# Deep Learning Based Weather Prediction Architecture Comparison

Math 156, Winter 2024

David Spector, Melody Chen, Carter Harrison, Gavin Joyce

https://github.com/davidspector67/weather_predictor/tree/main

## 1. Introduction

### 1.1 Objective

Weather forecasts are of great use to society, providing helpful information for day to day life. Weather forecasts rely on complex physics simulations, requiring large amounts of computational power [1]. Moreover, these rule-based models require manual input and tracking, and meteorologists require radar, satellite, and other supplemental data to aid them [2]. However, more recently, machine learning techniques have been explored and implemented to create more sophisticated weather models[3]. Machine learning opens the potential to build weather forecasting models more accurately while being more efficient and less labor-intensive. To do this, we explored several model architectures and metric combinations to forecast hourly average temperatures given short-term historical data.

### 1.2 Dataset

This project considers data from the Global Integrated Surface Database (ISD), managed by The National Oceanic and Atmospheric Administration[4]. This database contains raw hourly and sub-hourly observations from over 35,000 weather stations around the world. This dataset was carefully chosen over other historical data as it directly sourced from individual weather stations, without any aggregations or data cleansing. Each row in the dataset represents a singular capture of the weather stations' sensors at a specific time, without modification. This is important as it ensures that no other pre-processing has been applied to the data, allowing more control over the method of aggregation. Furthermore, such unprocessed data is what is available when forecasting future weather data at a current point in time.

Since the model is forecasting data at a singular point (LAX), five nearby weather stations were chosen strategically. Each weather station chosen is from an airport, as they tend to have more frequent observations compared to other nearby stations.

1. LAX (Los Angeles International Airport), chosen as it is the location the model is predicting.
2. SAN (San Diego International Airport), as it is south of LAX, and a similar climate.

3. PAM (Palm Springs Regional Airport), as it is directly east of LAX, and has a variable climate that may influence the weather in Los Angeles.
4. LAN (Lancaster Regional Airport), as it is directly north of LAX, and also has a variable climate that may influence the weather in Los Angeles.
5. CAT (Catalina Airport), as it is west of Los Angeles. Since winds often tend east off of the coast, this station may act as a preview of weather soon to hit Los Angeles.

For each of the above weather stations, the model considers hourly, and sub-hourly metrics of:

- Air temperature (F)
- Sky coverage (oktas)
- Dew point (F)
- Sea level pressure (millibar)
- Visibility (miles)
- Precipitation amount (in)
- Wind speed (mph)
- Wind direction
- Time of observation

from the past 6 calendar years (2023, 2022, 2021, 2020, 2019, 2018). The model aggregates, windows, and pre-processes this data detailed in **Section 2**. Furthermore the dataset is divided into 3 subsets as follows: 2018-2021 data is used for training, 2022 data is used for validation, and 2023 is used as test data. In total this encompasses 2,190 (365*6) hourly data points.

### 1.3 Tools

The project was written in Python, inside of a Jupyter Notebook found here. The models utilized the machine learning framework TensorFlow to create and train all models in our experiments. Parsing the integrated surface hourly (ISH) files from the ISD database uses the package ish_parser[5].

## 2. Methodologies

To find the optimal machine learning setup to predict the average hourly air temperatures in LA for one day in the future, we experimented with model architecture, input parameters, regularization techniques, and the number of hours in the past to consider before making our predictions. For simplicity, each 24 hour weather forecast window will start and end at midnight.

## 2.1 Data Preprocessing

While temperature typically cycles across each 24 hour period, time of year is also crucial to daily temperature values in LAX[6]. To take this into account in our models, we decided to model time of year continuously using a sine and cosine curve where the period of each of these curves amounts to exactly one year. This provides a unique combination of sine and cosine values denoting the exact time of year for each day, enabling our models to learn annual weather patterns in daily temperature forecasts. Similarly, we also applied the same sine and cosine preprocessing technique to wind direction.

Also, the ISH dataset we used contained a small amount of randomly distributed missing weather observations as well as some observations with multiple data entries. To mitigate this, we averaged all observations for each hour and handled missing observations by using the previous value in its place. This ensures that no future data is a factor in the present observation, which is needed for accurate time series training data.

Next, we standardized both inputs and outputs by metric so that all inputted metrics were inputted equally regardless of range and variance, and that range of our outputs match the range of our inputs for best results [7], [8].

Finally, to avoid potential patterns in our training data, we shuffled the order of training inputs and outputs before feeding them into the model.

## 2.2 Data Input Window

One of the hyperparameters we look to optimize in this project is the total number of hours in the past to use as inputs for time series forecasting. Due to the limited size of our dataset, the tradeoff we will be manipulating is between the number of windows which can be used for training, validation, and testing and the size of the windows our models can learn from before making predictions. Note that since we’re only estimating 24 hour temperature values starting at midnight, we maximize the number of windows we can use by setting our window values as multiples of 24.

Additionally, we also experimented with including weather metrics from only LAX and metrics from nearby cities to see if this additional data would positively affect LAX temperature predictions.

## 2.3 Model Architectures

Given the sequential nature of our time series data as well as the potentially long window sizes, we decided to explore the applicability of an LSTM model architecture. In contrast, we also tested the Multilayer Perceptron (MLP) architecture to explore potential non-sequential correlations between weather metrics of the previous days and the temperature of the next day, which we hypothesized to exist. Additionally, we implemented a hybrid model which uses both LSTM and Dense layers to determine if a happy medium approach could provide the best of both worlds.

Hyperparameter tuning and regularization techniques were conducted on each approach independently in order to compare optimal results from each architecture on the same dataset.

### 2.4 Model Comparisons

To simultaneously explore the effect of input window size, model architecture, and included input metrics, we split up model training into four experiments:

- **Experiment 1:** 24 hour input windows with only LAX metrics
- **Experiment 2:** 72 hour input windows with only LAX metrics
- **Experiment 3:** 24 hour input windows with LAX and neighboring station metrics
- **Experiment 4:** 72 hour input windows with LAX and neighboring station metrics

Models from each of our three architectures were trained and tuned in these experiments, and the resulting MSE and MAE values on test data were used to draw conclusions about the optimal architecture, input window size, and included input metrics.

While in theory, additional input metrics and input window sizes should either improve or not affect model accuracy, the small amount of data we have can result in the model overfitting to unhelpful features. Thus, these experiments will also help determine the most promising metrics and window size for temperature prediction models at a larger scale.

### 2.5 Loss & Accuracy Measures

To train and determine the accuracy of our model, we decided to use mean squared error (MSE) as our loss function because of its increased penalty on very poor predictions. Due to the fact that very poor predictions can prove detrimental to the useability of weather forecasting models and throw off entire prediction windows [9], this scheme enables our models to aim for more small errors and less large ones.

To ensure that model predictions aligned with the original air temperature metrics, model outputs were de-standardized before being returned as air temperature predictions. These post-processed results were used to assess and compare the MSE and mean absolute error

(MAE) of each model to provide insight into the robustness of our predictions in each experiment. We decided to include MAE as an additional prediction accuracy metric because of its intuitive characterization of model quality in the context of real-world use.

### 2.6 Early Stopping & Regularization

Due to the complex nature of the models we experimented with, many required a large number of epochs to converge. To allow the model to train to convergence without training for too long and inducing overfitting, we used early stopping to stop our model from training once its loss function indicates that it has converged.

We also used ridge regression (L2) to combat overfitting in all of our models while maintaining complexity. We decided to use ridge regression instead of lasso because of the inherent dependent features in weather data that we used in our model inputs [10]. For example, cloud cover and temperature are intertwined with each other.

## 3. Results

### 3.1 MSE and MAE Comparisons

The following table shows both MSE and MAE of the denormalized data in each model (LSTM, Hybrid, MLP) for each experiment. Each model was trained on the same data in the process described in the Methodologies section. In all experiments, the MLP model outperformed the LSTM and Hybrid model in both MAE and MSE. Also notice that the LSTM and Hybrid models had comparative MAE and MSE in each experiment. Moreover, see that experiment 3 produced the best results in terms of MAE and MSE for the MLP model.

The MLP were significantly more accurate than the LSTM and Hybrid model in all experiments, as the Experiment 1 had the closest results between the MLP and all other models, which had a MSE difference between the MLP and LSTM models of roughly 1.4 degrees Fahrenheit squared.

**Figure 3.1: Comparison Metrics**

| **Experiment 1**<br>**24 hrs of all metrics at LAX** | MAE | MSE |
|---|---|---|
| LSTM | 2.048329 | 7.987299 |
| Hybrid | 2.136263 | 8.274816 |

| | | |
|---|---|---|
| MLP | 1.874671 | 6.578485 |
| **Experiment 2**<br>**72 hrs of metrics at LAX** | | |
| LSTM | 2.772549 | 14.894351 |
| Hybrid | 2.485569 | 11.389804 |
| MLP | 2.044712 | 7.535338 |
| **Experiment 3**<br>**24 hrs of metrics at all weather stations** | | |
| LSTM | 2.175356 | 8.887671 |
| Hybrid | 2.172691 | 8.869209 |
| MLP | 1.853091 | 6.277174 |
| **Experiment 4**<br>**72 hrs of metrics at all weather stations** | | |
| LSTM | 2.331207 | 10.068363 |
| Hybrid | 2.469323 | 10.921197 |
| MLP | 1.958351 | 6.822703 |

**Figure 3.1: Comparison Metrics**: This table shows the MSE and MAE of each model (LSTM, Hybrid, MLP) for each experiment. The table also gives a brief description of each experiment (more detail can be found in the Methodologies section). Analysis of this table is found in the Discussion section.

### 3.2 Forecast Examples for Experiment 3

Below are figures of graphs that show the prediction at a randomly selected point in time for all of the models considered in Experiment 3 – LSTM, Hybrid, and MLP – alongside with the actual temperature, labeled “Truth.” The red bar in the middle represents where the forecasting starts. Hence, the 24 hours of temperature data fed into the model is on the left hand side of the red bar.

**Figure 3.2.1: Forecast Example 1 for Experiment 3**

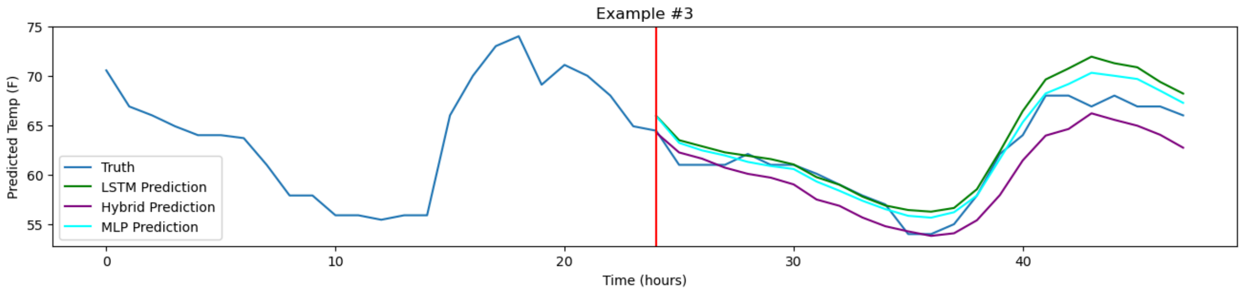


**Figure 3.2.1: Forecast Example 1 for Experiment 3**: In this graph we see that all models generally capture the trend and shape of the real temperature. MLP is the closest of all models to the actual temperature, while the LSTM prediction is above the MLP, and the Hybrid model is below.

**Figure 3.2.2: Forecast Example 2 for Experiment 3**

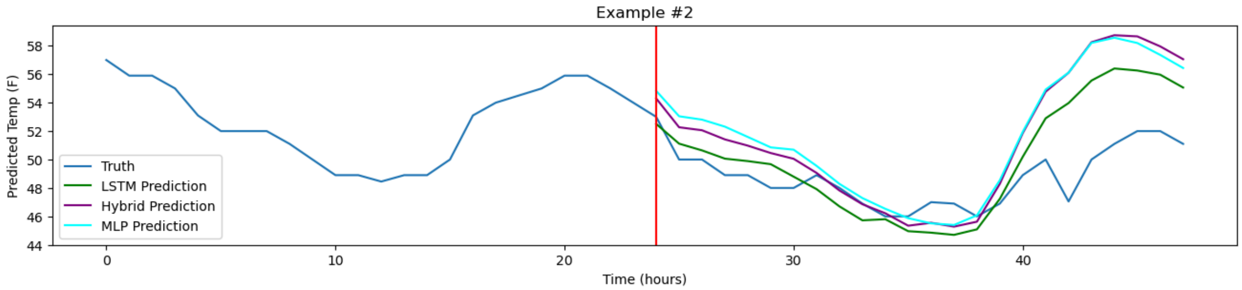


**Figure 3.2.2: Forecast Example 2 for Experiment 3**: In this graph, from hour 24 to 35, the models are close to the actual temperature. However after that benchmark, the predicted temperature deviates from the true forecast for all models.

**Figure 3.2.3: Forecast Example 3 for Experiment 3**

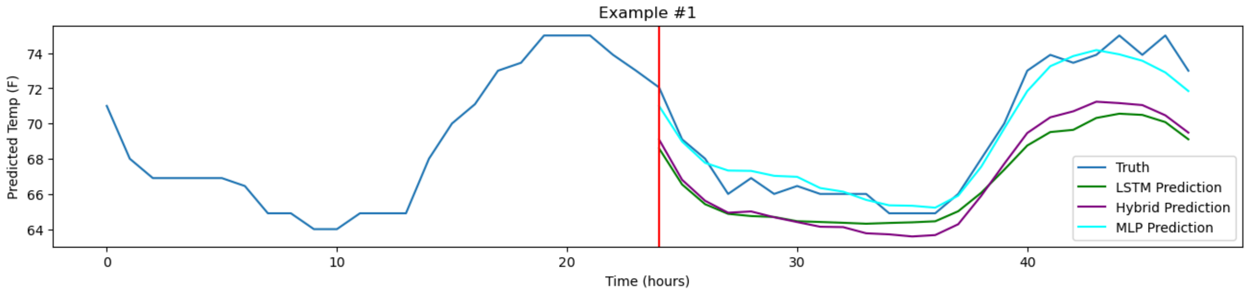


**Figure 3.2.3: Forecast Example 3 for Experiment 3**: In this graph, the MLP model prediction is generally close to the truth. However, the hybrid and LSTM model, while capturing a similar shape to the actual temperature, is notably and consistently lower than the actual temperature.

# 4. Discussion

## 4.1 Effects of Regularization

The results of ridge regression can be seen below in Figure 4.2.1, as the training loss and the validation loss diverge in the training of the hybrid model in Experiment 3 – a symptom of the model overfitting. Specifically, the training loss decreasing (overfitting), and the validation loss increasing (becoming less accurate). The results of adding L2 regularization is seen in Figure 4.2.2, where the validation loss and training loss are in parallel. This loss chart is from the training of the best- performing model (MLP model in Experiment 3). The following figures show both the loss on the training set (MSE+L2) and the loss on the validation set (MSE) over the number of epochs while training.

**Figure 4.2.1: Experiment 3 Hybrid Model Without Regularization**

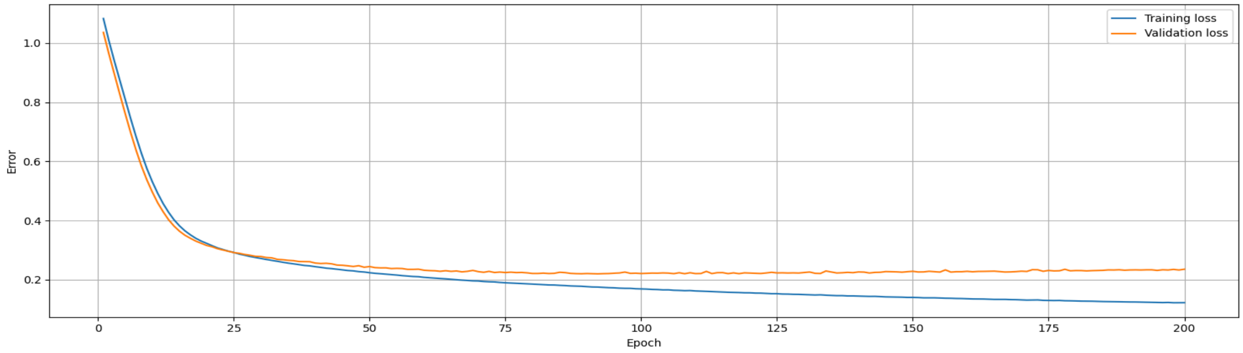


**Figure 4.2.1: Experiment 3 Hybrid Model Without Regularization**: This graph shows the loss (MSE+L2) on the training set below the loss (MSE) on the validation set. The figure shows the loss of the validation set and training set diverging over the number of epochs (200).

**Figure 4.2.2: Experiment 3 Hybrid Model with Suitable L2 Scaling Coefficient**

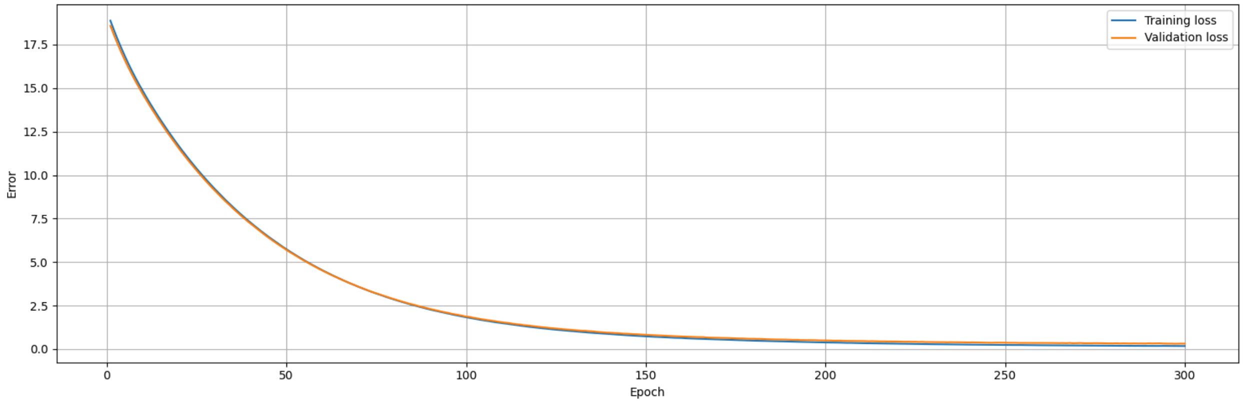


**Figure 4.2.2: Experiment 3 Hybrid Model with Suitable L2 Scaling Coefficient:** This graph shows the loss (MSE+L2) on the training set and the loss (MSE) on the validation set. The figure shows the loss of the validation set and training set in parallel with each other the the number of epochs (300).

### 4.2 Model Forecast and Actual Temperature Comparison

As shown in section 3.2, the models' forecasts generally capture the shape of the actual temperature well with predicted temperature falling within a reasonable range of the actual temperature. We see some instances (such as figure 3.2.3) where some models consistently overestimate or underestimate the actual temperature. In other scenarios, models may capture the true temperature accurately for only certain time frames in the future (figure 3.2.2). This may be due to the variability and unpredictable nature of localized weather data and/or lack of relevant data and context in model training – the most complex of the models only observes weather from 5 nearby weather stations, and hence is unable to predict large scale events (e.g. a powerful storm)

In visual inspection of the graphs in section 3.2, the MLP model captures the truth best, supporting our findings in section 4.4 that MLP is the most effective model. However, as highlighted in figure 3.2.1, all models were able to learn the general shape and trend in each prediction.

### 4.3 Benchmark to Real World Forecasts

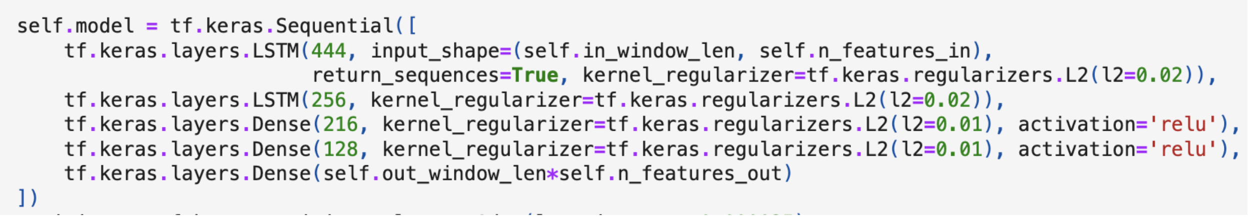
Currently, meteorologists use standard models to forecast weather and predict temperature which rely on complex physics simulations. These forecasts (3-day) typically have MAEs under 3 degrees [14], however this accuracy gradually decreases past 3 days in the future. For our best model (MLP Model, Experiment 3), the MAE was 1.85 degrees for 24 hours into the future (Figure 3.1) which is in the range of the accuracy of current weather forecasts. However, our model was trained and tested on a single point in Southern California, where the climate is predictable. We also chose to only predict 24 hours into the future, which is relatively short compared to real world weather forecasts. However, a machine learning model is more efficient and less labor-intensive in analysis than what is currently done to forecast weather and with further improvement, has the potential to improve weather forecasting processes.

### 4.4 Comparing MLP, LSTM, and Hybrid Models

Based on **Figure 3.1: comparison metrics**, we found that the optimal combination of input window length, input metrics, and model architecture is an MLP with 24 hours of metrics from all stations, as the model had the lowest MSE score of 6.27 and lowest MAE of 1.85.

One reason for this could be the ability of an MLP to access the past as an input while an LSTM-based architecture must make predictions sequentially[11], causing it to disregard past inputs it deems unimportant when making predictions. Furthermore, we observed that hybrid and LSTM models had correlated results across all experiments, which follows as the first two layers of the hybrid model are LSTM layers followed by Dense layers (Figure 4.1.1). This ordering ensures that all inputs sent to the Dense layers of this model are based on sequential observations, inhibiting the ability of the Dense layers to make non-sequential connections.

**Figure 4.1.1: Hybrid Model Layer Reference**



```
self.model = tf.keras.Sequential([
    tf.keras.layers.LSTM(444, input_shape=(self.in_window_len, self.n_features_in),
                         return_sequences=True, kernel_regularizer=tf.keras.regularizers.L2(l2=0.02)),
    tf.keras.layers.LSTM(256, kernel_regularizer=tf.keras.regularizers.L2(l2=0.02)),
    tf.keras.layers.Dense(216, kernel_regularizer=tf.keras.regularizers.L2(l2=0.01), activation='relu'),
    tf.keras.layers.Dense(128, kernel_regularizer=tf.keras.regularizers.L2(l2=0.01), activation='relu'),
    tf.keras.layers.Dense(self.out_window_len*self.n_features_out)
])
```

**Figure 4.1.1: Hybrid Model Layer Reference:** The code above is found in the model definitions of the Jupyter Notebook for the Hybrid Model.

Another reason the MLP was able to outperform the LSTM-based approaches was that the primary temperature oscillation behavior (i.e. days and seasons) were explicitly accounted for in our experimental design and added features. Had these not been accounted for or if other impactful patterns in the data were found, the LSTM could have found this to gain an edge over an MLP model for a large enough input window size due to their ability to learn oscillation behavior [12].

Also, a separate key finding from our results is that for each combination of metrics and architectures, all models became worse after moving the input window length from 24 to 72. For each model, the MSE increased (Experiment 1 compared to Experiment 2, and Experiment 3 compared to Experiment 4, in Figure 3.1). While this could suggest that the most recent inputs are the most important in predicting future weather, the more generalizable and well-founded interpretation of this event would be that in the tradeoff between the window size and number of windows mentioned in the Methodologies Section, the number of windows proved to be more important given the size of the dataset we used. We expect that with a larger dataset, the accuracy of 72 hour windows will be more than or equal to the accuracy of 24 hour windows.

When considering variations in input metrics for fixed window inputs, we observed increased accuracy after adding more locations as inputs for all cases except for the Hybrid and LSTM models between experiments 1 and 3, which had a small decrease in accuracy (Figure

3.1). This suggests that the LSTM-based models needed a window input size greater than 24 to differentiate additional helpful inputs from irrelevant ons. Thus, for all LSTM-based models with window input size of 72 and all MLP models in general, this suggests that these models were able to accomplish more than overfitting to the parameters they were given, showing positive potential for these machine learning setups.

In conclusion, given our MAE of 1.85 and our MSE of 6.28 (Figure 3.1) for the MLP model in Experiment 3, we have shown the MLP architecture to be a promising solution to the temperature time series forecasting problem, despite the limited data we worked with.

### 4.5 Future Directions

To better grasp the true potential behind this MLP approach for weather forecasting, a useful next step would be to aggregate more and cleaner data spanning additional years. More data could provide an easy starting point for other weather metric predictions to enable a fuller picture for machine learning based weather forecasting.This increased data could also extend to locations in other parts of the world, increasing usability and real world applications. Moving in this direction could further determine whether or not it would be good practice to include as many locations as possible to train our models, or if focusing on additional locations in specific climates is more effective. We could also determine if our methods and models might be more effective in forecasting weather in areas with more volatile weather.

To extend the models' forecasting flexibility we could train the models on non-quantized windows – rather than training the model on the past 24 since *midnight* to predict the next 24 hours, train the model at *any point in time*. This would allow the model to give a 24 hour forecast into the future at any time. To do so, we would shift the data window over by 1 hour, instead of the input window size. Furthermore, we will also need to feed the model the time of day similar to how the model is currently fed the time of year, as described in the Methodologies section.

Expanding our models' ability to provide spatial analysis as part of weather predictions is another promising possibility. The MetNet model[13] gives an in-depth example of the direction such an implementation could explore. Developing a convolutional LSTM similar to the one employed in the MetNet model would allow us to capture spatial (CNN) and temporal (LSTM) patterns in parallel and reveal complex dependencies across time and space, increasing the scope of our work.

Finally, to determine whether or not LSTMs have a useful place in weather forecasting, we could predict weather phenomena taking place on a monthly scale such as tide height and rainfall to take advantage of the architecture's ability to detect long-term oscillations. An experiment like this could include sequential data at a much larger scale, which is better tailored towards the LSTM architecture’s strengths.

## References

1. Chen, Liuyi, et al. “Machine Learning Methods in Weather and Climate Applications: A Survey.” *MDPI*, Multidisciplinary Digital Publishing Institute, 3 Nov. 2023, www.mdpi.com/2076-3417/13/21/12019.
2. “Forecast Process.” *National Weather Service*, 31 July 2017, www.weather.gov/about/forecast-process.
3. Chantry, Bouallegue et al. “The Rise of Machine Learning in Weather Forecasting.” ECMWF, 21 June 2023, www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting.
4. “NOAA Hourly Weather Data.” *Index of /Data/Global-Hourly*, National Oceanic and Atmospheric Administration, www.ncei.noaa.gov/data/global-hourly/. Accessed 13 Mar. 2024.
5. Haydenth. "haydenth/ish_parser: Parser for NOAA ISH Files." GitHub, 15 Mar. 2024, https://github.com/haydenth/ish_parser
6. National Centers for Environmental Information (NCEI). "Past Weather." National Oceanic and Atmospheric Administration, 15 Mar. 2024, https://www.ncei.noaa.gov/access/past-weather/lax
7. Sarle, Warren S. "comp.ai.neural-nets FAQ, Part 2 of 7: Learning." FAQs.org, 3 Mar. 2024, http://faqs.org/faqs/ai-faq/neural-nets/part2
8. Bishop, Christopher M. *Neural Networks for Pattern Recognition*. Oxford University Press, 2013.
9. Alake, Richmond. “Loss Functions in Machine Learning Explained.” *DataCamp*, DataCamp, 24 Nov. 2023, www.datacamp.com/tutorial/loss-function-in-machine-learning.
10. Parr, Terence. “A Visual Explanation of MLP Model Regularization.” The Difference between L1 and L2 Regularization, explained.ai/regularization/L1vsL2.html.
11. Brownlee, Jason. “On the Suitability of Long Short-Term Memory Networks for Time Series Forecasting.” *MachineLearningMastery.Com*, 5 Aug. 2019, machinelearningmastery.com/suitability-long-short-term-memory-networks-time-series-forecasting/.

12. Gers, Felix A., et al. “Applying LSTM to time series predictable through time-window approaches.” *Artificial Neural Networks — ICANN 2001*, 2001, pp. 669–676, https://doi.org/10.1007/3-540-44668-0_93
13. Chen, Liuyi, et al. “Machine Learning Methods in Weather and Climate Applications: A Survey.” *MDPI*, Multidisciplinary Digital Publishing Institute, 3 Nov. 2023, www.mdpi.com/2076-3417/13/21/12019.
14. “Assessing Forecast Accuracy.” *Assessing Forecast Accuracy* | *METEO 3: Introductory Meteorology*, PennState Department of Meteorology and Atmospheric Science, www.e-education.psu.edu/meteo3/node/2285. Accessed 13 Mar. 2024.