A Method Using LSTM Networks to Impute Missing Temperatures in Temperature Datasets and to Predict Future Temperatures

Chenyang Qiu*

Department of Physics and Astronomy, University College London, London, United Kingdom

* Corresponding Author Email: chenyang.qiu.19@ucl.ac.uk

Abstract. With the growth of population and the development of industry, global warming has become a serious problem. It is necessary to focus on temperature change now. This paper designed and tested methods to impute the missing temperature in global temperature datasets and predict the future temperature. Imputation was divided into two steps, the first step is filling small gaps with equally spaced points for efficiency, and the second step is filling larger gaps with Long Short-Term Memory (LSTM) neural networks. This combination of imputation methods gave valid points with both accuracy and efficiency and successfully made some datasets continuous. Another LSTM network was used to predict future temperatures. The $\chi 2$ test for the goodness of prediction with real data in Australia showed that this prediction method using LSTM network is highly accurate.

Keywords: LSTM imputation; LSTM prediction; RNN.

1. Introduction

Global warming is a concerning issue in the present world. The rise of temperature is now threating health, economics, and population of human beings, and harming the environment and animals [1,2]. It is essential to focus on temperature changing and this experiment is to design a computing method to predict future temperature.

Neural networks represent a computational paradigm in which the solution to a problem is learned from a set of examples. As a non-linear mathematical function, a feedforward neural network converts a collection of input variables into a set of output variables. A group of parameters known as weights, whose values may be established based on a collection of instances of the necessary mapping, dictate the precise shape of the transformation. It may need extensive computational resources to determine these parameter values, a process that is sometimes referred to as learning or training. However, the network can handle fresh input quickly after the weights have been set [3].

The Long Short-Term Memory neutral networks (LSTM), integrated in the Recurrent Neural Network (RNN) [4, 5], display dynamic temporal behavior during a time sequence since LSTM was initially developed in 1997 [6]. It is a well-liked model that has demonstrated excellent promise in several prediction models that rely on historical data [7]. Therefore, this experiment would apply LSTM neural network on temperature forecast.

The memory blocks that make up the LSTM architecture are a collection of recurrently linked sub-networks. The memory block's purpose is to preserve its state over time and control data flow using non-linear gating units. A cell, an input gate, an output gate, and a forget gate make up an LSTM unit. The three gates control the flow of information related to the cell, and the cell remembers values across arbitrary time periods [8].

2. Problem Definition and Algorithm

This whole experiment was done in Python 3.0 with build-in and open-source software libraries. Tensor Flow is a machine learning software package that is free and open-source with an emphasis on deep neural network training and inference [9, 10]. It is possible to design a neural network and execute RNN with LSTM using Tensor flow using the Keras package, which offers a Python interface for artificial neural networks.

With the database provided by Global Historical Climatology Network which stored the climate data including the daily highest and lowest temperature for the recent decades in the world, a set of LSTM networks could be designed and trained to predict future temperature. However, the climate data stored in some stations were not continuous due to data missing or recording problems, while a LSTM network required a continuous data set to train and predict.

The simplest way to fix it was deleting data before the recent missing part, but as the consequence, some data set would be too short to provide an accurate prediction. Therefore, the missing parts should be imputed before training LSTM networks.

2.1. Turning the Daily Temperature into Monthly Temperature

It would take an insanely long time on training and predicting temperatures days by days. In order to save time, monthly average values for both highest and lowest temperature recorded should be derived for LSTM networks.

Each element in the Date array stored in the database indicated the specific year and month and was aggressively sorted, so a loop comparing the year value and month value with the previous element in the array was created, all the elements with identical year and month values would be combined as one element in a new array, then a Month array would be derived as a 4-D array, with the year, month, number of valid days (that record the temperatures), and an index to indicate the position of the first valid day in this month and its corresponding temperatures in the original dataset.

After that, the corresponding highest and lowest temperatures for each day should be combined as an average value for each month in order to make month and temperature as a one-to-one function. With the number of valid days and the index to indicate the position of corresponding temperatures, two arrays of all the highest and lowest temperatures in each month was evaluated respectively, and the average values of these arrays would be the monthly highest temperature and lowest temperature for each month. Fig. 1 gives an example of this algorithm, the average y value of the points in the red box will be the temperature value for month”2017-12”.

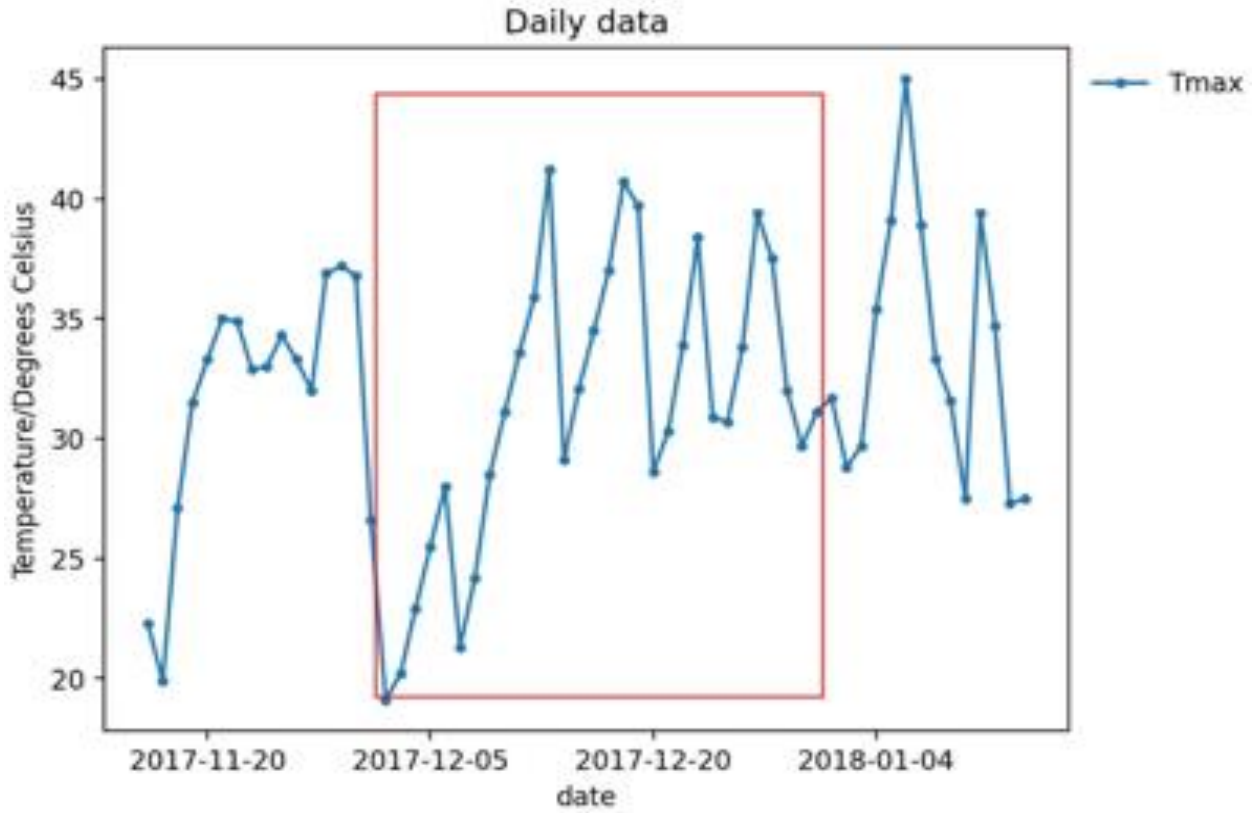


Fig 1. Highest temperatures from 2017-11-15 to 2018-01-15 recorded in MILDURA AIRPORT, Australia

2.2. Deriving the List of Missing Months

Then all the missing months needed to be pointed out in each climate station. The first element and the last element in the Month array gave the earliest and latest years and months stored in the station. For the first and last years, loops were created to check if all the months after the earliest or before the latest month were stored in the station respectively, if not, the missing months with their corresponding year would be recorded in a Missing Months array. At the same time, another function was set to check if the years were continuous between the earliest and latest, if not, all the 12 months will be added to the Missing Months array. The middle years were also checked month by month to looking for the missing ones, and all of them would be stored in MissingMonths array aggressively sorted.

If the first continuous part was shorter than the first missing part, the first continuous one would be removed, and the first month in the next continuous one would be considered as the first month of the dataset. If there were too much missing data, then this station should be removed from the database.

2.3. Imputation for Missing Data

Imputation for missing data was similar to the prediction for future, so LSTM networks could also be the method for filling it. In fact, LSTM networks were proved as a very accurate method for missing data imputation [11]. Nevertheless, if the missing period was too short, the accuracy of the imputation would not influence our final forecast, but LSTM networks took a lot of time to run for each gap. Therefore, short-term missing data (less than 6 months) would be filled with a simple method, while LSTM networks were used to fill long-term missing data (equal or greater than 6 months). Temperature in 6 months usually only have one or zero turning point, so a simple method could be a time-saving way for these tiny gaps.

2.3.1. Imputation of short-term missing data

This imputation method was quite simple. The missing part will be filled with equal equally spaced values between the nearest neighbour. For example, if data in October and November was missing, while the highest temperatures recorded in September and December were 15 degrees Celsius and 6 degrees Celsius, then temperature in October would be 12 degrees Celsius and 9 degrees Celsius respectively.

2.3.2. Imputation of long-term missing data

The nearest continuous data (after all short-term missing data were filled) before the long-term missing data would be used to train a LSTM network with 4 layers in total (1 LSTM layer with 80 notes, 3 Dense layers with 110 notes each except the last one for output). A dataset in a station may have multiple large gaps, all of them could be filled with LSTM networks.

2.4. Prediction of the Future

With the continuous datasets in different stations, an 8-layer-LSTM network (1 LSTM layer with 120 notes, 7 dense layers with 110 notes each except the last one for output) was trained, and gave prediction of both highest and lowest temperature for each month in the next year.

2.5. Accuracy Test

After the dataset was filled, data for the last 12 months would not be used for training. These 12-month data could be compared with the predicted ones, and a $\chi 2$ test was used to test the goodness of fit, which also indicated the accuracy.

$$\chi^2 = \sum_i \left(\frac{(PredictedTemperature_i - RealTemperature_i)^2}{RealTemperature_i}\right) \quad (1)$$

P-value could be derived for each prediction by build-in function scipy.stats.chi2.sf ($\chi 2$, degree of freedom) in Python 3.0, where degree of freedom equals the number of predicted points in one prediction minus 1, which would be 11 in this case. Larger p-value means better fitness.

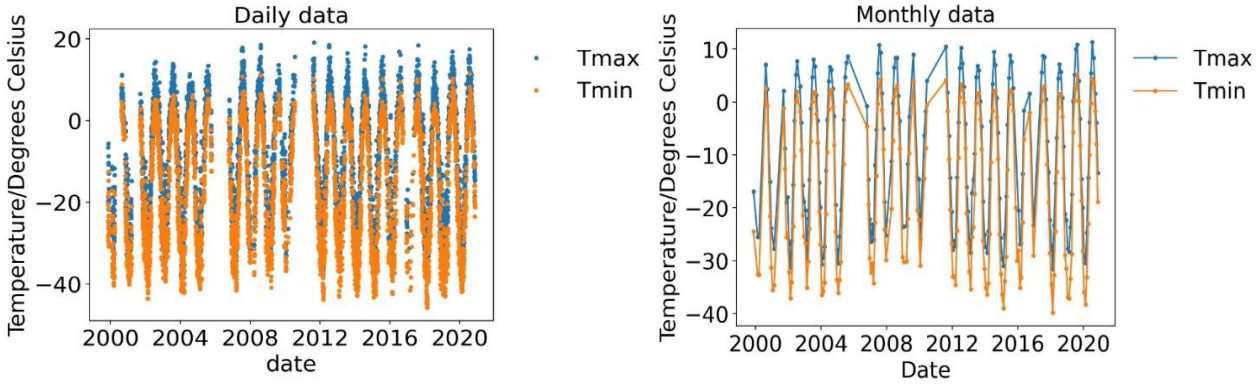
An average p-value for all the prediction in a region would be the probability that this LSTM network methods gave accurate forecast for the future in this region.

3. Experimental Evaluation

3.1. Methodology

3.1.1. Turning the daily temperature into monthly temperature

In order to test the accuracy of prediction after imputing the missing data, a station in ARCTIC BAY CS, Canada was chosen for its acceptable discontinuity as shown in Fig. 2(a), and the monthly figure were derived in Fig. 2(b).



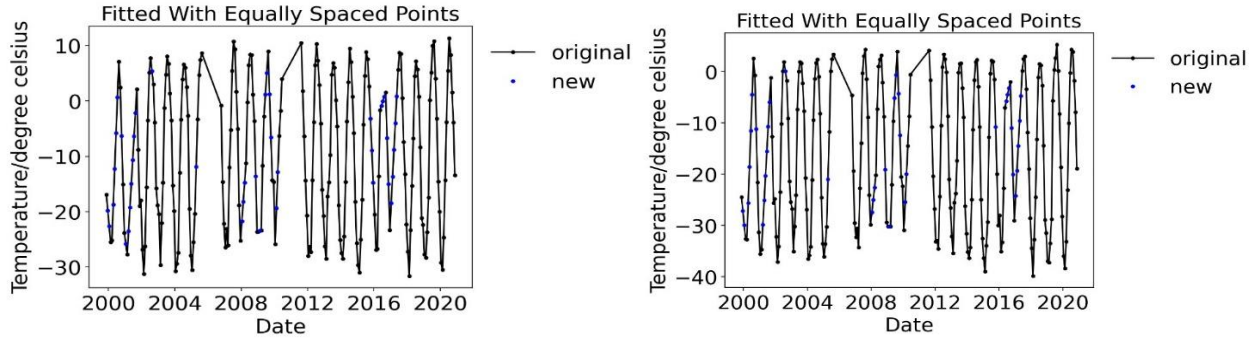
(a) (b)

Fig 2. Highest and Lowest temperature (Tmax and Tmin) stored in ARCTIC BAY CS, Canada. (a)Original Daily Temperature, (b) Derived Monthly Temperature.

Both figures in Fig. 2 had same pattern which means the functions to turn the daily data into monthly one worked.

3.1.2. Imputation of missing data

Obviously, there were two giant gaps in Fig. 2(b), which needed to be filled by LSTM networks, and plenty of small gaps should be imputed by equally spaced points.

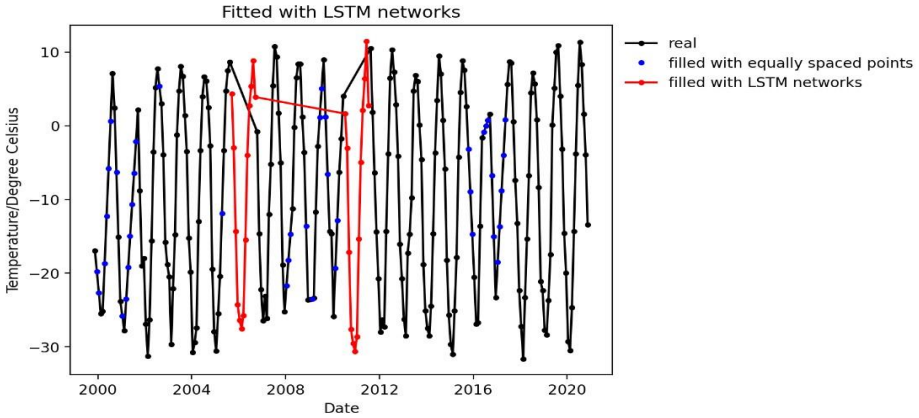


(a) (b)

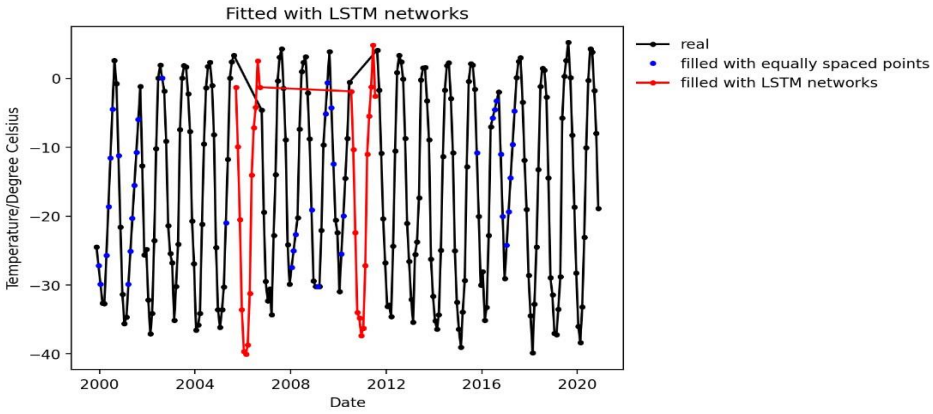
Fig 3. Filling after equally spaced points imputation. (a) Highest Temperature, (b) Lowest Temperature.

Fig. 3 shows the data filling after equally spaced point's imputation. Clearly, most points in both Fig. 3(a) and Fig. 3(b) looked reasonable, except the ones in 2016. This was the expectable result, so

the experiment would carry on to find out the final influence of this error. The two gaps in each plot were then filled by LSTM networks trained with existing data, and the outcome is shown in Fig. 4.



(a)



(b)

Fig 4. Filling with LSTM networks. (a) Highest Temperature, (b) Lowest Temperature.

The imputed points in both Fig. 4(a) and Fig. 4(b) gave acceptable pattern and fitted in the rest of the points, which meant temperature data in this station was continuous and ready for training and prediction, although some points in 2016 were obviously not properly fitted.

3.1.3. Training and prediction

Two LSTM networks were established for highest and lowest temperature stored in this station. Through these networks training model, 2000 to 2019 as training data, and 2020 data as test data. As shown in Fig. 5, both networks gave forecast of highest and lowest temperature respectively.

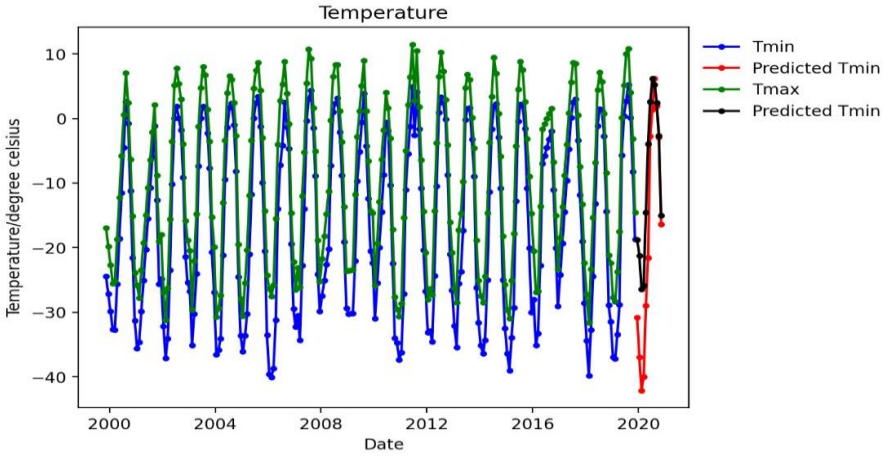


Fig 5. Original dataset with prediction.

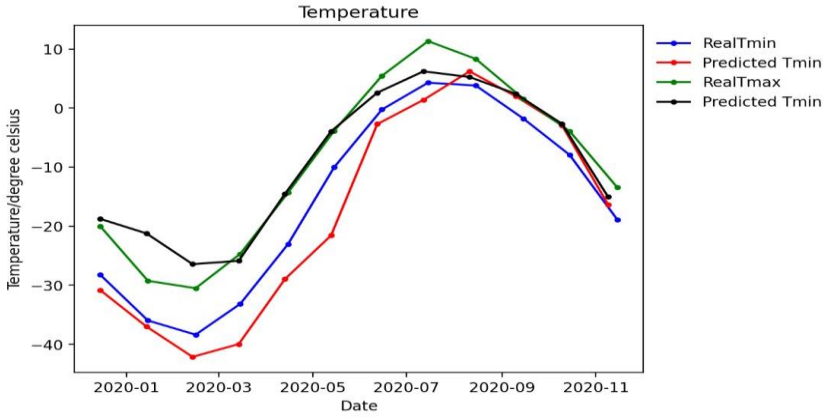


Fig 6. Real and predicted temperature in 2020.

While comparing the predicted ones with the real one in Fig. 6, both lines gave similar but not exactly same patterns. The outcome seems reasonable considering the dataset itself was discontinuously at the beginning, and a further accuracy would be test to verify this assumption.

3.1.4. Accuracy test

The average p-value for these two predictions (highest and lowest temperature) was 0.9993 with a standard deviation of 0.0005, which means these LSTM networks was good enough to predict the future temperature, even if a small number of imputed data in 2016 for the missing gaps (Fig. 3) did not fit enough. This could drive a conclusion that some tiny errors during the equally spaced imputation would not influence the final prediction a lot.

3.2. Results

3.2.1. Region choosing

After testing the accuracy of these imputation and prediction method, a region should be chosen to give a general prediction to test if these methods work for other stations.
Australia was chosen as the region for forecasting, because most stations in Australia provided datasets with no or few gaps and stations were all cross Australia.

3.2.2. Stations choosing

By viewing the original temperature data plot in all 56 stations in Australia, 41 of them were chosen as shown in Fig. 7 due to their general continuities. Examples of continuous, acceptable discontinuous and useless datasets were shown in Fig. 8. Datasets in Fig. 8(a) and Fig. 8(b) would be used while dataset in Fig. 8(c) would be removed.

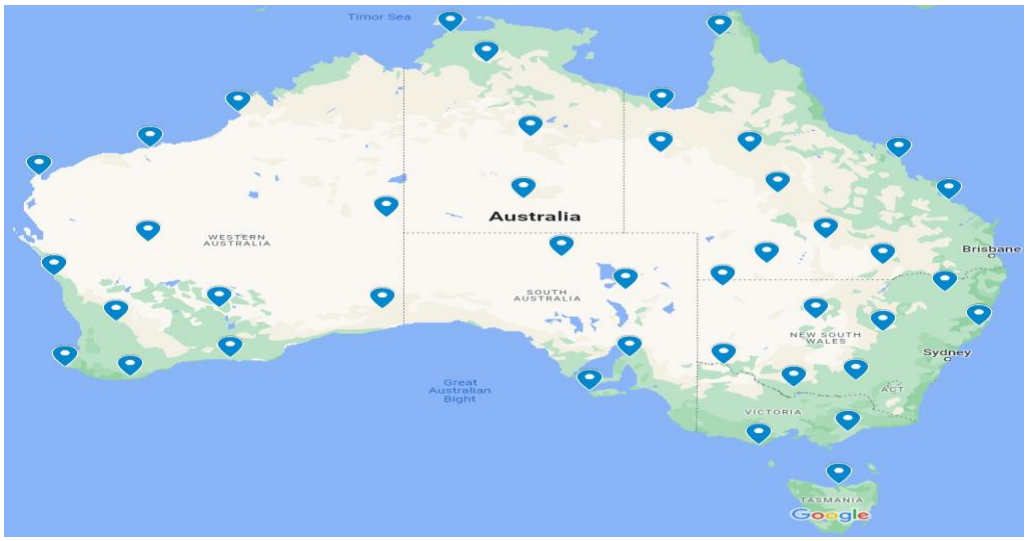
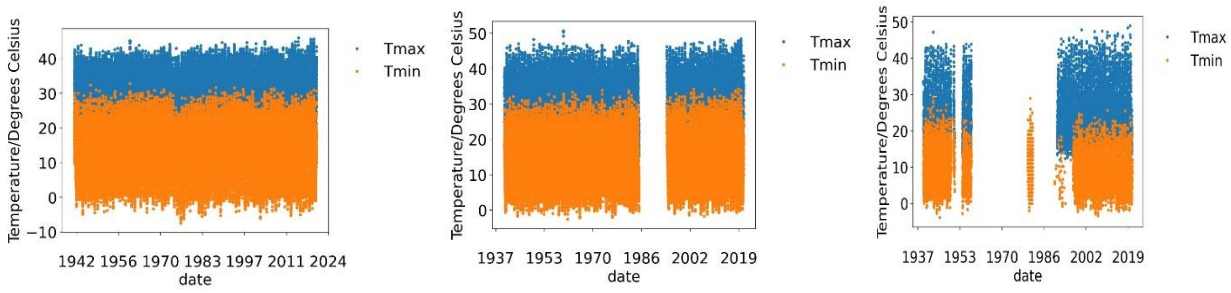


Fig 7. Stations with valid datasets in Australia.

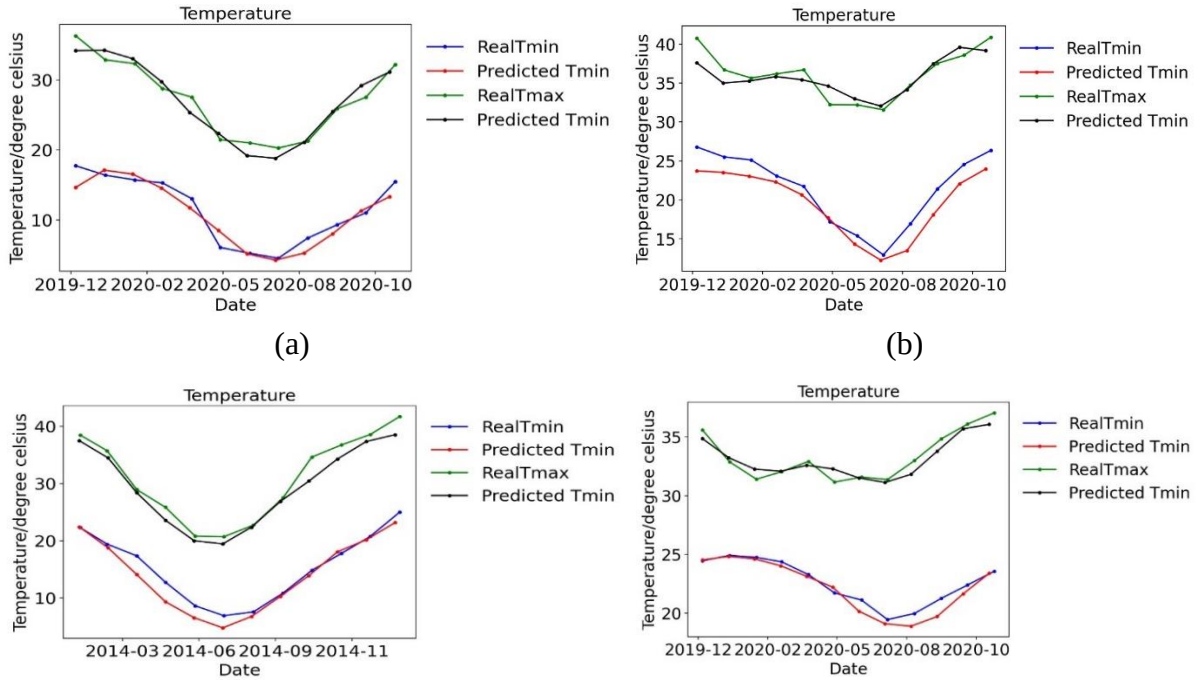


(a) (b) (c)

Fig 8. Examples of stations. (a) Continuous dataset stored in ALICE SPRINGS AIRPORT, Australia, (b) Acceptable discontinuous dataset stored in OODNADATTA AIRPORT, Australia, (c) Useless dataset stored in CEDUNA POST OFFICE, Australia.

3.2.3. Results and discussion

Four of the predictions in these 41 stations were selected randomly as examples shown in Fig. 9.



(a) (b)

(c) (d)

Fig 9. Examples of predictions. (a) FORREST, Australia, (b) TINDAL RAAF, Australia, (c) MARREE COMPARISON, Australia, (d) WEIPA AERO, Australia.

In Fig. 9, obviously, all the predictions were accurate enough to show the pattern and values compare with the real temperature. The p-values for these 82 predictions from both highest and lowest temperature stored in the 41 stations were derived. The average p-value was 0.9994 with the standard deviation of 0.0011, which supported the idea that the prediction method using LSTM networks worked well in forecasting temperature.

Specifically for the discontinuous datasets filled by LSTM network imputations, the average p-value was 0.9984 with the standard deviation of 0.0016, which was close to the one for all predictions, which means the imputation methods successfully filled the datasets for prediction, and did not lead to any significant errors of the final forecast. Fig. 10 showed a discontinuous dataset and how it was imputed and predicted step by step. Fig. 10(e) and (f) gave acceptable imputations for the missing part. Consequently, by comparing the real and predicted temperatures shown in Fig. 10(h) also showed very close patterns. With the p-value evaluated above, these predictions after imputation using LSTM networks worked as well as the ones did not need an imputation.

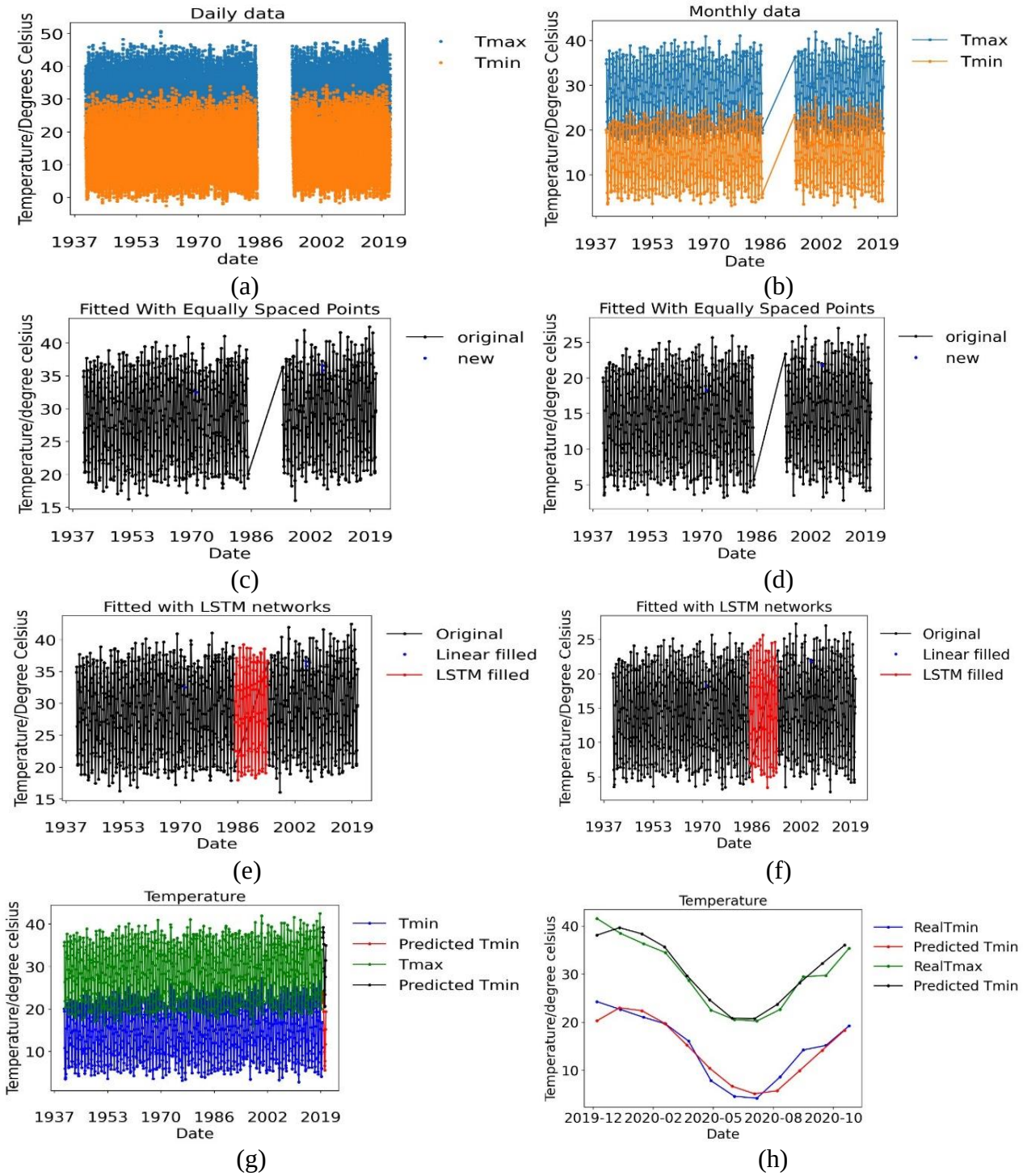


Fig 10. Dataset stored in OODNADATTA AIRPORT, Australia. (a) Original Daily Data, (b) Original Monthly Data, (c) Imputation of highest temperature with equally spaced points, (d) Imputation of lowest temperature with equally spaced points, (e) Imputation of highest temperature with LSTM networks, (f) Imputation of lowest temperature with LSTM networks, (g) Original dataset with predictions (h) Real and predicted temperatures.

4. Future Work

Current methods only dealt with temperature data, while there are other datasets like weathers in the Global Historical Climatology Network, with well-trained LSTM networks, they could also be predicted with temperatures.

If there are more powerful graphic cards and enough timing, the precision for prediction could be changed to daily, which would give a better plotting for each station.

5. Conclusion

In this experiment, a two-step imputation method was designed to fill missing data with equally spaced points for small number of missing data and LSTM networks for large ones, giving consideration of both efficiency and accuracy. Apart from that, a LSTM network method was also established to predict future temperature, with the filled data. $\chi 2$ goodness of fit test indicated this prediction methods gave very close predictions. These imputation and prediction methods worked properly while dealing with missing temperature in climate stations, and should work as well when dealing with other time-based datasets.

Particularly, Predictions of future temperature in Australia for a year with the average p-value equals to 0.9994 in $\chi 2$ fitness test, which means these methods worked perfectly. With better hardware and longer time, it is possible to use LSTM networks to predict global temperatures in a few decades, which might help when making strategy to reduce global warming.

References

[1] Diffenbaugh N S, Burke M. Global warming has increased global economic inequality. Proceedings of the National Academy of Sciences, 2019, 116(20): 9808-9813.

[2] Boni R. Heat stress, a serious threat to reproductive function in animals and humans. Molecular Reproduction and Development, 2019, 86(10): 1307-1323.

[3] Bishop C M. Neural networks and their applications. Review of scientific instruments, 1994, 65(6): 1803-1832.

[4] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with lstm. Neural Computation, 2000, 12(10): 2451-2471.

[5] Lin T, Horne B G, Tino P, et al. Learning long-term dependencies in narx recurrent neural networks. IEEE Transactions on Neural Networks, 1996, 7(6): 1329-1338.

[6] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780.

[7] Tsai Y T, Zeng Y R, Chang Y S. Air pollution forecasting using rnn with lstm//2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). IEEE, 2018: 1074-1079.

[8] Van Houdt G, Mosquera C, Napoles G. A review on the long short-term memory model. Artificial Intelligence Review, 2020, 53(8): 5929-5955.

[9] Hope T, Resheff Y S, Lieder I. Learning tensorflow: A guide to building deep learning systems." O'Reilly Media, Inc.", 2017.

[10] Shukla N, Fricklas K. Machine learning with tensorflow. Manning Greenwich, 2018.

[11] Li L, Zhou H, Liu H, et al. A hybrid method coupling empirical mode decomposition and a long short-term memory network to predict missing measured signal data of shm systems. Structural Health Monitoring, 2021, 20(4): 1778-1793.