be the 14th number, the prediction made is still equal to the offset of 10 values. For the data used in this project, an LSTM would examine the patterns shown in previous weather and climate measurements within a chosen window size to predict a recording of a chosen metric some given offset days or months into the future. This is then repeated by shifting the window size to predict the next measurement and so on, analogous to the example presented above.

II. Cleaning the Data

It has been discovered that many of the stations from the GHCN have incomplete data sets, where there exist sizeable 'gaps' in the daily recordings of the weather metrics, or measurements have not been taken for certain weather metrics. An example of this is in the *TMAX* and *TMIN* recordings at the KUT-AL-HAI station in Iraq, as shown below.

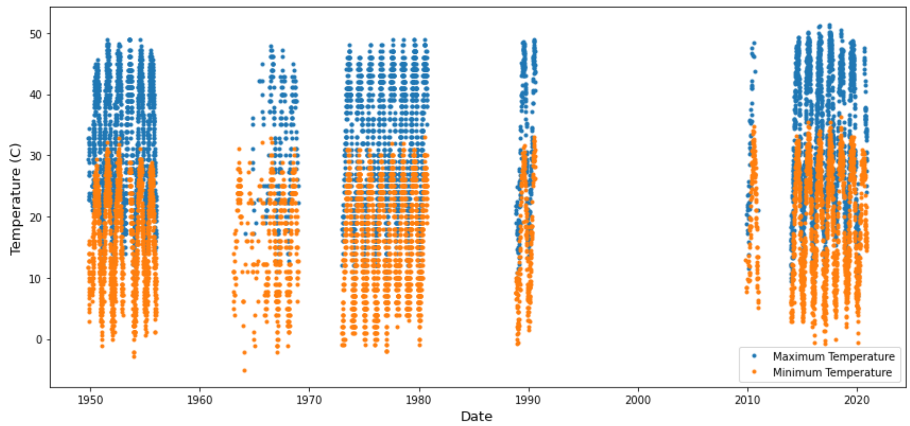


Figure 3: 'TMAX' and 'TMIN' recordings at KUT-AL-HAI Station in Iraq at 32.17, 46.05, 15.0

To train a neural network on such data would result in a recognition of improper patterns which would undoubtedly affect the prediction for a metric with a significant number of data holes. This is due to the fact that a given window of data may contain non-consecutive recordings, displaying an incorrect trend that the network will learn and make unfitting predictions on. Thus, it is evident that data points must be inserted into the dates where measurements are absent.

Filling the Gaps

Linear interpolation has been chosen for addressing the gaps within the datasets, allowing for the holes to be filled based on the existing data that surrounds them. This has been done for the station's recordings shown in Fig. 3, and the resultant completed dataset is shown below.

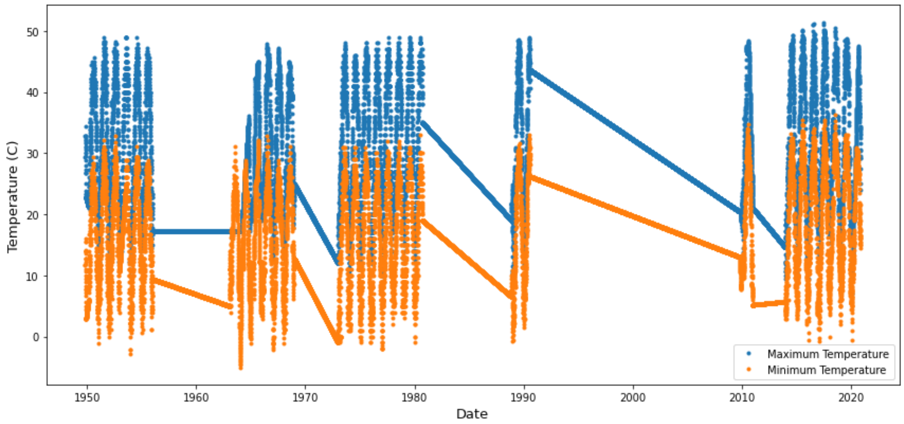


Figure 4: 'TMAX' and 'TMIN' data from Fig. 1, where all gaps in the data have been linearly interpolated.

A closer look at Fig. 4 reveals that linear interpolation is not appropriate for inserting data into the larger lapses in measurements as it does not preserve the observed patterns in temperature fluctuations demonstrated by the original readings. This method however does seem to prove effective in the areas where there exist smaller holes, such as those between 1960 and 1970, and so finding a station where the gaps are minimal, and the number of recordings is plentiful should theoretically enable the network to perform optimally.

Finding the 'Best' Station

The best station is defined as the one that meets the following criteria:

1. Data must be present for all five metrics.
2. For each metric, there must be at least 5000 recordings.
3. Has the lowest number of missing measurements (gaps) for each metric.
4. The measurements should be up to date, ending in 2020.

An algorithm was designed in order to assess each of the 991 stations against this criterion, returning the 15 station indexes that most closely aligned with these requirements. The output was as follows:

[954 931 918 959 938 894 946 902 937 897 905 958 926 379 929]

with station 954 as the best fit. While this station does in fact have the 'most optimal data', it was found that the measurements for *SNWD* and *SNOW* were all 0. In order to provide the neural network with more formidable complex patterns, the second listed station, 931, was chosen instead as it enables the assessment of the network's ability to forecast snowfall. The station data was then 'cleaned' using linear interpolation, which was valid since there exists a maximum of 5 holes for any given metric's measurements. The full datasets for each metric are shown in Fig. 5 below.

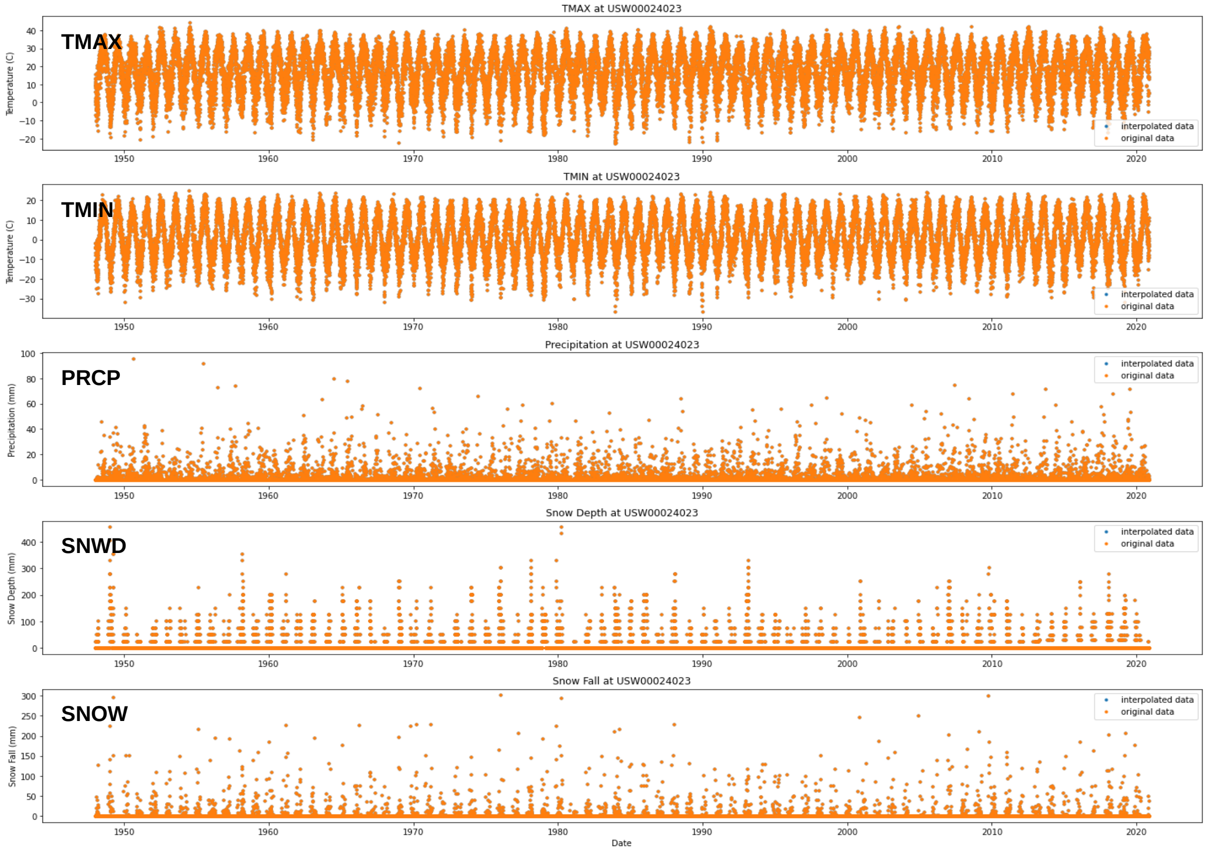


Figure 5: Recorded and linearly interpolated measurements for weather metrics at NORTH PLATTE RGNL AP, United States at 41.1214, -100.6694, 846.7.

As shown in Fig. 5 above, the datasets are mostly completed for each metric via the original recordings, where the linearly interpolated points have negligible effect on the weather patterns yet have successfully filled the small number of gaps present.

III. Predicting the Climate in 1 Year

To make a climate prediction one year into the future, it is necessary to convert the measurements for each metric into their monthly averages, as this meets the definition of 'climate'. Clearly, for data with voids, months with missing recordings may be subjected to skewed averages, and so it was chosen to linearly interpolate the data before reframing to a monthly basis. Now a neural network may be trained and permitted to successfully predict the future climate for the chosen weather station.

Network Architecture

The chosen model architecture for weather and climate forecasting is shown in the diagram below.

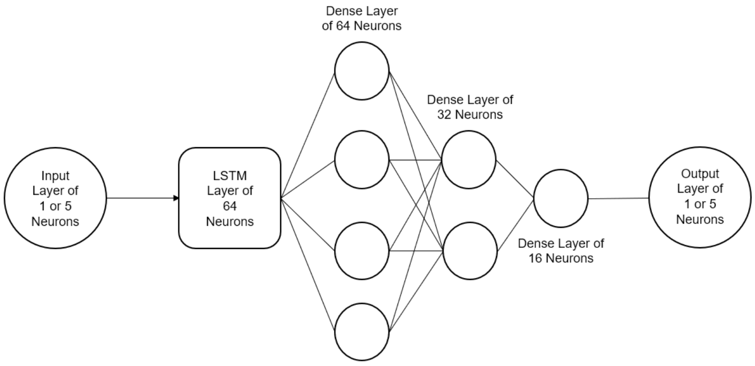


Figure 6: Chosen network structure consisting of 1 input layer, 1 LSTM layer, 3 dense layers and 1 output layer.

The input and output layers contain either 1 or 5 neurons, depending on the number of metrics being trained on and predicted at once. Dense (fully connected) layers were used in conjunction with the LSTM layer to map the output to the desired size and further assist the model in learning the complex relationships present in the datasets. The number of dense layers however has been limited as overfitting may arise if too many fully connected neurons are involved in the training process.

Training

To prepare for model training, the 875 monthly averages are split into training, validation and testing sets. These were of sizes 539, 168 and 168 respectively, where each were sliced from the interpolated data set sequentially. The training set is used during fitting to allow the network to 'learn' the patterns and dependencies of the historical data, while the validation set tests the network's performance against an 'unseen' continuation of the dataset. Observing the loss function for the validation set provides key insight into how well the network is performing, as an increasing loss typically indicates overfitting. Overfitting describes the phenomenon whereby the network is adhering to the training data too well, memorising it rather than learning its underlying patterns that allow it to make accurate predictions and evaluations on new data. It is ideal to minimise or, if possible, eliminate overfitting as it will severely hinder the model's ability to predict the future climate.

The batch size and epoch count are important factors during the fitting process. The batch size refers to the number of samples processed by the network before its parameters are updated and the epoch count refers to the number of times the network will train on the full training set [7]. During training, a batch size of 64 was selected as this number is low enough so that the model is updated frequently and can converge quickly, but high enough such that the process remains computationally efficient and remains stable. An epoch count of 50 was chosen as it was found that this was the optimal number of training iterations that minimised the training cost function while also keeping overfitting to a minimum, as the validation loss was found to highly fluctuate past this point. This is evidenced by the loss functions when training on a matrix of all metrics' data at 50 epochs versus training on 100 epochs, which are shown in Fig. 7 and Fig. 8 respectively below.

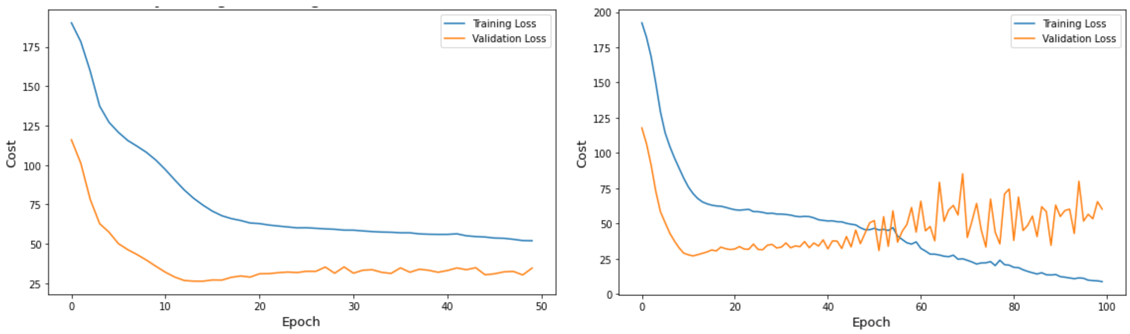


Figure 7 and Figure 8: Training and Validation loss functions when training on stacked climate data for 50 epochs (left) versus training for 100 epochs (right).

Predicting

It was found that the total deviation from the original climate data was minimised when training the network on each metrics' data individually rather than on stacked data, where normalisation benefitted the performance of the system. This was deducted via an evaluation of how these factors affected *TMAX* and *PRCP* as the assumption was made that *TMIN* and *SNWD* would follow the patterns of *TMAX* with *SNOW* resembling the spontaneous nature of *PRCP*. A window size of 36 was used to predict an offset of 12 months into the future, essentially using the previous 3 years' worth of climate data to predict one year into the future for each data point. The resulting predictions are shown below.

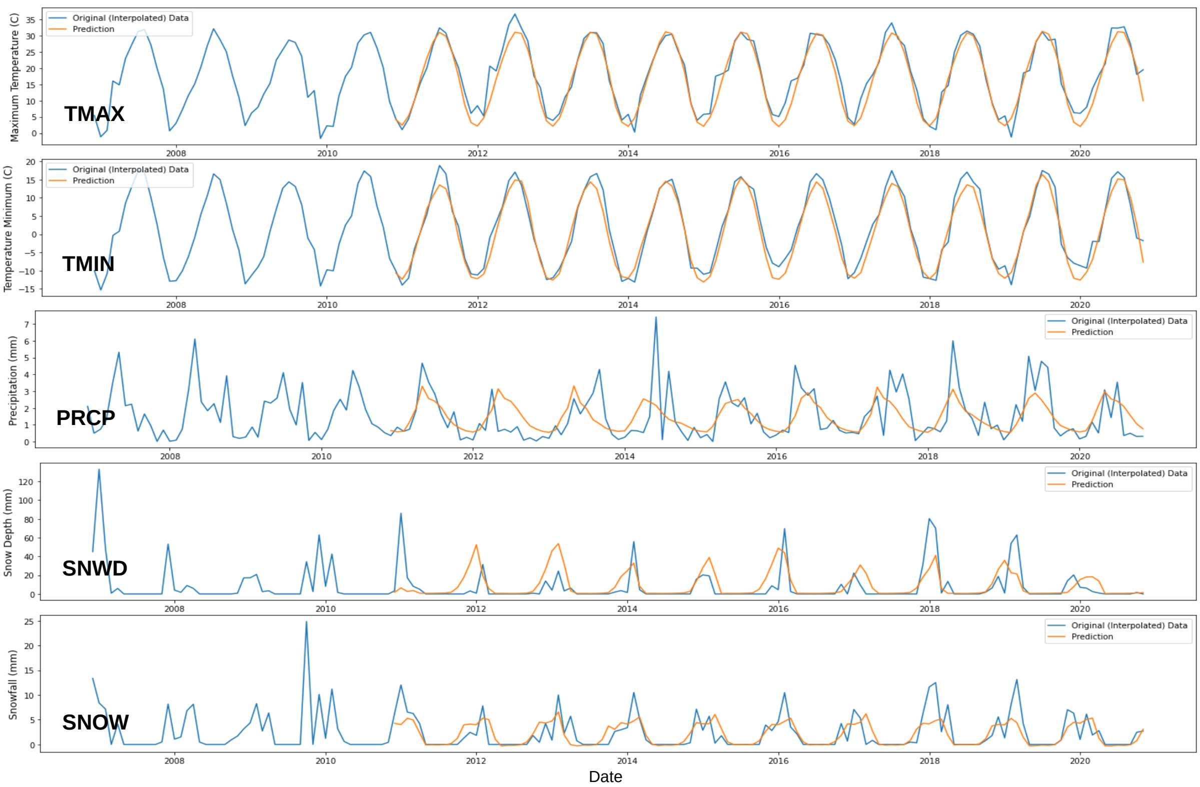


Figure 9: Climate predictions for TMAX, TMIN, PRCP, SNWD and SNOW arranged in vertical order, at station (41.1214, -100.6694, 846.7) with a forecast of one year ahead.

The predictions shown in Fig. 9 show that the model is particularly good at predicting the monthly temperature averages, perhaps due to their repeated sinusoidal nature of which the network may easily recognise and be able to predict readily. These predictions however do seem to underestimate the temperature maximums and predict lower minimums for years further into the future when compared to the original data, although not to a great extent. This may demonstrate the impact of global warming in recent years, as temperature maximums are being exceeded year on year at a faster rate than ever before, with minimums increasing at an equivalent pace, which the model is unable to account for. As for the precipitation, snow depth and snowfall, the network competently predicts when such weather events are likely but is unable to accurately facilitate the exact amount of rain or snow which may fall on a monthly averaged basis. This may be due to the almost spontaneous nature of such weather events, where no obvious patterns excluding the periodicity are present, unlike the consistent shape displayed by the average temperature year on year. Nevertheless, the predictions for the three lattermost metrics still provide good indication of the future climate and remain within proximity to the peaks of the original data such that the prediction may be called successful.

IV. Predicting the Weather

By employing the same network architecture as before, it is possible to forecast the weather one day in advance utilising the same approach applied for predicting the climate.

Training

To prepare for model training, the 26,617 data points for each metric were again split into training, validation and testing sets of sizes 16,617, 5,000 and 5,000 respectively. Ensuring that the model has enough data to train on and understand the underlying patterns of is highly crucial for maximising the prediction performance, and so it was chosen to again use the majority of the dataset for training.

As substantially more values are being used in the training process than for the climate data, the batch size was increased to 96. This value simultaneously enables the network to update its weights and biases frequently enough such that convergence is sufficiently fast while also retaining computational efficiency, minimising the hindrances to runtime caused by low batch sizes. To further ensure that runtime was not severely impeded, and also to reduce overfitting, the epoch count has also been adjusted, taking a new value of 15 iterations.

Via an identical method of evaluation as done prior, it was found that training on stacked data yielded predictions with the lowest total deviation from the original data and so a matrix of each metric's data was constructed for the model to train and predict on. Each metrics' data was then extracted individually from the predicted matrix produced by the network. The loss function using this method and the parameters mentioned is shown below.

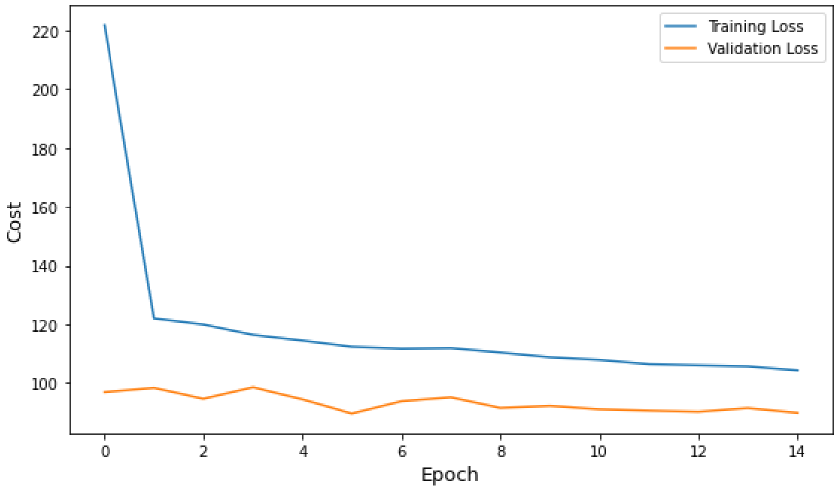


Figure 10: Training and validation loss functions obtained when training on stacked metric data.

The loss function does not display the same curved nature of the climate predictions, but this may be attributed to the large convergence shown within the first epoch.

Predicting

In order to predict one day into the future, an offset of 1 is required with a chosen window size of 4, where the network essentially uses data from the previous 4 days to predict the weather on the next. The predictions obtained were then compared to an assumption where "the weather tomorrow was exactly the same as the weather today", essentially a shift of the original data by one position. The results obtained for *TMAX* and *TMIN* are shown below.

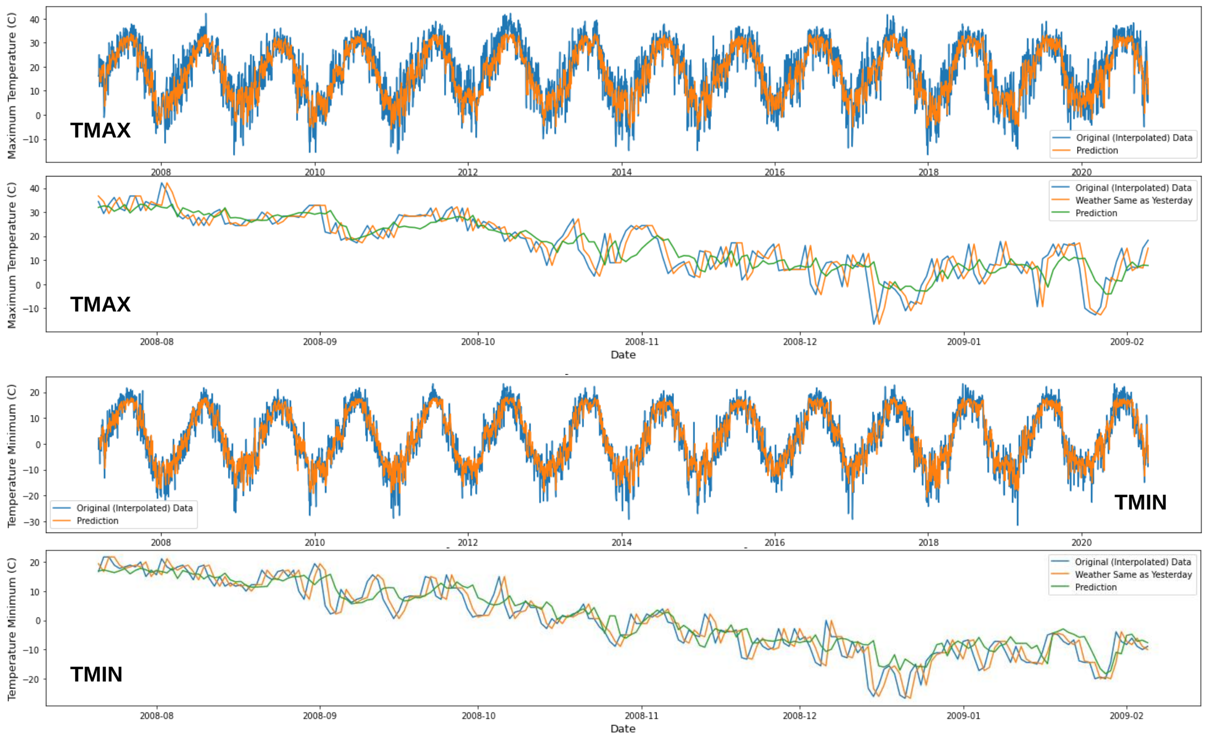


Figure 11: TMAX and TMIN daily weather predictions shown in their entirety, then with comparison to "same as yesterday" prediction" in order.

Fig 11 shows that the predictions follow the general trend of the original data, showing that the network has understood the underlying patterns to some extent, however, fails to meet the exact patterns displayed, missing many of the peaks and troughs. The model is also unable to predict unusual temperature peaks during non-summer months, which is expected as these are 'outliers' in the original data trend, giving a noisy appearance. As seen in the second and final subplots of Fig. 11, the network is unable to surpass the compared assumption but still may give valuable insight into future weather trends such as general increases or decreases in temperatures over a fixed time period.

The data for *PRCP*, *SNWD* and *SNOW* demonstrate more complex patterns than *TMAX* and *TMIN* and so it is expected that the network will have a reduced performance when predicting these, as an underlying pattern is not as prevalent. The model predictions are shown below, again with a comparison to the assumption used prior.

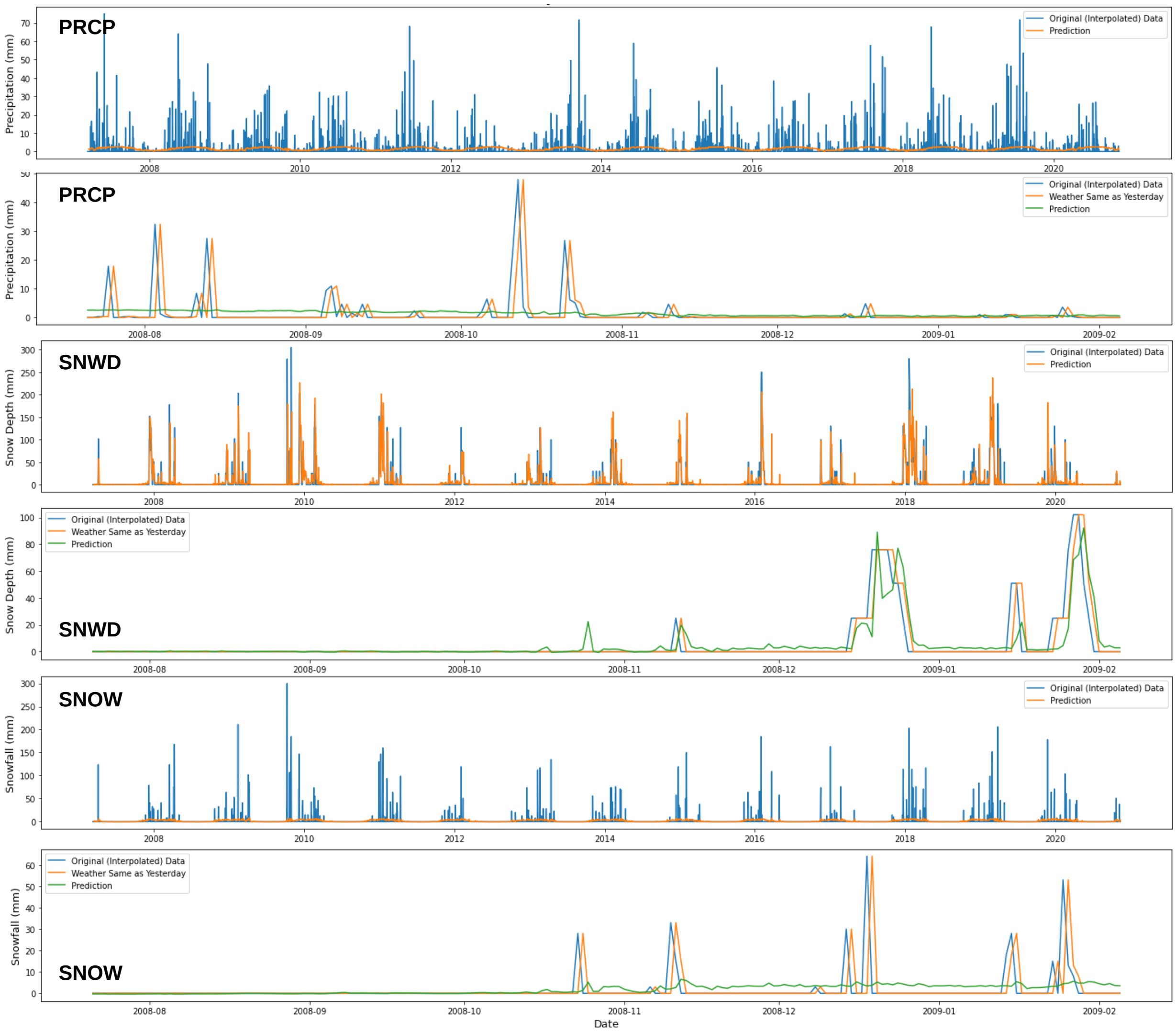


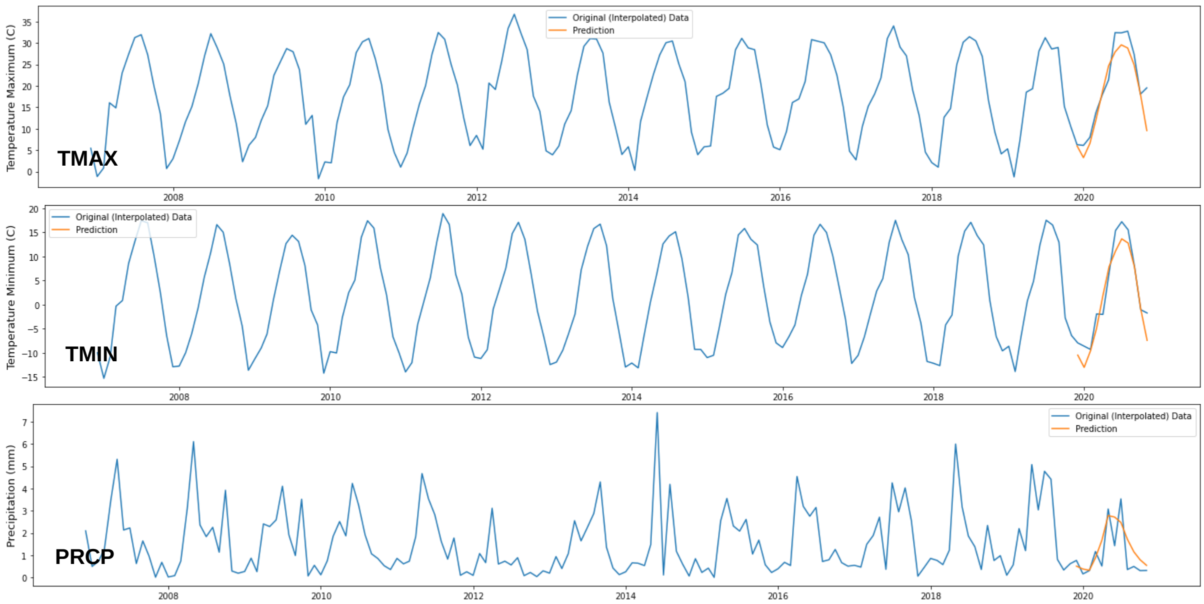
Figure 12: PRCP, SNWD and SNOW daily weather predictions shown in their entirety, then with comparison to "same as yesterday" prediction" in order.

Fig. 12 demonstrates that the network is significantly poorer at predicting these metrics than the temperatures as shown in Fig. 11. The model is unable to predict the amount of daily rainfall and snowfall to any sort of accuracy, providing only small perturbations in an otherwise zero line in positions which may roughly correlate with days where such weather events occur. This method of prediction is therefore insufficient for determining the rainfall or snowfall for future days. For the snow depth however, the network performs relatively well, giving accurate predictions that align with the peaks of the original data, although tends to sporadically predict non-zero depth on days where no depth is measured. This contrasts with the other two metrics' predictions in Fig. 12, perhaps arising due to the dependence of snow depth on previous readings, whereas the patterns in rainfall and snowfall are relatively random as these are spontaneous events, with no definite impact on the weather on the next day.

V. Predicting the Climate in 10 Years

Based on the findings of the previous two tasks, it has been discovered that forecasting the climate is relatively easier for a neural network to predict with accuracy compared to predicting the weather. This is most likely due to the explicit periodicities of the climate data, whereas there exist large fluctuations between consecutive readings in the daily weather, displaying no set pattern on a small scale. For this reason, it can be tested how well a network is able to predict the climate even further into the future and assess whether the accuracy of such predictions would still remain to an acceptable level.

By implementing an identical methodology used to produce a forecast of the climate in one year, and modifying the offset accordingly, it is possible to produce predictions for the climate in 10 years' time. For this length of prediction, an offset of 120 months is required where the window size has remained as 36 months due to the lengths of the training and testing sets, as a larger window size would in turn force the network to be trained on less data points. The climate predictions produced by the model for each metric in the next 10 years' time are displayed below.



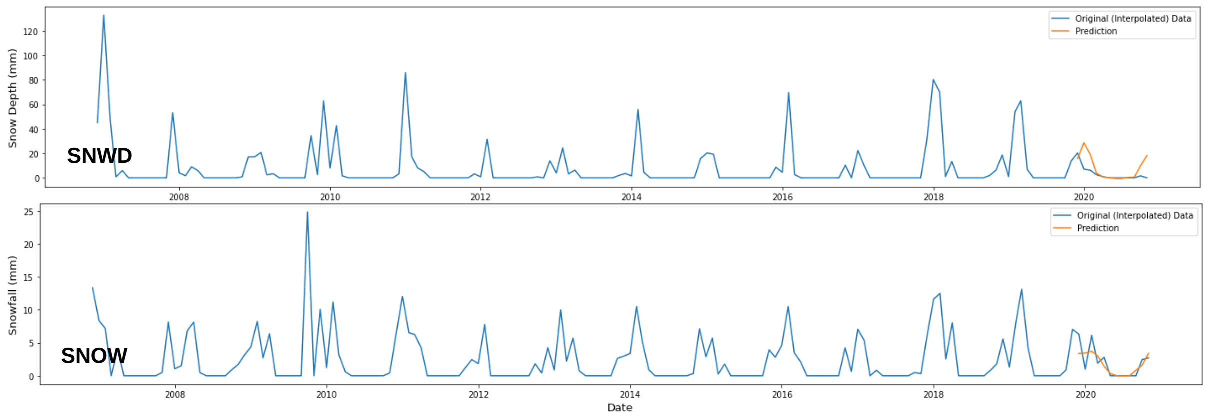


Figure 13: TMAX, TMIN, PRCP, SNWD and SNOW 10-year climate forecasts, shown in vertical order.

For *TMAX* and *TMIN*, the network accurately predicts the temperature rises in 2020, before peaking significantly below the exact measurement as shown in Fig. 13. This was also found to be a common shortcoming of the model's predictions when forecasting the climate one year in advance, however the fact that the disparity has become larger in this case possibly indicates the substantial impact of global warming within the last two decades. The network actively predicts lower temperature maximum and minimum peaks in 2020 than those recorded as it fails to account for the exponential increase in global warming over this period. The rises in the average temperature in this time have increased to a degree unfound in the network's training set (ending in 1993), explaining the inability of the network to forecast such outcomes much farther into the future. Also, the measurements for 2020 indicate one of the warmest winters, which the model does not anticipate correctly, instead projecting climate patterns resemblant of those from the late 20th and early 21st centuries.

Fig. 13 demonstrates similar predictions to those found in Fig. 9 for *PRCP*, *SNWD* and *SNOW*, where no significant discrepancies are found between forecasting these metrics one year in advance compared to ten years in advance. The network is still able to roughly gauge how much rain or snow may fall within a span of a few months but remains unable to accurately predict the amplitudes of the exact recorded points which follow a strictly monthly basis.

Overall, the network's success in predicting the climate 10 years in advance may be comparable to its success in forecasting for one year. However, this argument may undergo significant changes as the amount of training data has not been sacrificed to allow for more data to be predicted, which may present new flaws in the predictions made for the climate in 10 years' time.

VI. CONCLUSION

In conclusion, a machine learning technique was successfully developed and implemented to predict the future weather and climate to varying degrees of accuracy using station data from the GHCN. The general process for each prediction primarily entailed the ‘cleaning’ of a weather metrics’ recordings, linearly interpolating the gaps between measurements in order to form a complete dataset. This interpolation method was only valid for stations with small holes in their dataset, and so the ‘best’ station data was found and used in this project. Then, a neural network consisting of an LSTM layer and multiple dense layers was trained on this ‘cleaned’ data in order to form future predictions from it.

Overall, the network was able to make successful forecasts of the climate to varying extents depending on the metric, whereas predicting the daily weather proved to be more difficult due to the existence of large fluctuations between readings across all metrics. It was found in all cases that forecasts of the temperature and snow depth on both a daily and monthly basis were more aligned with the real measurements when compared to predictions on rainfall and snowfall. This performance disparity was attributed to the nature of such weather patterns, where *TMAX*, *TMIN* and *SNWD* exhibited clearer underlying patterns as there exists a dependency on previous recordings, unlike *PRCP* and *SNOW* where these weather events are more spontaneous and do not necessarily depend on prior observations.

The forecasts made by the model for the monthly averaged *TMAX* and *TMIN* may have implications relating to global warming and climate change. When predicting the climate, it was found that the network frequently underestimated the temperature peaks achieved on a yearly basis, where the distinction between the apexes of the prediction and the original data curve became more prominent with increasing time. This may be attributed to the exponential effect of global warming within the last two decades as the network had only been trained on measurements taken before 1994, where the pattern of rapidly increasing temperature maximums may not have been as prevalent in this time period and thus not available for the network to detect.

Some improvements that may be made to the methodology of this investigation may include an alternative approach to filling the gaps within a stations’ dataset as linear interpolation only remains valid for data with a minute number of missing measurements. A method which assesses the underlying patterns of the surrounding data (e.g. spline) would allow for increased validity, making the investigation of many more stations’ data possible and may even lead to an enhancement of prediction performance. Other developments of this project may include the investigation of other network architectures, possibly involving multiple LSTM layers, or a further study into the performance of a network when predicting the climate further into the future than was completed. Stations in the GHCN also have recordings for multiple other weather metrics such as sunlight and cloud cover, which could also have the potential to be forecasted by a neural network.

VII. BIBLIOGRAPHY

[1] G. R. Kanagachidambaresan, K. Saravanan, and R. M. Chandrasekaran, "Weather prediction using machine learning techniques: A review," International Journal of Computer Science and Mobile Computing, vol. 3, no. 6, pp. 394-401, Jun. 2014.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533-536, 1986.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proc. 25th Int. Conf. Neural Inf. Process. Syst., Lake Tahoe, Nevada, USA, 2012, pp. 1097-1105.

[4] M. Nielsen, "Neural Networks and Deep Learning," Determination Press, 2015, pp. 269-279. [Online]. Available: https://neuralnetworksanddeeplearning.com/chap5.html. [Accessed: Mar. 29, 2023].

[5] R. Nichol, "Week6_LSTM". UCL Moodle Resource, 2021. [Online] Available: https://ucl-eu-west-2-moodle-sitedata.s3.eu-west-2.amazonaws.com/d8/a3/d8a34259a1dfc051558d85049ce71fe615c3032b?response-content-disposition=inline%3B%20filename%3D%22Week6_LSTM.pdf%22&response-content-type=application%2Fpdf&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA47YHZF637GKGWUJC%2F20230325%2Feu-west-2%2Fs3%2Faws4_request&X-Amz-Date=20230325T185247Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21553&X-Amz-Signature=ba81b6c040612829a15b5aff27b55eb4ef0932ebef3344d3316a016cffb1df87. [Accessed: March. 18, 2023].

[6] R. Nichol, "Week6_Hint". UCL Moodle Resource, 2023. [Online] Available: https://moodle.ucl.ac.uk/pluginfile.php/5987863/mod_resource/content/2/Week6_Hint.pdf. [Accessed: March. 18, 2023].

[7] S. Chollet, "Deep Learning with Python," Manning Publications Co., 2018, pp. 130-136. [Online]. Available: https://www.manning.com/books/deep-learning-with-python. [Accessed: Mar. 29, 2023].