Forecasting The Future Climate With Machine Learning Using GHCN Daily Data

Giorgi Markhulia

March 2025

Abstract

This report forecasts average monthly temperatures in Italy for 1 and 10-year horizons using the GHCN daily dataset. We utilise long-short-term memory (LSTM) and recurrent neural networks (RNN), comparing them against a naive baseline. Single-station data with a 1-year horizon underperforms compared to the baseline due to the sparsity of the data. Combining multiple stations reduces the mean squared error of the models substantially. For longer 10-year horizons, the LSTM model shows over-fitting issues, ineffectively capturing the temporal patterns in the climate data. A station-removal test further reveals the individual contributions of the stations, and how missing entries can hamper accuracy. Solutions like adding more robust climate features, better data parsing and improved hyperparameter tuning are proposed to increase the prediction accuracy.

Table of Contents

1 Introduction

Every day, people around the world rely on accurate weather forecasts to make important decisions - when to evacuate a city threatened by a tropical cyclone, how to prepare for extreme temperatures such as a heat wave or a blizzard, or how to optimise the use of renewable energy such as wind turbines and solar panels in a power grid [1]. Recognising this, scientists have long pursued robust techniques to measure and predict the future climate. In 1904, the Norwegian scientist Vilhelm Bjerknes proved that the behaviour of the atmosphere could be described mathematically. Only a few years later, his ideas were given a physical application by an English mathematician and meteorologist, Lewis Fry Richardson [2]. These breakthroughs created the basis for numerical weather forecasting as we know it. Today, analysing global climate records from data sets such as the Global Historical Climatology Network (GHCN) provides insights into the probabilistic nature of our climate and creates a means of identifying climate extremes, facilitates monitoring of global warming effects, and serves as a basis for predictive models [3]. This report focuses on the analysis of the GHCN daily database, specifically, the maximum and minimum temperature variables, and details a method for training long-short-term memory (LSTM) and recurrent neural networks (RNN) to forecast the average monthly temperature in Italy 1 and 10 years ahead.

The second section provides a detailed explanation of the GHCN data set, giving context to how the data is extracted and processed. Also, numerical weather forecasting (NWF) remains fundamentally uncertain [1], because it relies heavily on initial conditions and physical models that are limited in precision and/or computational complexity. Consequently, improvements in NWF are not only associated with increased predictive accuracy but also with reductions in energy consumption per unit of computing power [4]. In light of these challenges, recent advances in machine learning (ML) have emerged as an attractive alternative due to their ability to model complex relationships from large datasets with fewer forecast errors [1].

Accordingly, the second section also details the functionality of simple RNNs and LSTMs along with a description of a benchmark baseline test designed to evaluate the predictive power and accuracy of the models over 1- and 10-year horizons.

After establishing a comprehensive understanding of the dataset and the ML methods, the third section presents the technique of converting sequential data into a dataset of input-output pairs for prediction. Additionally, we present and discuss the outcomes of our models using one and multiple stations with a prediction horizon of 12 and 120 months. This section summarises all the key findings, explores the implications of the data and evaluates the success and effectiveness of implementing our ML approach. These insights pave the way for the concluding discussion where we propose improvements for fine-tuning the ML models as well as enhancements in the cleaning of the GHCN dataset.

2 The GHCN dataset and the machine learning methods

The section focuses on a detailed explanation of the structure of the GHCN dataset and the problems and limitations that the format of this dataset introduced during the project. In addition, this section will provide a description of general machine learning techniques and the methods that were used to develop a predictive model for this specific report.

2.1 The format and parsing of the GHCN dataset

The GHCN-Daily dataset is one of the most comprehensive global daily weather data sets we have [3]. Since the release of [3], the updated dataset now features data from 218 countries with records going back to 1763 in some cases [5]. An extensively dense network of recent data is present for regions like Australia, Europe, North America etc.. Hence, Italy was chosen as an area of interest in this report.

2.1.1 Format and Contents

For this study a subset Global Surface Network (GSN) of the dataset is used and the components are as follows:

1. **Metadata files:**
    - ghcnd-stations.txt - this text file contains the stations identifiers, coordinates including the elevation and station names.
    - ghcnd-countries.txt - maps the unique stations to the country codes.
2. **Daily data files:**
    - Individual .dly files: These files come from a URL path and contain daily weather measurements that are explained in more detail in Figure 1.

As is shown in Figure 1, the .dly files contain 5 different variables. These include: minimum and maximum temperature (TMIN and TMAX respectively), precipitation (PRCP), snowfall and snow-depth (SNOW and SNWD respectively. Every .dly file gathers data for at least one of these variables for each station, encompassing data from around 100 000 stations around the world [5]. For this report, only the TMAX and TMIN variables were used as these variables had the most non-zero entries and encompassed the aim of this report i.e. predicting the average monthly temperature the best. Moreover, it is important to note that the temperature values require re-scaling to tenths of degrees. This was done to ensure that the temperature values were in the logical range.

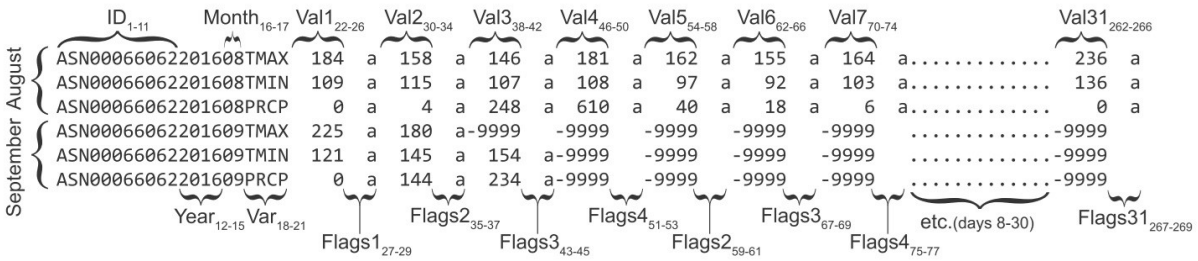


Figure 1: Depicts the structure of the GHCN-Daily .dly file. Subscripts indicate the column positions. Missing values are the entries with -9999. The definition of the columns is as follows: ID - weather station identifier; Var - variable name; Val1 - value of the variable on day 1; Flags1 - Three flags for day 1: quality flag, measurement flag and source flag; Flags2: the same three flags for day 2 and so on. Figure taken from [5].

2.1.2 Extraction and parsing

The extraction of the data from the .dly files is not a simple task. The necessary functions for the extraction of the variable values were provided before the start of the report.

The first step in the study was to choose a random station in Italy. In this case, it was a station with the following ID and coordinates: IT000016134 is MONTE CIMONE, Italy at 44.2, 10.7, 2165.0. After plotting the extracted values TMAX and TMIN, which are depicted in Figure 2, it became clear that the values between 2009 and 2013 were completely missing. This was something that had to be noted, as it would affect the training accuracy of the model.

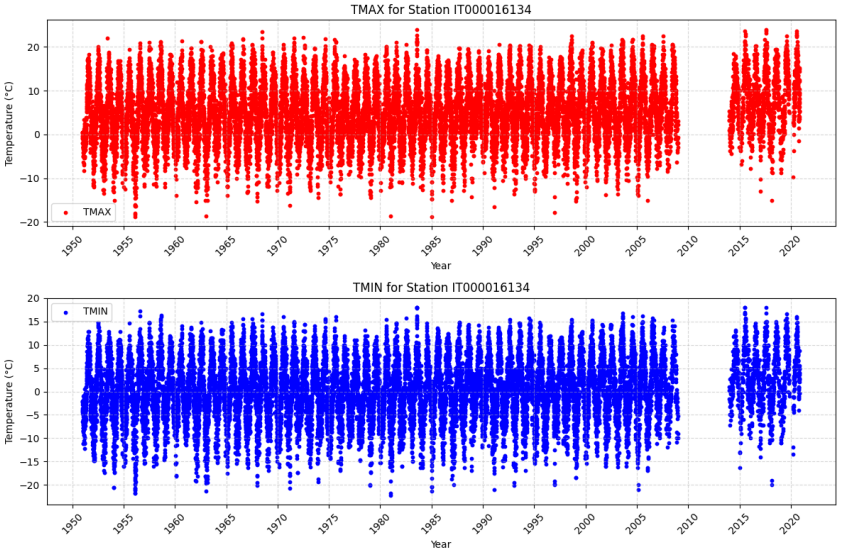


Figure 2: Depicts the distribution of the TMAX and TMIN values for station ID: IT000016134 for every year of the entry.

Furthermore, the oscillatory trend of the values in Figure 2 confirmed the accuracy of extraction. The TMAX and TMIN were expected to increase and decrease throughout the year as the seasons changed, and this trend is clearly visible in the plot.

Following this, to directly answer the question of forecasting an average monthly temperature, it was necessary to calculate the monthly average values for the station, which are depicted in Figure 3.

This time the data-frame (NaN) values were dropped completely, but the gap between 2009-2013 persists, as the plotting function took the entry each year and did not have any values for that specific range. Furthermore, the temperature value range was set between $[-50, 50]$°C filter for invalid entries.

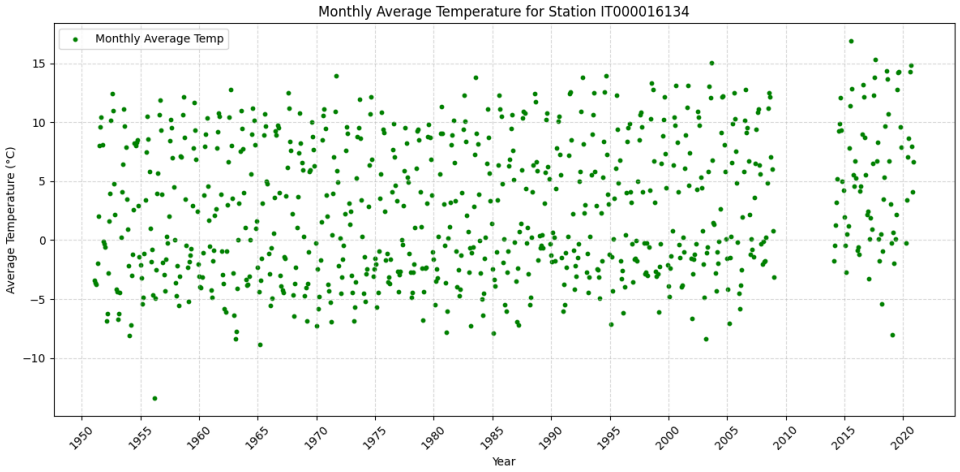


Figure 3: Depicts the distribution of the monthly average temperature values for station ID: IT000016134 for every year of the entry.

Finally, to create a larger dataset for training, all the stations present in the GSN dataset were identified. Afterwards, their co-ordinates were extracted and used for plotting on the map of Italy to get an idea of where these stations are located exactly. This is depicted in Figure 4. In addition, Table 1 depicts how the data is stored in each station .dly. Here the following observations can be made:

1. The stations are quite far away. Therefore, they could be capturing different climate trends which could affect the training of our model.

2. The elevation of the stations is also different, which would have an effect on the average temperature.

3. Most of the stations have the same starting year and the same ending year, which helps with the consistency of the data. It is worth noting the missing years and months for each year. For example, station ID: IT000016134 has the most missing years, so the expectation is that it would contribute negatively to the combined data-frame. In contrast, station ID: IT000016550 has no missing years and only 3 missing months, so it would be expected for it to contribute positively to the data-frame.

Station	Start Year	End Year	Missing Years	Incomplete Years
IT000016134	1951	2020	2009, 2010, 2011, 2012, 2013	2019: [12]; 2020: [1, 2, 12]
IT000016550	1951	2020	None	1977: [11, 12]; 2020: [12]
IT000160220	1951	2020	2009	2020: [12]
IT000162240	1954	2020	None	1954: [1, 2, 3, 4, 5, 6, 7, 8, 9]; 1958: [1, 9]; 2020: [12]
IT000162580	1951	2020	None	1951: [1, 2, 3, 4, 5, 6, 7, 8, 9]; 1961: [1, 2, 3]; 2020: [12]

Table 1: Station summary of available data.

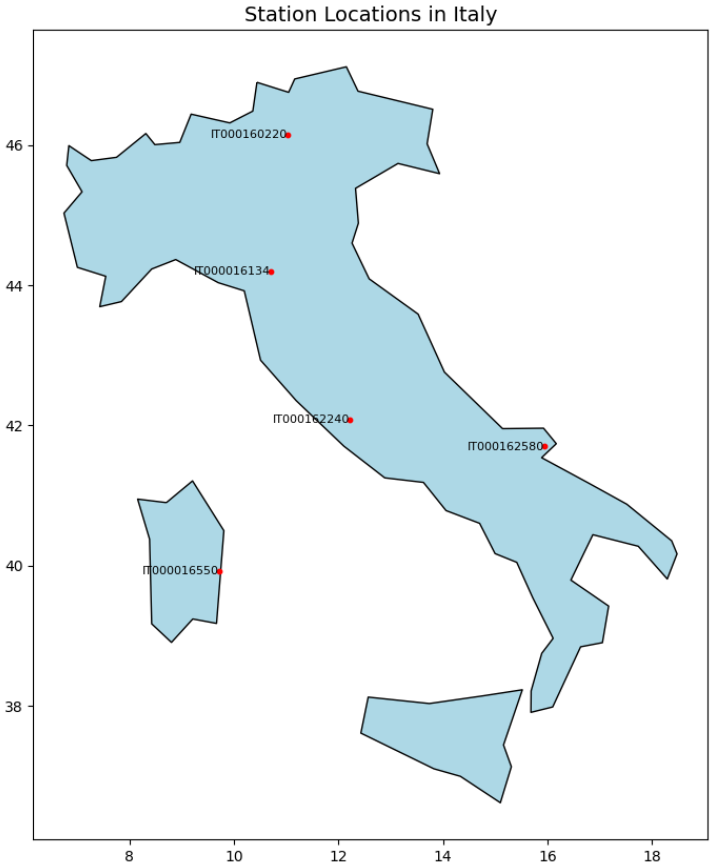


Figure 4: Depicts the exact locations of the stations used for the training dataset.

After assessing the locations and the respective entries in the datasets of the stations, they were ready to be combined into a single data-frame which would be used down the line for training. The first 10 entries of the data-frame are depicted in Table 2. At this stage, all the necessary data-frames were ready to start the machine learning step.

Index	Station ID	Date	Avg Temp
0	IT000016134	1951-01-31	-3.391935
1	IT000016134	1951-02-28	-3.619643
2	IT000016134	1951-03-31	-3.793548
3	IT000016134	1951-04-30	-2.003333
4	IT000016134	1951-05-31	2.041935
5	IT000016134	1951-06-30	7.980000
6	IT000016134	1951-07-31	9.562903
7	IT000016134	1951-08-31	10.409677
8	IT000016134	1951-09-30	8.083333
9	IT000016134	1951-10-31	-0.104839

Table 2: Sample data for the combined data-frame showing monthly average temperatures for the first 10 entries.

2.2 Model Descriptions: RNN and LSTM and baseline tests

This section will provide an overview of LSTMs and RNNs and explain why they were chosen for this report. This will include a comparative analysis of their strengths and weakness. Finally, an explanation of the baseline tests performed for the assessment of the accuracy of the models will be provided.

2.2.1 Overview of RNNs, LSTMs

RNNs are mainly used to detect patterns in a sequence of data. This neural network architecture type is different to Feed-forward Neural Networks in the way that the information gets passed through the network. For a feed-forward network, the information gets passed through without cycles, whereas the RNN has cycles and will transmit the information back to itself [6]. This difference makes RNNs particularly well-suited for modelling sequential data, where the past inputs have an influence on the future outputs [7], in this case, the climate data.

Similar to other networks, RNNs face the problem of vanishing or exploding gradients. For very small values, the gradient tends to decrease with each time step, whereas for very large values the gradient explodes. This essentially stops the contribution of the time steps that happened much earlier than the current one. In contrast, if the gradient explodes it will value each weight too much. The proposed solution to this problem was the creation of LSTMs to address the issue with the vanishing gradient [6].

Since LSTMs use a more constant error and allow RNNs to learn over more time steps, they effectively address the vanishing gradient problem [7].

The name LSTM refers to a type of RNN network that has both a long-term and short-term memory. The weights and biases of the connections in such a network change per training episode. Hence, LSTMs employ a more sophisticated architecture with gating mechanisms and memory cells [8].

The main motivation for choosing RNNs and LSTMs in this report stems from the sequential nature of the GHCN-Daily dataset. This data contains temporal dependencies that are necessary for an accurate forecast. Even though simple RNNs offer quite a straightforward approach to modelling such a sequence, their performance on long-term forecasting is limited compared to LSTMs. The differences are highlighted in Table 3

Model	Advantages	Disadvantages
RNN	• Simpler architecture and easier to implement. • Fewer parameters, which can result in faster training on short sequences.	• Prone to vanishing and exploding gradient problems. • Limited ability to capture long-term dependencies.
LSTM	• Effectively captures long-term dependencies with memory cells and gating mechanisms. • Mitigates the vanishing gradient problem, making it more robust for long sequences.	• More complex architecture with an increased number of parameters. • Higher computational cost and longer training times.

Table 3: Comparison of RNN and LSTM models in terms of their advantages and disadvantages.

In conclusion, both RNNs and LSTMs are well suited for modelling climate data. However, it is important to assess the capabilities of simpler models like RNNs compared to LSTMs to understand if the task at hand can be completed at a lower computational cost. In addition to this, the limitations that arise from using the GSN dataset, i.e. missing entries, limited number of stations, missing or invalid entries could prove a serious challenge for a more complicated network like an LSTM. With this foundation, it is now crucial to establish a baseline test for the models to be compared to.

2.2.2 Baseline test

The baseline test in this report serves as a benchmark for the predictive power of our models. It simply assumes that the temperature tomorrow is identical to today's.

This test creates a good reference point: it provides a minimum level of performance that our RNNs and LSTMs have to exceed in order to be considered effective. By comparing the mean squared error MSE of the models against the MSE of the naive baseline, we get a picture of how well the models are actually predicting, i.e. if the added complexity actually results in a better prediction.

If the models fail to outperform the naive baseline, this will be an indicator that they are not capturing useful patterns.

3 Predicting the average monthly temperature in Italy

This section provides a comprehensive explanation of how the dataset is divided into training and test sets and describes the implementation of a sliding window approach for prediction. Subsequent subsections present the performance of our models, illustrated with figures and Mean Squared Error (MSE) comparisons against a naive baseline.

3.1 Data splits

To finalise the preparation of the data for prediction, our report leverages a sliding window approach to convert the sequential monthly temperature data into structure input-output pairs. In this specific case, a window of 12 consecutive months is used as the input feature and, depending on the forecasting horizon, the subsequent 12 or 120 months serve as the target. This way, the temporal dependencies of the data are captured to the full extent.

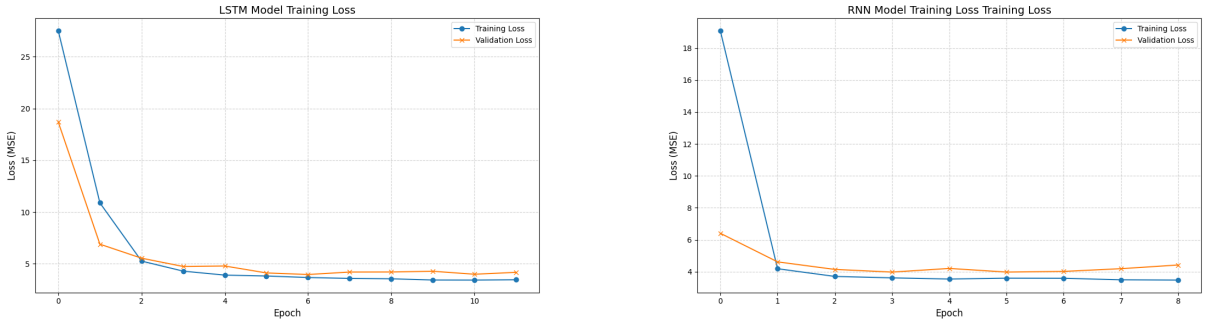
To ensure a robust evaluation, the data is split into training and test sets at a ratio of 80/20. The majority is used for the training and the remainder is reserved for evaluating their performance. Finally, the input data is reshaped into a three-dimensional format representing samples, time steps and features to meet the requirements of the RNN and LSTM architectures.

3.2 Using only one station with a forecasting horizon = 12

In this section, we focus on creating and evaluating our forecasting models for a one-year (12-month) horizon first. Using the sliding window approach, we use the 12 months of historical data as the input for the prediction of the subsequent 12 months. In this case, only one station ID: IT000016134 is MONTE CIMONE, Italy at 44.2, 10.7, 2165.0 is considered.

Our main models - LSTMs and RNNs-are trained on the sliding window dataset. A naive baseline, which uses the assumption that today's weather = tomorrow's weather, is implemented to serve as a performance benchmark.

3.2.1 Results



(a) LSTM Model training loss over the epochs. **(b)** RNN Model training loss over the epochs.

Figure 5: Comparison of two model performances for one station.

From Figure 5, the LSTM model's training and validation losses start relatively high but decline quickly. This reflects effective learning of temporal patterns. By epoch 5, the curves start to plateau, signalling that most of the learnable structure has been captured. The gap between the validation and training loss curves is negligible, and they converge to the same point, suggesting that the model is neither over-fitting nor under-fitting. After about 10 epochs, the loss metrics do not change significantly, prompting early stopping.

A similar analysis of the RNN graphs shows a rapid decline in both training and validation losses early on, but they plateau sooner than the LSTM. To put these results into perspective and assess their respective performance Figure 6 and Figure 7 depicts the forecast per month for both followed by a comparison of the MSE's to the naive baseline MSE tabulated in Table 4.

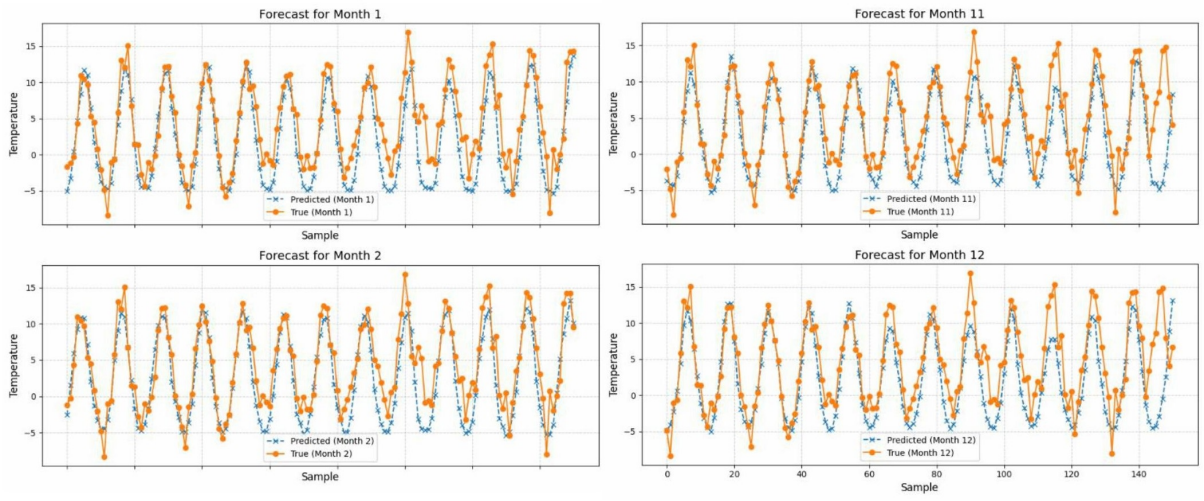


Figure 6: Depicts the first two and the last two months of prediction vs truth values for the LSTM forecast.

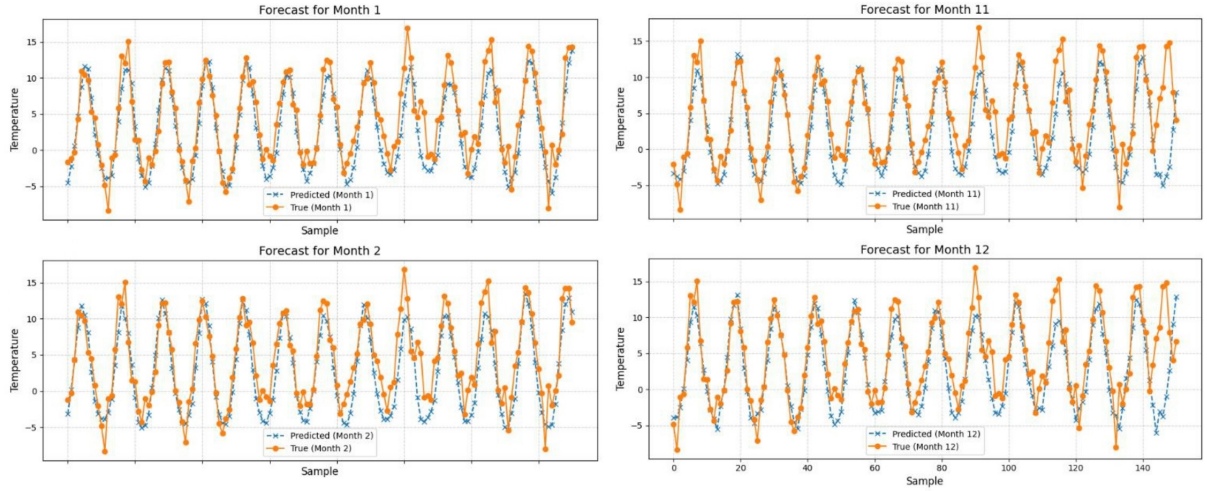


Figure 7: Depicts the first two and the last two months of prediction vs truth values for the RNN forecast.

Model	Test MSE
LSTM	11.32
Naive Baseline	9.599
RNN	10.32

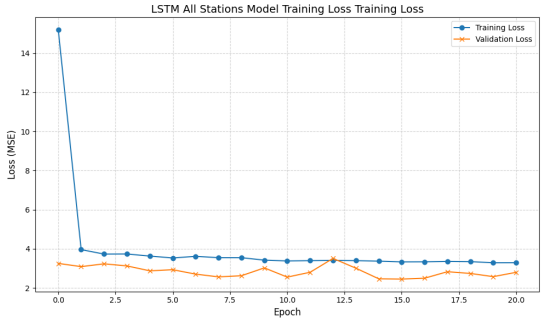
Table 4: Comparison of test MSE for different models.

It is clear from Figure 6 and Figure 7 that both models capture the shape and amplitude of the real data. They show a strong correlation to the ground truth, but the MSE results in Table 4 suggest that these models perform worse than our naive baseline. This suggests that there might not be enough data for the models to capture the temporal patterns accurately. This is the motivation for the next section, where the models are trained on a combined data-frame of 5 stations in Italy for 1-year and 10-year horizons.

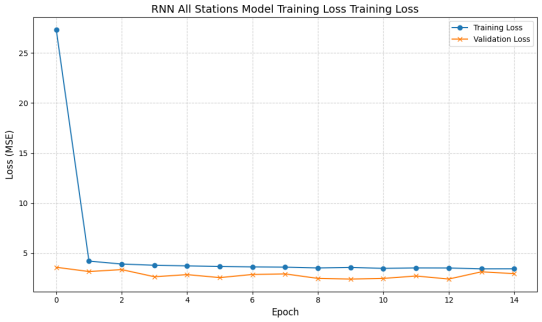
3.3 Using all stations in Italy

For this section, we incorporate the data from all the stations available in Italy to increase the training set for the models. This is done to try and improve the MSE results. Certain problems arise with this approach which will be discussed along with the results.

3.3.1 Results: Forecast horizon = 12



(a) LSTM Model training loss over the epochs for all stations.



(b) RNN Model training loss over the epochs for all stations.

Figure 8: Comparison of two model performances for all stations.

The LSTM's training and validation loss in Figure 8 displays a similar trend to Figure 5. We can see the loss start at higher values and drop sharply. By about 5 epochs, both curves plateau and converge and by epoch 10 again, the early stopping is initialised due to no appreciable improvements. These observations again show no sign of significant over-or under-fitting due to a negligible gap between the curves.

Similarly, the RNN's training and validation loss in Figure 8, shows a steep early decline and plateaus somewhat earlier than the LSTM, again in-line with the single-station results. The key difference here is that both models are now learning from all stations. This often yields a higher initial loss due to more varied data but allows the networks to capture patterns shared across the stations. To draw final conclusions, it's necessary to put these results into perspective by referencing the forecast plots in Figure 9 and Figure 10 for both models, followed by a comparison of their MSE against the new baseline MSE in Table 5.

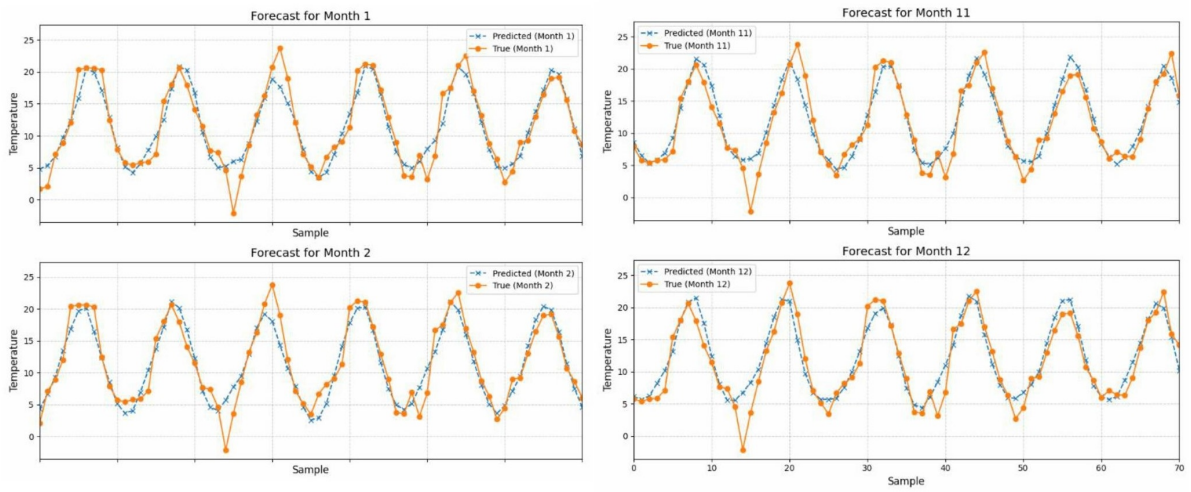


Figure 9: Depicts the first two and the last two months of prediction vs truth values for the LSTM forecast for all stations.

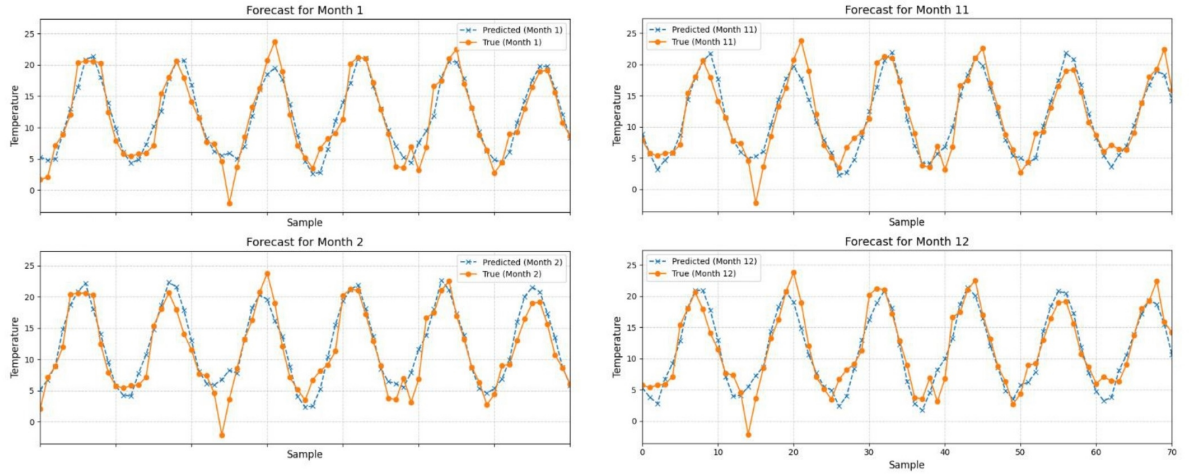


Figure 10: Depicts the first two and the last two months of prediction vs truth values for the RNN forecast for all stations.

Model	Test MSE
LSTM	4.985
Naive Baseline	5.133
RNN	4.890

Table 5: Comparison of test MSE for for all stations.

Similarly, the results in the previous section, Figure 9 and Figure 10, show a strong correlation between the ground truth and the predictions, but now with fewer deviations towards higher time steps. Table 5 confirms the improvements of the models predictive accuracy this time as both model perform better than the baseline test which means they are effectively capturing the key patterns of the climate between all the stations. Conse-

quently, we can say that increasing the size of the data-frame affected the performance of the models positively.

3.4 Results: Forecast horizon = 120

The results presented in this section will only be for the LSTM model. This is because a simple RNN is inferior to an LSTM for a larger forecast horizon and results in an explosion of the gradient.

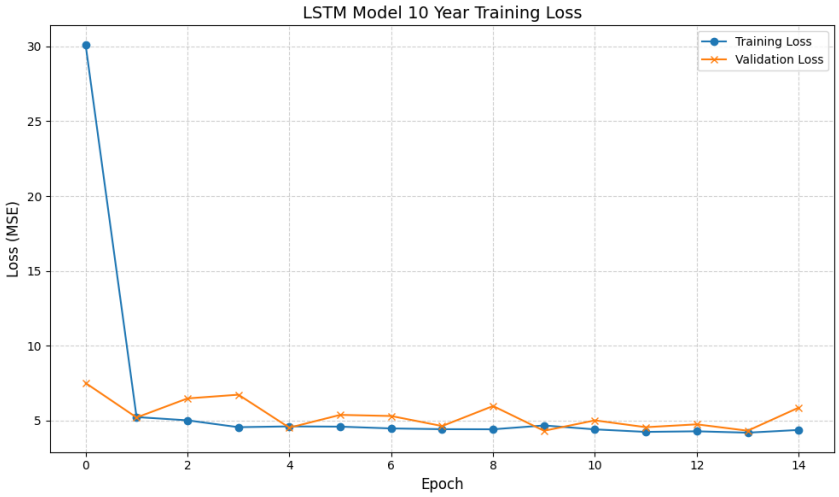
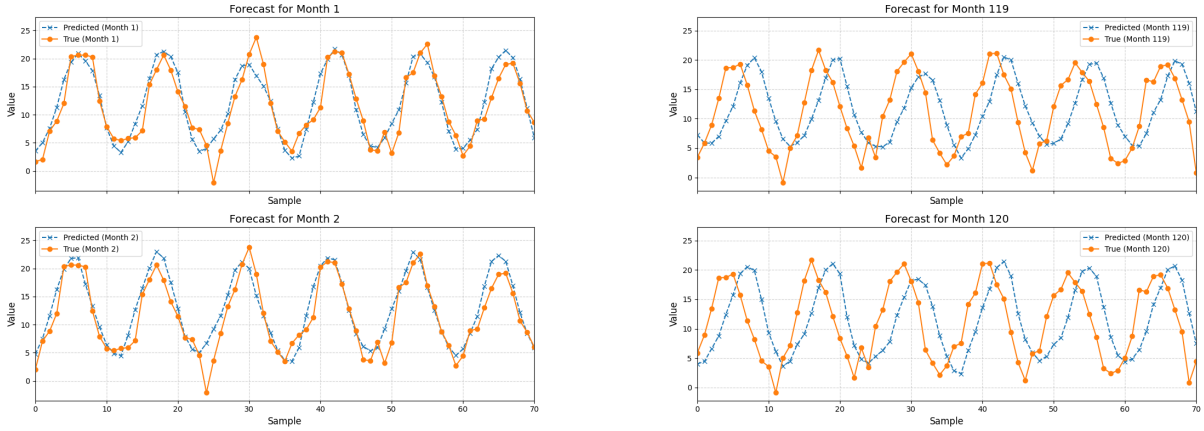


Figure 11: LSTM Model training loss over the epochs for all stations for 10 years.



(a) First two months

(b) Last two months.

Figure 12: Forecast vs ground truth per month for LSTM for 10 years.

From Figure 11, it appears the model is capturing the patterns of the data over 10 years quite well as both curves have a similar shape and they converge. However, Figure 12 shows the pattern is only captured in the first few months, the last months have a clear misalignment in the phase of the curves and the amplitude. This implies that the model is over-fitting and not actually learning the patterns of the data after a certain time step.

This result is not out of the ordinary as the dataset is quite small for a forecast horizon this large. Improvements to the prediction could be achieved by increasing the input data, i.e. using data from 10 years before. Moreover, increasing the size of the training set will also positively affect the results. Finally, further hyperparameter tuning and/or increasing the layer number of the LSTM could result in better curves, but sometimes increasing the complexity of the model might not benefit the prediction accuracy.

4 Station removal experiments and sensitivity analysis

This section assesses the robustness and contribution of the individual stations to the data-frame. As discussed before, each .dly file has different holes. Each station has different co-ordinates and elevation. Therefore, it is crucial to understand which station affects the MSE results of the models positively. Here, we return to a forecast horizon of 12 months and find the best combination of stations that yield the lowest respective MSE for each model.

4.1 Results

Stations Removed	Initial MSE before removal	New MSE after removal	Percentage improvement/decline
['IT000016134']	Baseline LSTM MSE: 5.7305 Baseline RNN MSE: 4.7807 Baseline Naive MSE: 5.1332	Naive Baseline MSE_134: 4.8246 LSTM MSE_134: 3.6822 RNN MSE_134: 4.0239	LSTM_134: 23.68% RNN_134: 16.60%
['IT000016134', 'IT000160220']	Baseline LSTM MSE: 5.7305 Baseline RNN MSE: 4.7807 Baseline Naive MSE: 5.1332	Naive Baseline MSE_134_220: 4.1886 LSTM MSE_134_220: 4.1823 RNN MSE_134_220: 4.3119	LSTM_134_220: 0.15% RNN_134_220: -2.95%
['IT000016134', 'IT000016550']	Baseline LSTM MSE: 5.7305 Baseline RNN MSE: 4.7807 Baseline Naive MSE: 5.1332	Naive Baseline MSE_134_550: 5.6034 LSTM MSE_134_550: 4.0468 RNN MSE_134_550: 4.0705	LSTM_134_550: 27.78% RNN_134_550: 27.36%
['IT000016134', 'IT000162580']	Baseline LSTM MSE: 5.7305 Baseline RNN MSE: 4.7807 Baseline Naive MSE: 5.1332	Naive Baseline MSE_134_580: 4.2953 LSTM MSE_134_580: 2.8282 RNN MSE_134_580: 2.7203	LSTM_134_580: 34.16% RNN_134_580: 36.67%
['IT000016134', 'IT000162580', 'IT000162240']	Baseline LSTM MSE: 5.7305 Baseline RNN MSE: 4.7807 Baseline Naive MSE: 5.1332	Naive Baseline MSE_134_580_240: 4.6190 LSTM MSE_134_580_240: 5.7153 RNN MSE_134_580_240: 5.2970	LSTM_134_580_240: -23.74% RNN_134_580_240: -14.68%

Table 6: Comparison of baseline vs. new MSE after station removal

After examining Table 6, we can see the following:

- Removing stations IT000016134 and IT000162580 resulted in the highest percentage improvement. If we look back to Table 1, we can see that these stations have most years and months missing respectively. Therefore, this result is logical as the datasets from these stations would provide a significant number of holes in our final combined data-frame.
- Removing stations IT000016134, IT000162580 and IT000162240 resulted in the highest percentage decrease. At first sight, after looking at Table 1, we would

assume that as these stations have the most holes, removing all of them will yield the best result. However, after removing a certain number of stations, we reduced the size of the training data to the point where the models struggled to capture the temporal pattern of the climate between the stations. Another reason for this result could be that the remaining stations do not share similar data patterns, which affects the accuracy of the models negatively.

- The location of the stations does not seem to play an extremely high role until the dataset is small enough that the prediction accuracy solely depends on the correlation of the regional climate between the stations. This observation comes from the fact that removing stations IT000016134 and IT000162580 resulted in the greatest percentage increase even though these stations are much closer to the others compared to IT000016550 and IT000160220.

5 Conclusions and future improvements

The results demonstrate that both RNN and LSTM architectures can effectively capture the monthly temperature patterns for a one-year horizon, especially when data from multiple stations is combined. In single station settings, the baseline outperformed the models, indicating that the sparsity and inconsistency of the data negatively affect the learning accuracy of the models. Meanwhile, extending the forecast horizon to 10 years results in over-fitting likely due to similar factors. Combining the data-frames and analysing the contributions of each station revealed a combination of stations that yielded the highest accuracy.

Moving forward, an immediate step is to incorporate more stations with consistent datasets i.e. without holes. Potentially, introducing other variables from the GHCN-Daily data-frame (e.g., precipitation, snowfall) will help capture more of the underlying climatic dynamics. Utilization of early stopping and proper hyperparameter fine-tuning yielded good results, but a method to extrapolate the data to fill the holes in the data-frames could yield more complete results. Also, incorporating a more robust mechanism that takes into account the location and elevation of the stations could increase the accuracy of the model further. These refinements, together with continued effort to clean and expand the dataset, will pave the way for more robust and reliable temperature forecasting methods.

References

[1] Ilan Price et al., "Probabilistic Weather Forecasting with Machine Learning", (2024), [DOI: 10.1038/s41586-024-08252-9].

[2] Science Museum. *Weather Forecasting and Climate Modelling: A Short History*. https://www.sciencemuseum.org.uk/objects-and-stories/weather-forecasting-and-climate-modelling-short-history. Accessed: 2024-03-22. 2024.

[3] Matthew J. Menne et al., "An Overview of the Global Historical Climatology Network-Daily Database", 29 (2012) pp. 897–910, [DOI: 10.1175/JTECH-D-11-00103.1].

[4] Anatoliy Doroshenko et al., "Machine Learning to Improve Numerical Weather Forecasting", (2020) pp. 353–356, [DOI: 10.1109/atit50783.2020.9349325].

[5] Jasmine B.D. Jaffrés. *GHCN-Daily: A Treasure Trove of Climate Data Awaiting Discovery*. https://doi.org/10.1016/j.cageo.2018.07.003. Computers & Geosciences, vol. 122, Jan. 2019, pp. 35–44. Accessed: 2019-05-05. 2019.

[6] Robin Schmidt. *Recurrent Neural Networks (RNNs): A Gentle Introduction and Overview*. https://www.example.com. Accessed: 2025-04-01.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[8] The IoT Academy. *Main Difference between RNN and LSTM–(RNN vs LSTM)*. https://www.theiotacademy.co/blog/what-is-the-main-difference-between-rnn-and-lstm/. 17 July 2023. 2023.