Predicting Upper Atmospheric Weather Conditions Utilizing Long-Short Term Memory Neural Networks for Aircraft Fuel Efficiency

THESIS

Garrett A. Alarcon, 1st Lieutenant, USAF

AFIT-ENS-MS-20-M-129

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-20-M-129

PREDICTING UPPER ATMOSPHERIC WEATHER CONDITIONS UTILIZING LONG-SHORT TERM MEMORY NEURAL NETWORKS FOR AIRCRAFT FUEL EFFICIENCY

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Garrett A. Alarcon, B.S.

1st Lieutenant, USAF

March 2020

AFIT-ENS-MS-20-M-129

PREDICTING UPPER ATMOSPHERIC WEATHER CONDITIONS UTILIZING LONG-SHORT TERM MEMORY NEURAL NETWORKS FOR AIRCRAFT FUEL EFFICIENCY

THESIS

Garrett A. Alarcon, B.S.
1st Lieutenant, USAF

Committee Membership:

Lt Col A.J. Geyer, Ph.D.
Chair

Dr. Raymond Hill, Ph.D.
Reader

Abstract

Aviation fuel is a major component of the Air Force (AF) budget, and vital for the core mission of the AF. This study investigated the viability of Long-Short Term Memory (LSTM) networks to increase the accuracy of deterministic numerical weather prediction (NWP) models, while also investigating the ability to reduce model generation time. Increased forecast accuracy for wind speeds could be implemented into existing flight path models to further increase fuel efficiency, while reduced modeling times would allow flight planners to generate a flight plan in rapid response situations. The most viable model consisted of an ensemble of six LSTMs trained off six coordinates. The model's error was on average +1.2 m/s higher than the deterministic NWP with a computation time of 1.85 s. The LSTM generated a flight path that was on average 14.2 min slower for an approximately 7 hour 32 min flight. This forecast generation took seconds to complete compared to hours from the deterministic model. While the LSTM architecture in this study was not able to increase forecast accuracy, the speed at which it generates an approximately close forecast can be an integral tool for flight planners in the future.

Table of Contents

Page

List of Figures

List of Tables

PREDICTING UPPER ATMOSPHERIC WEATHER CONDITIONS UTILIZING LONG-SHORT TERM MEMORY NEURAL NETWORKS FOR AIRCRAFT FUEL EFFICIENCY

I. Introduction

1.1 Background

The Department of Defense (DoD) has an obligation to the American people to be stewards of their tax dollars in all defense related spending. As such, the DoD is always searching for ways to minimize spending while also increasing combat capabilities, military readiness, and operational effectiveness. Aircraft are an integral part of the United States Air Force (USAF) and by its very nature, fulfilling these goals incurs a substantial cost for procuring and consuming fuel. From the 2019 fiscal year budget for the DoD, $24 billion was requested for fuel consumption with $6.6 billion going to operations and $4.5 billion going to transportation [2]. The USAF consumes around half of this budget for aviation fuel with majority of it being used by Air Mobility Command (AMC), a major command (MAJCOM) within the USAF structure [3]. Many of the aircraft within the AMC inventory are responsible for global transportation of cargo and personnel, along with aerial refueling. These operations are not only vitally important to the AF, but also to fulfilling the national defense strategy. Increasing efficiency in fuel consumption within these aircraft can have an immense impact on cost savings for the USAF and the DoD as a whole, while not sacrificing on mission capabilities.

1.2 Overview

There are multitude of ways to address increasing efficiency in fuel consumption among aircraft. This study focuses particularly on headwind predictions in the upper atmosphere relating to mission planning for various air mobility operations. Having accurate predictions for the varying spatial regions of the atmosphere enable mission planners to develop the most fuel efficient route from origin to destination. Thereby being able to optimize altitude and flight path which are natural parameters of constructing a fuel-efficient route [4].

When looking at wind speed forecast there are two distinct classes of weather models to look at; deterministic and ensemble. While a deterministic model is comprised of a single model, an ensemble model is comprised of multiple deterministic models [5]. In an ensemble model, each of its members is initialized with slightly different values for their parameters. This generates a forecast giving a variety of results, which provides keener insights into accurate forecast predictions. The weather data for this study comes from the Global Data Assimilation System (GDAS), which is run by NOAA. The GDAS takes in all available global satellite, conventional (rawinsonde, aircraft, surface), and radar observations to report weather conditions across the globe every six hours. This report details conditions for every latitude and longitude coordinate across 31 different pressure layers. The system is responsible for providing the initial conditions for the deterministic and ensemble weather forecast produced by the global forecast system (GFS) [6]. Currently operations within AMC still rely on the deterministic forecast while NOAA has switched to utilizing results from the ensemble forecast.

Wind speeds are known to follow a nonlinear behavior when modeled over time. Their discontinuous and stochastic nature makes it difficult to provide accurate predictions utilizing linear approximation techniques [7]. Artificial neural networks

(ANN) have been shown to learn the underlying structure of data sets and provide accurate predictions for seemingly complex weather problems [8]. This ability has generated a research surge in investigating the application and development of ANNs to solve varying weather related problems [9].

1.3 Research Objectives

The goal of this study was to explore the viability for LSTMs to model and predict wind speeds either as accurate or more accurate than the deterministic numerical weather prediction (NWP) models. This enables the model to act as an error reducing post processor for current weather models to help reduce the error in their forecasting methods. It would also allow its inclusion into existing fuel optimization algorithms to help assist in further increasing fuel efficiency.

A secondary objective was to explore the LSTMs computational speed advantage over the deterministic NWP model. If the LSTM cannot achieve the accuracy goal, but it is still relatively close then exploring the difference in computing speed can bring an important benefit to flight planners.

The following chapter will discuss background literature relating to the methodologies being deployed within this work, including the difference in weather model types and ANNs.

II. Literature Review

2.1 Overview

There are many different methodologies being employed to construct a model in this study. As such, this section examines some of the previous research done in these areas, along with providing background information on the techniques themselves.

2.2 Weather Models

Two main techniques for weather forecasting are used in the industry today, these being deterministic and ensemble forecasting [7]. A deterministic forecast focuses on making a single forecast of the most likely weather outcomes given the best approximation and modeling of the initial conditions. This is done by having the initial state of the atmosphere established using observational data. Then an atmospheric model simulates evolution from the initial state. From this the output is processed and made available. According to a reference document put out by NOAA this method has some drawbacks due a few reasons relating to error [6]. These being that the equations used by the model do not fully capture processes in the atmosphere, model resolution is not sufficient to capture all features in the atmosphere, the initial observations are not available at every point in the atmosphere, and the observational data cannot be measured to an infinite degree of precision.

In an ensemble forecast, multiple deterministic forecasts are developed representing a set of possible future states. These can be developed in many different ways, but one technique used is to slightly perturb the initial conditions then develop deterministic models from each instance of perturbation. This approach addresses certain sources of uncertainty that are not captured in a deterministic forecast. These being uncertainty introduced as part of imperfect model formulation, and uncertainty

introduced as part of imperfect initial conditions [5].

Ensemble modeling for weather is the current method employed by large organizations, such as NOAA, while deterministic is still used by AMC. Ensemble modeling has been shown to perform better at forecasting than deterministic models in a myriad of applications [10]. For example, Keith and Leyton [11] displayed how ensemble weather models were better predictors of adverse weather conditions, which would require aircrafts to consume more fuel than originally expected. In a study by Taylor and Buizza [12], ensemble forecasting showed higher accuracy levels than deterministic for a one to ten day weather forecast looking at electricity demand. However, ensemble forecasting is not always superior in every instance. An incident in Venice showcased this, where the accuracy for predicting flooding due to storms more than four days out with a deterministic model was comparable to the ensemble model [13]. In another instance, Leonardo and Colle [14] found that a deterministic model gave the lowest total track error when predicting North Atlantic tropical cyclones, even when compared against several different ensemble models. In general, the World Meteorological Organization notes ensemble forecast produce more reliable results than the deterministic forecast, especially when the forecast is for more than 1-3 days out [15]. This is due to the ensemble's ability to capture the uncertainty inherent within the deterministic model.

Since AMC relies on using the deterministic model for forecasts and mission planning, this study focuses on performance metrics comparing techniques used in this study to the deterministic model outputs.

2.3 Weather Factors

While some of the factors in this study may be self-explanatory, others require further detailing. Within the GDAS, wind speeds are expressed in terms of their or-

thogonal velocity components, which is the zonal velocity (u) and meridional velocity (v). If relating to an x-y Cartesian coordinate system, u runs parallel to the x-axis and v runs parallel to the y-axis. Therefore positive u values represent winds blowing east while positive v values represent winds blowing north. These components are then combined using the Pythagorean Theorem to acquire the magnitude of the wind. With the magnitude calculated, it is a simple trigonometric expression to discover the direction, or angle, of the resulting wind vector [16].

Weather measurements are recorded in the GDAS by latitude, longitude, and pressure. Earth's atmosphere can be divided up into multiple layers which are measured for similarity around the globe by their pressure levels as opposed to actual altitude. For example, the upper edge of the troposphere may be 13 km in altitude above England, but may be 12 km high above China. Both will have similar pressure levels that are typically around 25 kPa.

2.4 Neural Networks

ANNs are brain inspired systems which are intended to loosely replicate the way humans learn. ANNs consist of input and output layers, as well as one or more hidden layers containing neurons (nodes, units, or processing elements) that transform the input into something that the output layer can use. The strength of an ANN is obtained as the result of the connectivity and collective behavior of the neurons within the layers. This technique has been shown to have a high degree of accuracy when predicting weather forecasts in single location studies when modeling temperature and wind speed [17]. A study by Wang and Balaprakash [18] showed the ability for ANNs in forecasting weather variables in a single location and generalizing to a limited location around that area. Limited literature has focused on utilizing ANNs for forecasting weather conditions across the upper atmosphere using a single model

to generalize across the entire globe.

The mathematical goal of a ANN is to approximate some function f^*, where f^* can be a continuous function or a classifier. The architecture of a basic ANN is shown in Figure 1. The neuron is the basic building block of the network and exist in all the layers. The number of neurons in a layer is commonly referred to as the width of the layer. The width of the input layer reflects the number of features or predictor variables used to characterize an observation or function f. The output layer width represents how many outputs the model is attempting to approximate for function f^*. This could be multiple in the case of multivariate regression, or one if approximating a single continuous output function or classification problem [19].

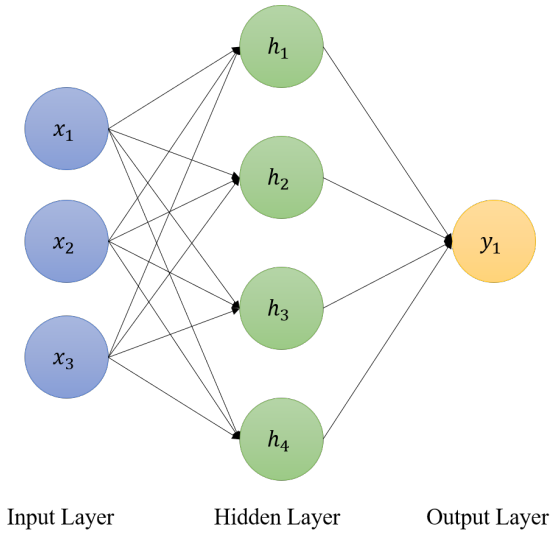

Figure 1. Basic ANN architecture

When an input vector is passed into the network, that vector passes through each node in the hidden layer, multiplied by that neuron's weight, and has some bias added to it. It is then acted upon by a chosen activation function. Activation functions are mathematical equations that determine the output of a neuron. It is simulating if a neuron should "fire" or not based on whether the input is relevant for the model's prediction. For example a sigmoid function will output a value between 0 and 1

while a hyperbolic tangent (tanh) function will output a value between -1 and 1 [19]. Equation 1 displays the math for computing the node output.

$$a_j = h(\sum_i w_{ij} x_i + b_j) \tag{1}$$

Where a_j represents the output from the j^{th} node, x_i is the i^{th} element of input vector x, w_{ij} is the weight for the i^{th} element of input vector x going to the j^{th} node, b_j is the bias on the j^{th} node, and h is the activation function. The function $f^*(y)$ is then calculated by summing all the neuron outputs going into the output layer. This is represented in equation 2.

$$f^*(y) = \sum_j w_{jk} a_j + b_j \tag{2}$$

Where w_{jk} is the weight for the j^{th} node to the k^{th} output. The goal of the network's learning process is to find the best weights to provide a mathematical model that best approximates a y for some given input x. To do this, a method called back propagation is employed. After the forward pass of information is sent through the network, a loss function C determines the error in the estimate. This error is then sent backwards through the network utilizing the gradient descent method in order to readjust the weights based on the loss. This is the main concept of back propagation. An observation is feed back into the network again with the readjusted weights, and the loss is sent backwards to adjust the weights again. This process iterates some number of times depending on the design of the network to meet convergence of some minimal error.

2.5 Long-Short Term Memory

Long-Short Term Memory (LSTM) networks are a special kind of Recurrent Neural Network (RNN). RNNs were developed by the need to use information from past observations in order to inform decisions made on the current observation. Traditional ANNs are not equipped for this kind of sequential analysis. RNNs address this by including a loop in the network allowing parameter sharing among observations. For example, separate parameters for every input feature are needed in a feed forward ANN. In a RNN, the parameter weights are shared across several time steps. This makes each member of the output a function of the previous members of the output. Also, the same update rule applied to the previous outputs is used to produce each member of the current output [19].

An issue with regular RNNs is with long-term dependencies. The benefit of RNNs are their ability to connect previous information to the current observation. The issue of long-term dependencies occurs when the gap between the previous information and current observation becomes very large [19]. In theory this should not be an issue, but in practice this is not the case. In a study done by Bengio, Simard, and Frasconi [20] it was shown that RNNs have an issue with long-term dependencies due to gradient based learning algorithms experiencing difficulty as the length of dependencies captured increases.

LSTMs help rectify this problem with long-term dependencies. Hochreiter and Schmidhuber [21] showed that by adding gates within the cell state to truncate the gradient, LSTMs can bridge the gap on information far in the past. Figure 2 shows the difference between unrolled repeating modules that exist within a standard RNN and a LSTM. While the RNN uses a single tanh layer to control the parameter sharing of past observations, an LSTM has four layers interacting with each other.

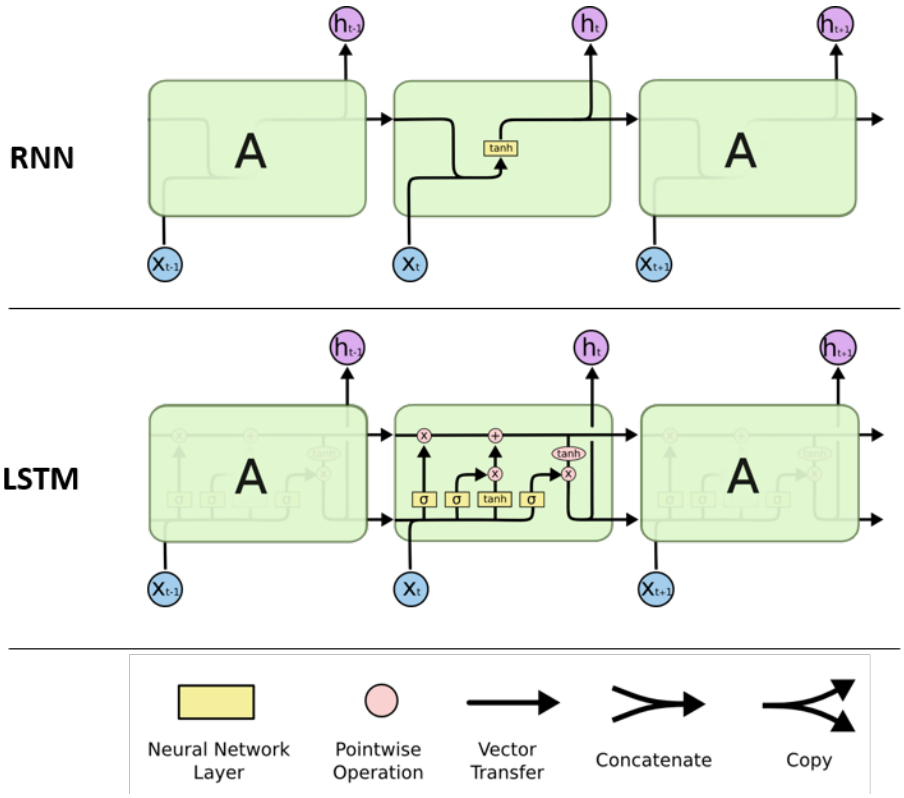

Figure 2. Comparison of RNN Structure to LSTM [1]

These gates allow the LSTM to do many things to configuring the cell state. The first step involves deciding what information needs to be removed followed by what new information needs to be kept. These layers are thought of as the "forget gate" and "input gate". The last step involves updating the cell state and producing an output based on the information [1]. A number of variants for LSTMs exist but for this study, a standard LSTM was investigated. Future work can explore the potential of advanced variants.

III. Methodology

3.1 Overview

This section explores in detail the methodology used within this study. The first section details the preprocessing done with the data, as this is an essential step in the process of a machine learning pipeline. Following that is an exploratory data analysis and architecture selection for the LSTM model. Finally, the different model types are discussed along with the performance metrics used to identify the best performing design.

3.2 Data Preprocessing

Weather data for this study was obtained from NOAA's data archives developed from the deterministic GFS model. The set used ranges from 20 July 2017 to 18 December 2017, with readings taken every six hours. From these readings, deterministic forecast are made from +6 hours to +168 hours out. Each reading contains weather data for each integer latitude and longitude intersection with latitudes ranging from $-90°$ to $90°$, and longitudes ranging from $-180°$ to $180°$. Additionally each coordinate has data for 31 different pressure levels within the atmosphere, with a range from 100 Pa to 100,000 Pa. This is approximately equivalent to an altitude measurement between 80 m to 41,700 m. This provides 2,008,800 unique coordinates across the entire globe, with each coordinate having 603 observations within the time range. A downside of this data set is that each coordinate has a small sample size. When considering the amount of data needed to sufficiently model the underlying non linear function using an LSTM, typically more would be appropriate for each coordinate [19].

The original data was contained in a GRIB2 file format which is not readily

readable on a Windows operating system. To mitigate this, MATLAB has a free tool called the "nctoolbox" which converts GRIB2 files to NetCDF. This new file format is readable by Windows, and can be manipulated in MATLAB to extract the variables of interest for any specific coordinate. For each coordinate and date, the temperature and wind components were extracted and organized into a file containing the time series data for each coordinate. Appendix A provides the code used in MATLAB to pull the data and structure it into a time series for every coordinate.

3.3 Parameter Tuning and Data Selection

As part of the design process, many parameters are tuned and optimized before the model designs are tested. With over 2 billion coordinates to select for training the model, a decision was made on selecting the optimal coordinates that best generalize to a majority of the others. The following subsections details the methodology used for selecting each of these along with other parameters that were tuned. These included feature selection, hyper-parameters of the LSTM architecture, and the sequence length of each observation.

3.3.1 Coordinate Selection

Due to time constraints, all 2 billion coordinates could not be tested and validated against each other. Instead a random sampling of coordinates was taken to find a suitable point for training the neural network. A suitable point allows the neural network to learn the underlying physics governing the weather dynamics, thereby enabling the model to generalize to other coordinates across all spatial dimensions of the globe.

The sampling of coordinates ensured both the northern and southern hemispheres were sampled, along with the eastern and western hemispheres. When sampling in

the spatial dimension of altitude, not every pressure layer was considered. Since typical cruising altitude for most aircraft exists between 9 - 14 km, only pressure layers from 30 - 15 kPa were considered. This was extended higher to 0.5 kPa (35 km) after investigation of multiple points. The higher altitudes generally showed smoother functions, which could be beneficial in the model learning the underlying dynamics. It was hypothesized that this could lead to a robust model. Therefore those pressure layers became part of the exploratory data analysis and parameter tuning. Figure 3 shows one example of this difference in smoothness for a random coordinate. The wind component at 7 km shows a highly chaotic nature compared to that same component at 26 km. The higher altitude chart displays a clearer pattern with less noise, albeit it still has a fair amount of chaotic nature to it.

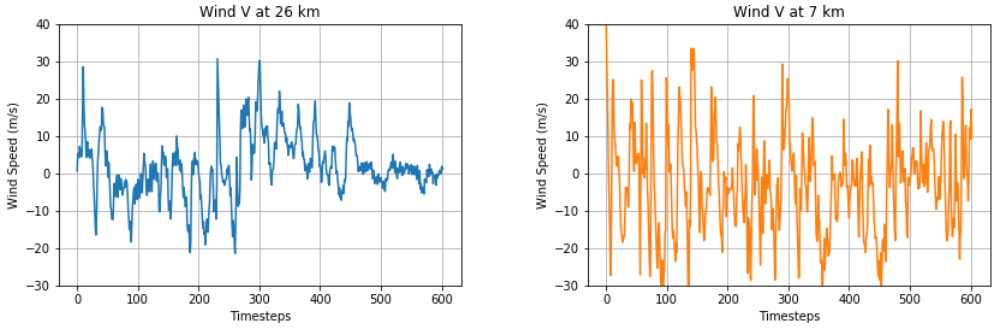

Figure 3. Wind V Component at Coordinate (40°, 110°)

Five latitudes from the northern and southern hemispheres each were randomly chosen along with seven longitudes each representing the eastern and western hemispheres. Seven pressure layers were randomly selected from the 11 layers defined earlier, between 30 kPa and 0.5 kPa. Producing combinations of these ranges gave 980 unique coordinates to test, sampled from all spatial dimensions of the globe. Since each coordinate was tested against all the others, this led to 980 models trained and validated and 959,420 total model evaluations conducted. This illustrates why every coordinate could not be tested in the time allocated for this study, as it would

result in approximately 2 billion models, and over 4 trillion tests.

A basic LSTM network with one layer, 256 nodes and a tanh activation function was built to test the coordinates for the one that could best generalize globally. The network was tested on a couple of coordinates to ensure it could converge to a solution, and had enough capacity to learn. Each coordinate was then trained and validated against a randomly sampled coordinate with the mean squared error (MSE) used as the loss function. Each trained model was then tested against the other 979 sampled coordinates, with RMSE for the two predicted outputs of the wind-u component, u, and wind-v component, v, being recorded. An ideal set of coordinates would have a low RMSE for both outputs.

Each set of 979 test results from each coordinate was averaged to give each of the 980 coordinates an average RMSE for the two outputs u and v. This distribution of the 980 average RMSEs was examined for each output in order to determine a proper range to pull the best performing points from. If non-normality was discovered in the distribution then the median was used since it is a more robust statistic.

A weighting scheme was employed to ensure the ideal coordinates minimized the total RMSE from u and v, and minimized the difference in RMSE between the two. Both of these objectives were equally weighted and added together to produce a final score for each coordinate, as shown in equation 3.

$$W = 0.5(RMSE^s_{total}) + 0.5(RMSE^s_{difference}) \tag{3}$$

Where $RMSE^s_{total}$ is the total RMSE between u and v for a coordinate, and normalized against the set of totals. Likewise, $RMSE^s_{difference}$ is the absolute difference in RMSE between u and v for a coordinate, and normalized against the set of differences. Finally, W is the final score for each coordinate.

A set of ideal coordinates, C_i, was built from those that scored low in all three

outputs as opposed to just one output. The overall best performing point, C_0, was then used for training the network for parameter tuning.

3.3.2 Sequence Length

The sequence length defines how many time steps in the past to use for the current observation. Every observation fed into an LSTM is a sequence, with two dimensions. One defining the number of features and the second defining the number of time steps in the past, or sequence length. For example, a sequence of length three includes the observation's u and v values, along with the values for two time steps before shown in Figure 4. The risk in this when dealing with a small data-set pertains to the reduction of observations available for the model as the sequence length increases. For example, every +1 increase in sequence length, S, yields -1 observations, n, from the data set. Stated mathematically, $n_t - S = n_f$, where n_T is the total observations in the original data set, n_F is the final number of observations, and S is the length of the sequence.

To test for the optimal sequence length, S_0, the same basic LSTM architecture was used. Training and validation was conducted on C_0 and a randomly selected point from C_i with sequence lengths ranging from 1-15 time steps. Additionally, this parameter was trained for predicting all four time steps of +18, +36, +54, and +90 hours as opposed to just 18 hours. Each instance of the model was then run 20 times to develop an average RMSE. This yielded 1200 models trained and validated with the average validation RMSE for each output used as the performance metric for determining S.

Statistical analysis was done on the results using box plots and Tukey's T-test to determine differences that existed within the groups and determining the best performing sequence length value for each prediction range.

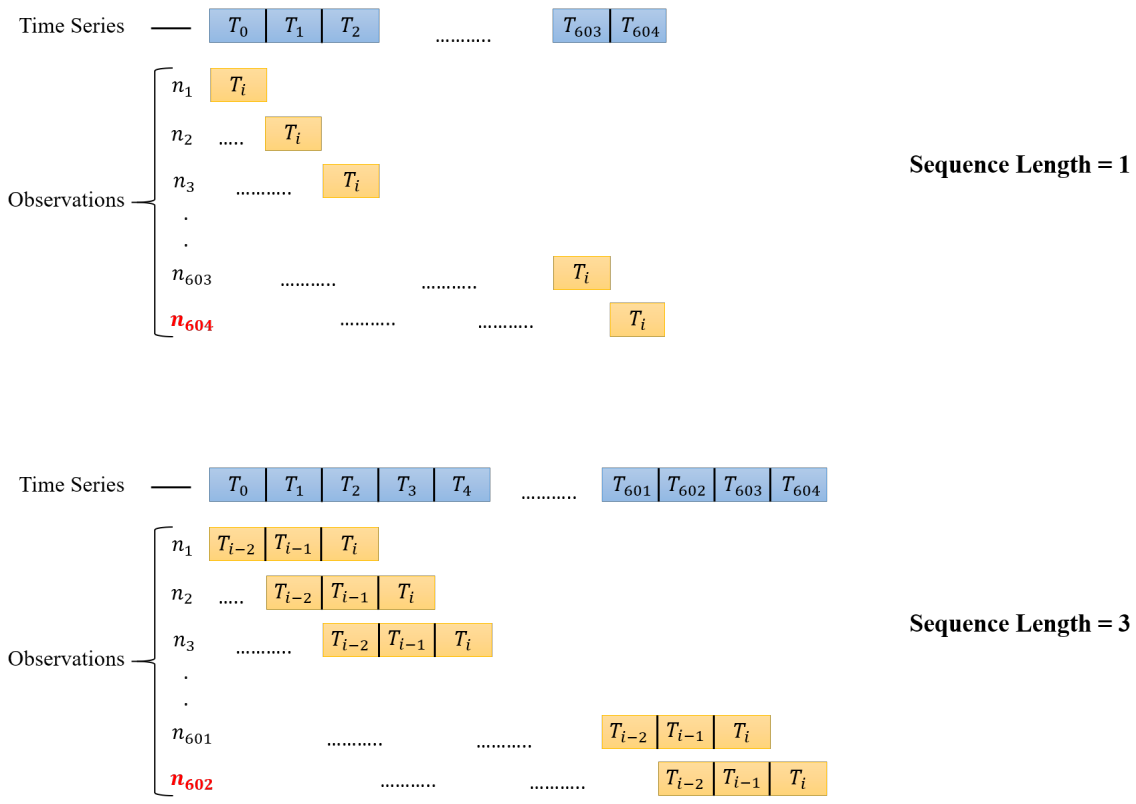

Figure 4. Sequence Length

3.3.3 Feature Selection

There were six different sets of features investigated in this study. The first set, which is feature set 1, included only the base features of temperature and wind components (u, v). This set is small, representing the bare minimum needed to run the model. The rest of the sets are compared against this base set to determine whether the added features increased prediction accuracy.

Feature set 2 consisted of the base features along with the base features of neighboring coordinates. This is called the neighbor set, as it uses the base features of the neighbors as a means to help predict wind speed for the primary coordinate. Six total neighboring coordinates were used. Each was taken from a +1 increase and -1 decrease to the initial coordinate's latitude, longitude, and pressure layer. Their base

features were then added to the data set for C_0, increasing the input dimensions to 21 features. Figure 5 shows a representation of the location of the points.

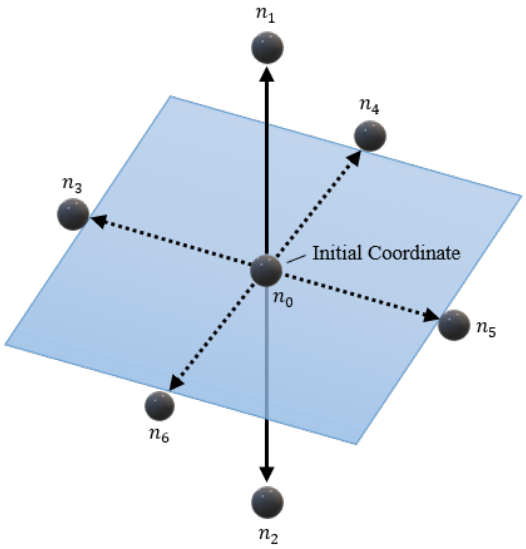

Figure 5. Neighbor Coordinates for Model

Feature set 3, 4, 5 and 6 are all very similar as they used different variations of the deterministic forecast to aid in predicting wind speeds. Feature set 3 used the base features from the +18 hour forecast, while feature set 4 used the base features from the +18 hour and +36 hour forecast. This pattern followed for the last two feature sets. Feature set 5 included base features from the +18, +36, and +54 hour forecasts, while feature set 6 included base features from the +18, +36, +54, and +90 hour forecasts. Using the deterministic forecast as a feature can assist the network in learning the underlying behavior in the data, and correcting some of the error in the forecast to adjust for a more accurate result. Table 1 summarizes the different feature sets examined.

Table 1. Feature Sets

Feature Set	Features
1	Base
2	Neighbors
3	Forecast +18hr
4	Forecast +18hr, +36hr
5	Forecast +18hr, +36hr, +54hr
6	Forecast +18hr, +36hr, +54hr, +90hr

Including the base feature set, there were six sets to test with each set trained and validated on C_0 and a randomly selected point from C_i not previously used. Each set was trained 20 times for each of the four prediction periods of $T = (18, 36, 54, 90)$ hours. This yielded 480 models trained and validated with the average validation RMSE for u and v being used as the performance metric for determining the best performing feature set. Statistical analysis involved using box plots and Tukey's T-test to determine differences that existed within the feature sets and determining the overall best performing set.

3.3.4 LSTM Architecture Parameters

The final step in parameter tuning is to fine tune the LSTM architecture utilizing all the results from the previous sections and the final training data set. Parameters that were tuned for in the LSTM architecture were the number of layers, layer width, optimizer, learning rate, activation function, and early stopping criteria. While many "deep learning" networks are considered deep because they extend past one hidden layer, and in truth have a multitude of hidden layers, LSTM networks do not necessarily need as many layers. Generally one layer is sufficient with two being utilized for more complex problems. Extending beyond two is generally done for image, video, and text prediction problems that have more underlying features to describe, and may be paired with convolution layers instead. Table 2 shows all the options examined for

these parameter values.

Table 2. LSTM Parameter Options

Parameter Options					
Layers	Layer Width	Activation Function	Optimizer	Learning Rate	Patience
1	64	Relu	SGD	0.001	25
2	128	Tanh	Adam	0.01	50
	256	Sigmoid	RMSprop	0.1	75
	512				100

In total 864 different designs were trained and validated using the combinations created from Table 2. Due to the large number of architectures explored, total validation MSE was used as the performance metric. This is not dissimilar to the previous sections as MSE is already used as the loss function, then converted to RMSE for each output. Here, time is a constraint preventing the examination of all these architectures if examined by RMSE for each output, which is why total validation MSE is being used instead. Since a difference in network architecture was examined, running multiple instances of each architecture design to achieve an average validation RMSE was not necessary. Model initialization kept the same random number generation in order to keep the initial weights of the network the same. This allowed observing the effect of changing a network parameter and the resulting change in prediction accuracy.

All the results were examined using an iterative elimination process to hone in on the optimal architecture. Parameter values that produced large errors were removed until only well performing combinations remained. Then the parameters were chosen from the remaining designs with the best validation MSE.

3.4 Model Types

Two model types were examined in this study, with both utilizing the same LSTM architecture designed in the previous section. The first is the classic LSTM network.

The second model is an LSTM ensemble network, composed of six of the classic LSTM networks. The goal of the ensemble network is to investigate the ability to combine the predictive powers of multiple models to achieve greater accuracy and generalization.

Each member of the ensemble network was trained using one of the top ten coordinates identified during coordinate selection, but not C_0. Validation was done using a randomly selected unused coordinate from C_i. Each trained model then acted as an input into a one layer basic neural network that acted as a meta-learner. The meta-learner was trained on C_0 to find the proper weights for combining the predictive scores from the input models.

The trained models were tested against 4,000 randomly sampled coordinates. Results were recorded for each prediction period, and statistical analysis involved box plots and a T-test to determine differences that existed within the two models. This determined the best performing model type, which was then used for final testing.

3.5 Deterministic Forecast Comparison

Comparison with the deterministic forecast is the final measure of performance conducted. The deterministic forecast represents the current technique utilized at AMC for mission planning. Performance of the constructed LSTM model against the deterministic model serves as a benchmark for determining the model's ability to predict future weather factors, or acting as a post processor for reducing error in the deterministic forecast. If predictions are comparable or better, then it is assumed that the model is beneficial in existing fuel optimization scenarios for reducing fuel usage.

The test was conducted by taking the best performing LSTM model design and the deterministic forecast, and evaluating both against 10,000 randomly sampled

coordinates for prediction periods $T = (18, 36, 54, 90)$ hours. The RMSE for u and v for each test was utilized in statistical analysis to determine the model's overall performance to current industry methods. Statistical analysis involved box plots and a T-test to determine differences that existed within the two models, and determining the performance of the LSTM compared to the deterministic forecast.

3.6 Flight Path Comparison

An indirect estimate of fuel efficiency can be taken from the travel time between two locations. To do this a shortest path optimization was conducted to find the optimal altitudes to fly at to maximize tailwinds. Maximizing tailwinds is one indirect way to gauge fuel efficiency, since positive tailwinds imply negative headwinds. With negative headwinds the plane receives more force from the back as opposed to the front, generally requiring less fuel to propel the plane forward.

To evaluate this, a network of paths was built for a flight between McGuire AFB to Ramstein AFB. The flight path was divided into 20 equally spaced way-points, with each way-point having five altitude levels to choose. The minimum and maximum for these altitudes was based on average cruising altitudes between 8 to 13.5 km. Table 3 shows the altitude levels, and Table 4 displays the coordinates for the 20 way-points.

Table 3. Flight Path Altitude Levels

Pressure (Pa)	Altitude (km)
35,000	8,078
30,000	9,126
25,000	10,327
20,000	11,758
15,000	13,562

Table 4. Flight Path Way-points

Waypoint	Coordinate
0	(40.04, -74.58)
1	(40.51, -70.47)
2	(40.98, -66.36)
3	(41.45, -62.25)
4	(41.92, -58.14)
5	(42.39, -54.04)
6	(42.86, -49.93)
7	(43.33, -45.82)
8	(43.8, -41.71)
9	(44.27, -37.6)
10	(44.74, -33.49)
11	(45.21, -29.38)
12	(45.68, -25.27)
13	(46.15, -21.16)
14	(46.62, -17.05)
15	(47.09, -12.95)
16	(47.56, -8.84)
17	(48.03, -4.73)
18	(48.5, -0.62)
19	(48.97, 3.49)
20	(49.44, 7.6)

A couple assumptions were made in order to populate the network with wind values given the dynamic nature of the network. First involves the cruising speed of the aircraft. This was set to a constant 450 knots or 231.5 m/s with no headwind or tailwind present. The second involves take-off and landing, which were ignored given the length of the travel between the two locations. The distance between way-points is sufficiently long enough for the aircraft to reach the desired altitude level without considering the time difference imposed.

Each way-point was populated with the predicted wind speeds for the time elapsed from traveling. Wind speed predictions came from the +36, +54, and +90 hour LSTM prediction models. This model was the same as the final one used in the deterministic comparison, except feature set 1 is used as opposed to any others. Limiting the feature set was done to highlight the computational speed and accuracy

trade-offs. If deterministic forecast features were included, then running the LSTM model would be reliant on having that forecast available. Leaving the features out allows the LSTM model to run independent of the deterministic model, and actually allow a computational speed comparison between the two.

Since weather readings occur every six hours, this implies that wind speed values for a set of way-points will become inaccurate as the plane moves through the network and time elapses. Therefore, a linear interpolation was conducted between the time steps in order to find the wind speeds at the elapsed time. Once the time values were known, linear interpolation was used again in the spatial dimension to find the wind speed values at the exact way-point's coordinate. This was done for wind speed values from the LSTM model, deterministic model, and the truth values from the GFS.

Performance was measured by looking at the difference in optimal paths compared to the GFS model, along with the estimated total travel time. Statistical measures were used for to analyze the time difference between the LSTM flight path travel compared to the GFS and deterministic outputs.

IV. Results

4.1 Overview

This section details the results from the testing phase along with results from the data pre-processing and parameter tuning. Since the significance of a machine learning model depends on the manipulation of the data-set and tuning of the model parameters, these results are integral to understanding the final model.

4.2 Parameter Tuning Results

There were many parameters that were investigated and tuned to design the optimal model for the given data-set. This section reviews the results from coordinate selection, sequence length selection, feature selection, and the final architecture design.

4.2.1 Coordinate Selection

Sampling for the optimal coordinates with the ability to best generalize required a multitude of models to run and evaluate. With a sample size of 980 coordinates, this required training 980 models along with 959,420 model evaluations.

Figure 6 shows the error distribution for u and v from the completed model evaluations along with their associated normal probability plots. The histograms indicate a distribution for the error that may be non-normal. Investigating the normal probability plots more clearly displays the non-normality of the error, with majority of the points falling out of the CI bands for the fitted line. This non-normality can be due to the non-negative nature of the error. In this instance zero acts as a lower bound and constraint on the distribution, giving this distribution more of a log-normal look.

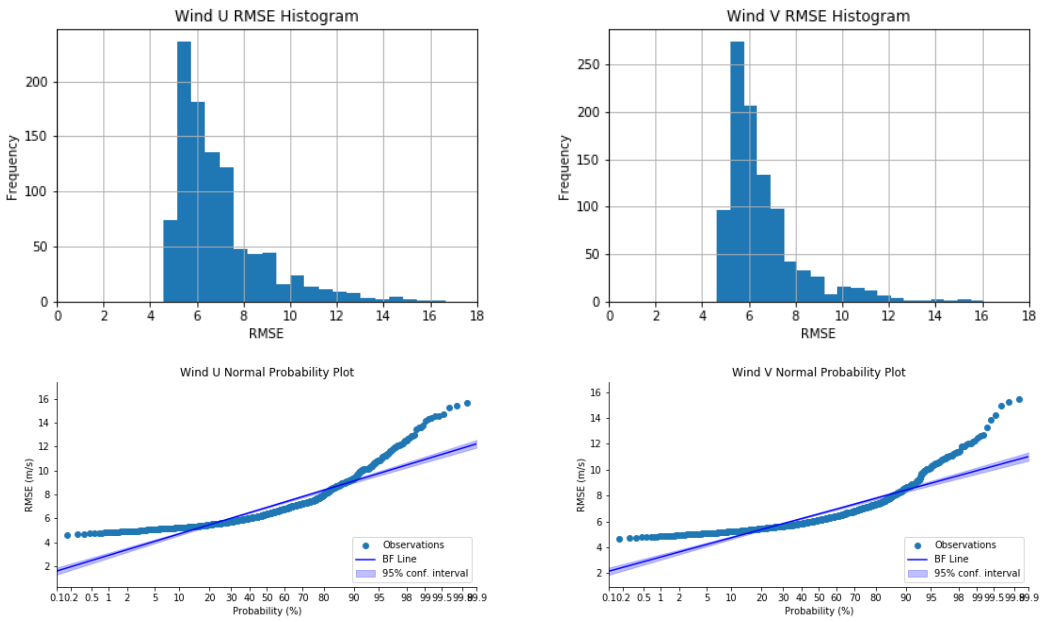

Figure 6. Distribution of RMSE for Coordinate Selection

Due to this non-normality, the median and quantile ranges of the data were used to identify the best performing coordinates since the median is a more robust statistic than the mean, and is less effected by outliers. Using the 5^{th} percentile as a cutoff for the weighted scoring scheme implemented, the upper limits for u and v RMSE respectively were 5.088 m/s and 5.075 m/s with the minimum values being 4.553 m/s and 4.647 m/s. Table 5 shows the final selection of coordinates in rank from their weighted score on performance. The top ranked coordinate was $(-78°, 68°)$, now referred to as C_0.

Table 5. Final Coordinate Selection and RMSE

Coordinate	Altitude (km)	Wind U RMSE	Wind V RMSE	Weighted Score	Ranking
(-78°, 68°)	20	4.810	4.816	**0.481**	**1**
(-77°, 68°)	18	4.863	4.826	**0.578**	**2**
(-77°, -68°)	16	4.863	4.882	**0.579**	**3**
(-78°, 93°)	16	4.893	4.866	**0.635**	**4**
(-78°, 7°)	26	4.903	4.754	**0.653**	**5**
(-78°, -78°)	20	4.952	4.931	**0.745**	**6**
(-78°, -4°)	26	4.958	4.985	**0.791**	**7**
(-78°, 68°)	31	4.949	4.994	**0.810**	**8**
(-78°, 113°)	16	4.953	5.039	**0.916**	**9**
(-78°, 93°)	18	5.046	4.829	**0.922**	**10**
(-78°, 139°)	31	5.072	4.925	**0.969**	**11**
(-78°, -4°)	18	4.907	5.065	**0.977**	**12**
(-78°, 7°)	18	5.076	4.993	**0.977**	**13**

The location of the top performing coordinates are of interest. They mostly all fall along the same latitude, but spread among various longitudes and altitudes. Figure 7 is a visual representation of the coordinate random sampling along with where the final chosen coordinates were located. Interestingly, the region representing the final points is close to the same region as the southern polar jet stream. This indicates a possible relationship exist here with this particular jet stream, and the rest of the global coordinates. The regions containing the other three jet streams did not pop up in the top scoring sampled coordinates suggesting a future area of investigation.

This insight might imply a bias within point selection but reviewing the sampling of points reveals that regions covering all four jet streams were part of the sampling set. Their non-inclusion into the top coordinates indicates that if a bias exist, it is not due to non-sampling of the regions.

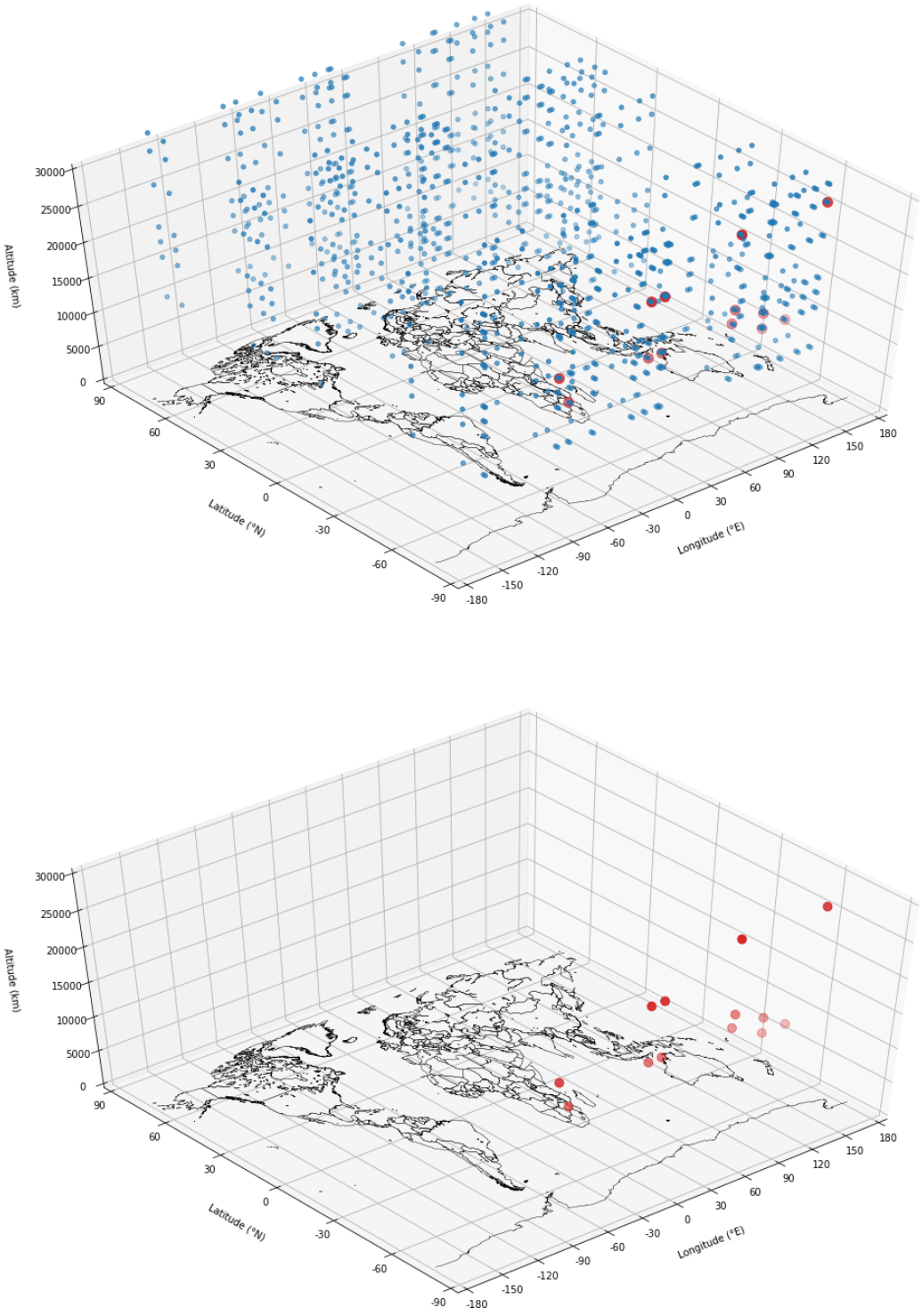

Figure 7. Initial and Final Coordinate Selection

4.2.2 Sequence Length

The following sections examine prediction periods +18, +36, +54, and +90 hours. The optimal sequence length could depend on the prediction range, with different lengths being appropriate for each range.

4.2.2.1 18 hours

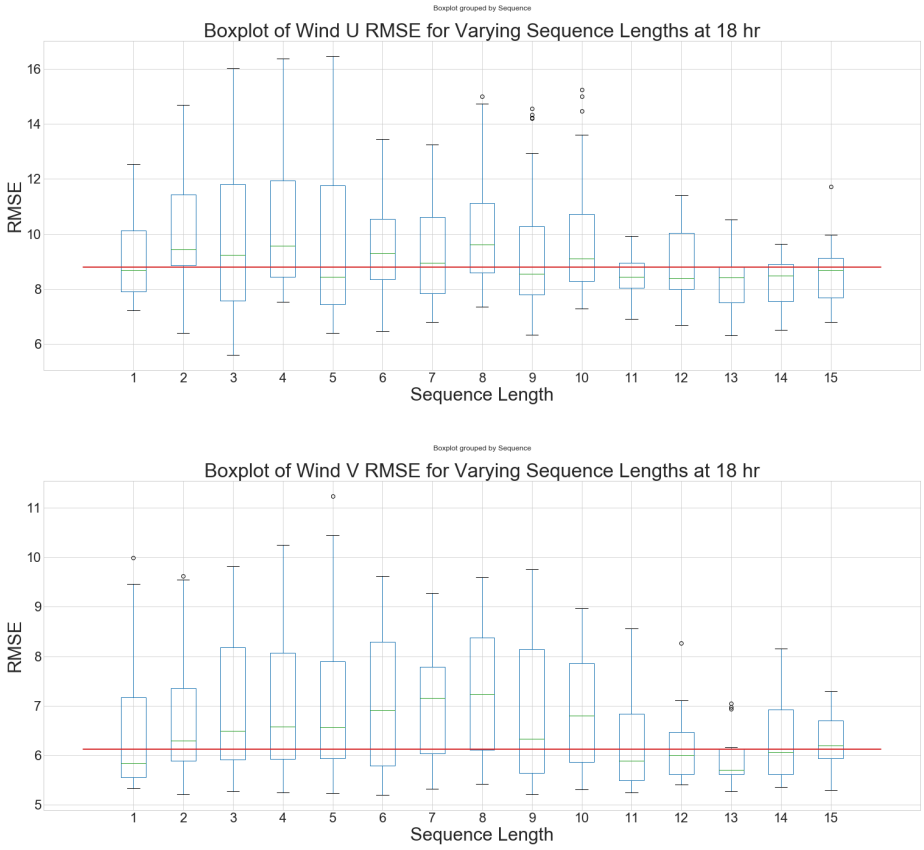

Figure 8. Sequence Length Box Plots, 18hr Results

Figure 8 above shows the box plots for u and v, for the +18 hour prediction range, $T = 18$ hours. The red line indicates the 75^{th} percentile for the value with the lowest median, which for both outputs is $S = 13$. Using this threshold certain length values can be removed from consideration by observing if their 25^{th} percentile does

not overlap with the red line. These groups are considered statistically different by observation. For u, only $S = 2$ is eliminated, while none are eliminated from v. For both outputs, the longer sequence length values generally had smaller spreads of data indicating smaller variance, while the lower ones had a much higher variance. Low variance is a highly desired trait here that comes into consideration for final sequence length selection.

The next step was comparing the CI on the difference of means for the groups to $S = 13$ using Tukey's test. Table 6 displays the results for u with $S = 13$ and compared to the rest of the values.

Table 6. Sequence Tukey Test Results, Wind U 18 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	13	-0.9541	0.9	-3.5564	1.6482	False
2	13	-1.7721	0.558	-4.3744	0.8302	False
3	13	-1.6046	0.6976	-4.2069	0.9977	False
4	13	-2.4014	0.1068	-5.0037	0.2009	False
5	13	-1.374	0.8899	-3.9763	1.2283	False
6	13	-1.3706	0.8927	-3.9729	1.2317	False
7	13	-1.2057	0.9	-3.808	1.3966	False
8	13	-1.9127	0.4367	-4.515	0.6896	False
9	13	-1.2223	0.9	-3.8246	1.3799	False
10	13	-1.8257	0.5133	-4.428	0.7766	False
11	13	-0.2183	0.9	-3.0003	2.5637	False
12	13	-0.5486	0.9	-3.3305	2.2334	False
13	14	-0.0782	0.9	-2.8602	2.7038	False
13	15	0.3416	0.9	-2.4404	3.1236	False

When compared against $S = 13$, none of the other values showed a statistically significant difference, indicating a statistical difference doesn't exist between the means given the sample size. With only 20 samples per group, the limited sample size could be a hindrance. Observing the mean difference being smaller for the higher valued sequence lengths illustrates the smaller variance that existed in those groups, and their similarity to each other compared to the lower values. Test results for v with $S = 13$ are shown in Table 7.

Table 7. Sequence Tukey Test Results, Wind V 18 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	13	-0.6656	0.9	-2.2022	0.8711	False
2	13	-0.8345	0.8572	-2.3711	0.7021	False
3	13	-1.087	0.5007	-2.6236	0.4497	False
4	13	-1.2232	0.2959	-2.7598	0.3134	False
5	13	-1.2463	0.2661	-2.7829	0.2903	False
6	13	-1.1146	0.4598	-2.6512	0.422	False
7	13	-1.1628	0.3837	-2.6995	0.3738	False
8	13	-1.2863	0.2196	-2.8229	0.2504	False
9	13	-0.9359	0.714	-2.4725	0.6007	False
10	13	-0.8868	0.7833	-2.4235	0.6498	False
11	13	-0.4293	0.9	-2.072	1.2134	False
12	13	-0.2044	0.9	-1.8471	1.4384	False
13	14	0.3732	0.9	-1.2695	2.0159	False
13	15	0.3638	0.9	-1.2789	2.0065	False

Similar results are observed for v, where none of the means of the groups compared against $S = 13$ are statistically different. Again, this could be due to the smaller sample size collected. Using the evidence of the smaller variances and lowest median values for RMSE, $S = 13$ was chosen as the sequence length parameter for $T = 18$ hours.

4.2.2.2 36 hours

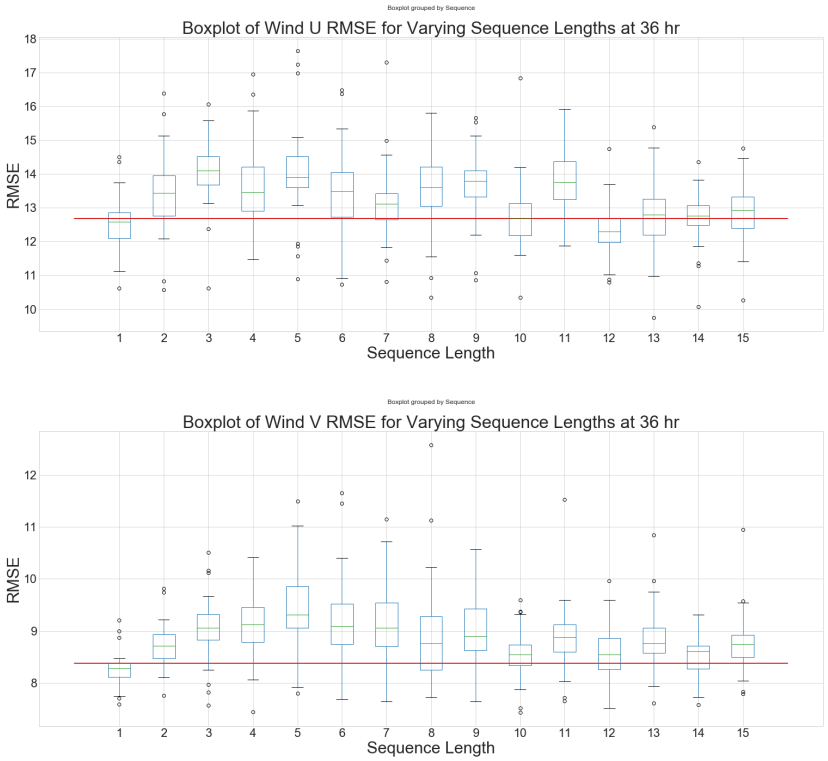

Figure 9. Sequence Length Box Plots, 36hr Results

For $T = 36$ hours, Figure 9 shows the box plots for u and v. The 75^{th} percentile here falls on $S = 12$ for u, and $S = 1$ for v. This is a surprising observation for v, while u seems to fall in line with the previous sections. It still holds that the larger S values tend to have lower median RMSE values along with smaller variances, with $S = 1$ being an exception. Also, both charts in Figure 9 show a similar pattern from $S = 1$ to $S = 15$ where the middle values tend to increase in error then drop back down. It is an interesting phenomenon that warrants further investigation.

Using the red line as a threshold, $S = (2, 3, 4, 5, 6, 8, 9, 11)$ and $S = (2, 3, 4, 5, 6, 7, 9, 11, 13, 15)$ were removed for u and v respectively. A few discrepancies exist among the two groups but they are overall very similar. The further out in prediction range, the declining

accuracy might have an effect on distinguishing between the two wind components more distinctly, and bringing out any nuanced differences that are starting to appear here between u and v.

The next step was comparing the CI on the difference of means for the groups to the lowest value using Tukey's test. Table 8 displays the results for u with $S = 12$ and compared to the rest of the values.

Table 8. Sequence Tukey Test Results, Wind U 36 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	12	-0.1898	0.9	-1.8654	1.4858	False
2	12	-1.126	0.5774	-2.8016	0.5496	False
3	12	-1.7468	0.0317	-3.4224	-0.0712	True
4	12	-1.302	0.3357	-2.9776	0.3736	False
5	12	-1.7814	0.0251	-3.4571	-0.1058	True
6	12	-1.1099	0.5983	-2.7855	0.5657	False
7	12	-0.8886	0.8848	-2.5643	0.787	False
8	12	-1.2826	0.3627	-2.9582	0.393	False
9	12	-1.3125	0.3219	-2.9881	0.3631	False
10	12	-0.419	0.9	-2.0946	1.2566	False
11	12	-1.507	0.2126	-3.2983	0.2843	False
12	13	0.4589	0.9	-1.3324	2.2502	False
12	14	0.4219	0.9	-1.3694	2.2132	False
12	15	0.5937	0.9	-1.1976	2.385	False

Only two groups are statistically significant in their difference, $S = (3, 5)$. This is also apparent from looking at the box plots. While many of the others did not exhibit a statistical difference in their means, it is shown that the higher value groups mean difference was smaller than the lower value groups. This again gives more evidence that the upper groups where $S > 10$ are more similar than the others (excluding $S = 1$ here) when comparing means. Similar insight comes when looking at v. Test results for v with $S = 1$ are shown in Table 9.

Table 9. Sequence Tukey Test Results, Wind V 36 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
2	1	0.4914	0.8924	-0.4413	1.4241	False
3	1	0.7722	0.2354	-0.1605	1.7049	False
4	1	0.8277	0.1457	-0.105	1.7605	False
5	1	1.2028	0.0014	0.2701	2.1355	True
6	1	0.8846	0.0843	-0.0481	1.8174	False
7	1	0.8771	0.0902	-0.0556	1.8099	False
8	1	0.6641	0.4905	-0.2687	1.5968	False
9	1	0.7411	0.2988	-0.1916	1.6739	False
10	1	0.2898	0.9	-0.643	1.2225	False
11	1	0.6231	0.6936	-0.3844	1.6305	False
12	1	0.307	0.9	-0.7005	1.3144	False
13	1	0.6119	0.7178	-0.3956	1.6193	False
14	1	0.2405	0.9	-0.767	1.248	False
15	1	0.5099	0.9	-0.4975	1.5174	False

Another way to examine these results is to look at their p-values. While all, except a couple groups, are not statistically different, the highest p-values for commonality exist among $S = 1$ and $S > 10$. This is true for both u and v, indicating any choice within that set could be reasonable as a parameter. Since $S = 1$ still seems like an oddity, $S = 12$ is being used for $T = 36$ hours since it scored best for u and second best for v. It also more closely follows the pattern being set for this parameter.

4.2.2.3 54 hours

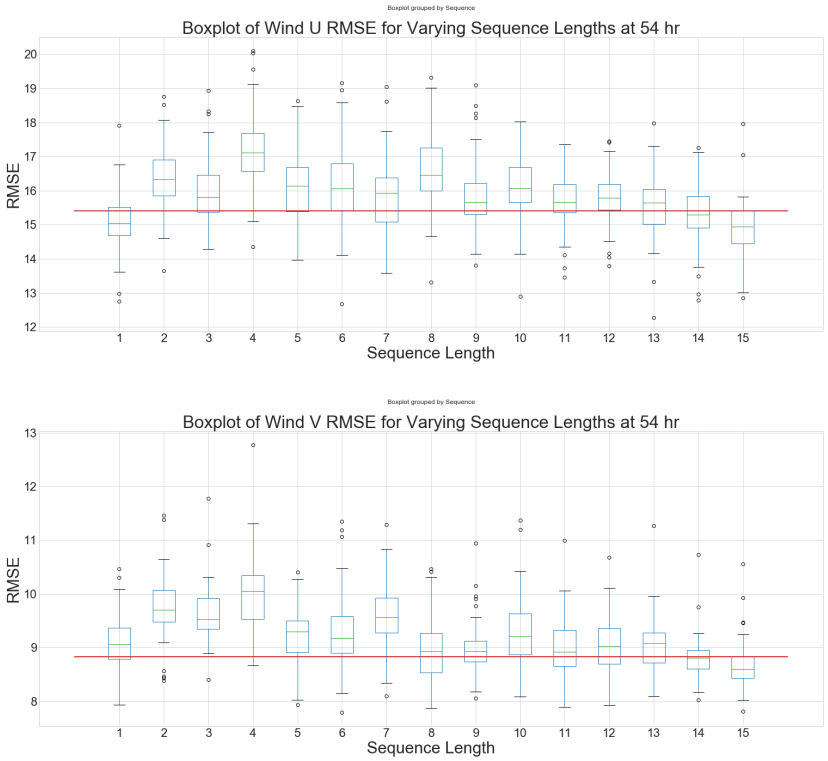

Figure 10. Sequence Length Box Plots, 54hr Results

For $T = 54$ hours, Figure 10 above shows the box plots for u and v. The 75^{th} percentile here falls on $S = 15$ for u, and $S = 15$ for v. Again a similar pattern exist with the groups, where $S = 1$ is low and the error generally rises before coming back down at $S > 10$. Little differences exist in these charts that was not already explored from the previous time ranges. Groups that were removed from consideration based on the red line were $S = (2, 4, 6, 8, 10)$ for u, and $S = (2, 3, 4, 5, 6, 7, 10)$ for v.

The next step was comparing the CI on the difference of means for the groups to the lowest value using Tukey's test. Table 10 displays the results for u with $S = 15$ compared to the rest of the values.

Table 10. Sequence Tukey Test Results, Wind U 54 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	15	-0.2371	0.9	-1.813	1.3388	False
2	15	-1.4827	0.0899	-3.0586	0.0932	False
3	15	-1.1563	0.441	-2.7321	0.4196	False
4	15	-2.2401	0.001	-3.8159	-0.6642	True
5	15	-1.1805	0.4034	-2.7564	0.3953	False
6	15	-1.2899	0.2527	-2.8658	0.286	False
7	15	-0.9345	0.7501	-2.5104	0.6414	False
8	15	-1.7493	0.0145	-3.3252	-0.1734	True
9	15	-0.986	0.6792	-2.5618	0.5899	False
10	15	-1.1101	0.5082	-2.686	0.4658	False
11	15	-0.8315	0.892	-2.4073	0.7444	False
12	15	-0.8094	0.9	-2.3853	0.7665	False
13	15	-0.6403	0.9	-2.2162	0.9355	False
14	15	-0.3703	0.9	-1.9462	1.2056	False

When compared against $S = 15$ only two values tested to be statistically different, $S = (4, 8)$. The same pattern exists where the higher values and $S = 1$ share higher p-values compared to the rest of the groups. Moving on to v this is also apparent again as shown in Table 11. Thus, $S = 15$ was chosen as the parameter value for both u and v.

Table 11. Sequence Tukey Test Results, Wind V 54 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	15	-0.3556	0.9	-1.2537	0.5424	False
2	15	-1.0374	0.0083	-1.9354	-0.1393	True
3	15	-0.9645	0.0222	-1.8626	-0.0665	True
4	15	-1.3917	0.001	-2.2897	-0.4936	True
5	15	-0.4889	0.8556	-1.3869	0.4092	False
6	15	-0.5746	0.6485	-1.4726	0.3235	False
7	15	-0.9038	0.0468	-1.8018	-0.0057	True
8	15	-0.2099	0.9	-1.1079	0.6882	False
9	15	-0.3666	0.9	-1.2646	0.5314	False
10	15	-0.6391	0.4923	-1.5372	0.2589	False
11	15	-0.3034	0.9	-1.2015	0.5946	False
12	15	-0.4012	0.9	-1.2993	0.4968	False
13	15	-0.3546	0.9	-1.2526	0.5435	False
14	15	-0.1484	0.9	-1.0464	0.7497	False

4.2.2.4 90 hours

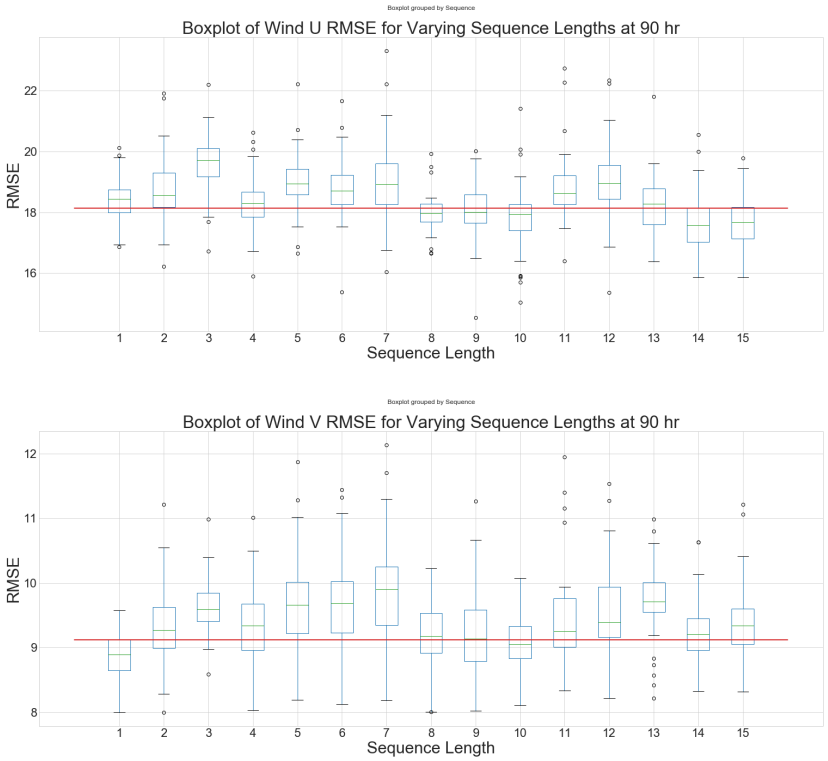

Figure 11. Sequence Length Box Plots, 90hr Results

For $T = 90$ hours, Figure 11 shows the box plots for u and v. The 75^{th} percentile here falls on $S = 14$ for u, and $S = 1$ for v. Again a similar pattern exist with the groups, where $S = 1$ is low and the error generally rises before coming back down at $S > 10$. This is much more pronounced for v since $S = 1$ again scores the lowest. Groups that were removed from consideration based on the red line were $S = (2, 3, 5, 6, 7, 11, 12)$ for u, and $S = (3, 5, 6, 7, 12, 13)$ for v. The major difference here is the removal of some of the high value sequence lengths. This is most likely due to the larger uncertainty pertaining to $T = 90$ hours requiring a larger sample size in the future.

The next step was comparing the CI on the difference of means for the groups to

the lowest value using Tukey's test. Table 12 displays the results for u with $S = 15$ compared to the rest of the values.

Table 12. Sequence Tukey Test Results, Wind U 90 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	14	-0.758	0.9	-2.2726	0.7566	False
2	14	-1.1051	0.4508	-2.6197	0.4095	False
3	14	-1.9035	0.0021	-3.4181	-0.389	True
4	14	-0.7455	0.9	-2.2601	0.7691	False
5	14	-1.403	0.1035	-2.9176	0.1116	False
6	14	-1.2238	0.2728	-2.7384	0.2908	False
7	14	-1.3849	0.1158	-2.8995	0.1297	False
8	14	-0.3283	0.9	-1.8428	1.1863	False
9	14	-0.5442	0.9	-2.0588	0.9704	False
10	14	-0.1526	0.9	-1.6672	1.3619	False
11	14	-1.2917	0.1945	-2.8062	0.2229	False
12	14	-1.4266	0.089	-2.9412	0.088	False
13	14	-0.5842	0.9	-2.0988	0.9304	False
15	14	0.0413	0.9	-1.4733	1.5559	False

When compared against $S = 15$ only one value tested to be statistically different, $S = (3)$. With this being the longest prediction range, it is expected that each group tested would have a higher variance. Therefore making it more difficult to discern more statistical differences among the groups. This is evident from the results for v shown in Table 13. Given that consistency has been with the higher values for u and v, the chosen sequence length for $T = 90$ hours was $S = 15$.

Table 13. Sequence Tukey Test Results, Wind V 90 hours

Sequence Group 1	Sequence Group 2	Mean Difference	P-Value	Lower	Upper	Reject
1	2	0.4515	0.9	-0.4722	1.3751	False
1	3	0.7317	0.3046	-0.192	1.6554	False
1	4	0.4492	0.9	-0.4745	1.3729	False
1	5	0.7791	0.2095	-0.1446	1.7028	False
1	6	0.8139	0.1542	-0.1097	1.7376	False
1	7	0.9969	0.0209	0.0732	1.9205	True
1	8	0.35	0.9	-0.5737	1.2737	False
1	9	0.3426	0.9	-0.581	1.2663	False
1	10	0.1885	0.9	-0.7352	1.1121	False
1	11	0.5939	0.6416	-0.3298	1.5176	False
1	12	0.6769	0.4431	-0.2468	1.6006	False
1	13	0.8001	0.175	-0.1236	1.7237	False
1	14	0.3355	0.9	-0.5882	1.2592	False
1	15	0.5085	0.8424	-0.4152	1.4321	False

4.2.3 Feature Selection

To examine the results from feature selection, the following sections discuss prediction periods +18, +36, +54, and +90 hours. This helps to determine if any top performing feature sets faltered in any particular prediction period, as opposed to performing high in all of them. The feature set chosen is the one that performs high in all four prediction periods.

4.2.3.1 18 hours

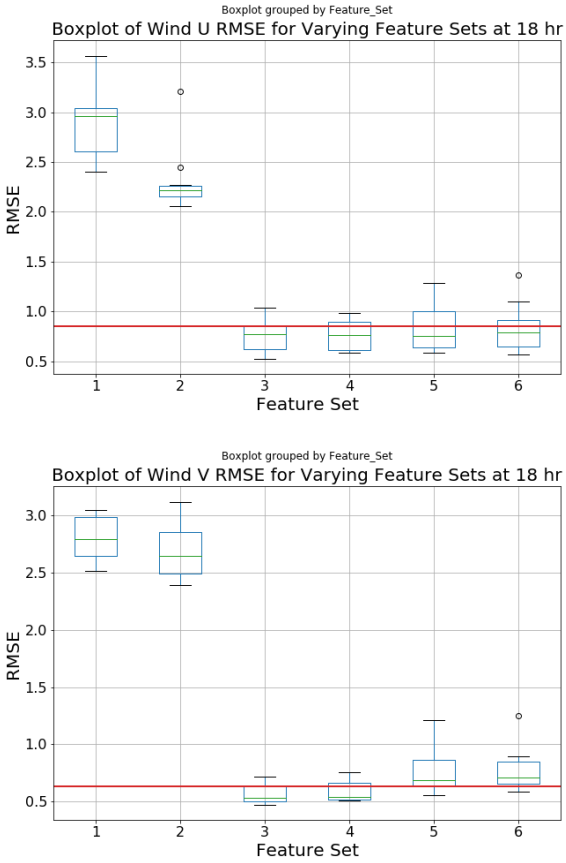

Figure 12. Feature Selection Box Plots, 18hr Results

Figure 12 above shows the box plots for u and v, for the +18 hour prediction range, $T = 18$ hours. The red line indicates the 75^{th} percentile for the value with the lowest median, which for both outputs is feature set 3. Using this threshold certain length values can be removed from consideration by observing if their 25^{th} percentile does not overlap with the red line. These groups are considered statistically different by observation. For u and v, feature sets 1-2 were eliminated.

Feature set 1 performed the worst, while feature set 2 performed only marginally better. This indicates that the neighbor set is contributing a slight benefit to accuracy but not as well as the forecast sets. Feature set 3 performed the best but it was only

marginally better than feature sets 4-6, indicating there might not be any statistical difference between the four sets. This would make sense since having extra forecast features beyond the +18 hour should not hinder the model, as long as the +18 hour feature is included. An interesting observation is that extra forecast features beyond the prediction period did not help in increasing accuracy for $T = 18$ hours. This is a detail that is examined in the rest of the prediction periods.

Examining results from Tukey's T-test showed similar conclusions. Table 14 and Table 15 show the results from these tests for u and v. These results mirror what was shown in the box plots. The forecast feature sets are statistically similar to each other while different from feature sets 1-2. The best performing feature sets for $T = 18$ hours are sets 3-6.

Table 14. Feature Set Tukey Test Results, Wind U 18 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.6222	-0.9808	-0.2637	True
1	3	-2.1813	-2.5398	-1.8227	True
1	4	-2.1676	-2.5261	-1.809	True
1	5	-2.0925	-2.451	-1.734	True
1	6	-2.098	-2.4565	-1.7394	True
2	3	-1.559	-1.9175	-1.2005	True
2	4	-1.5453	-1.9038	-1.1868	True
2	5	-1.4703	-1.8288	-1.1117	True
2	6	-1.4757	-1.8342	-1.1172	True
3	4	0.0137	-0.3448	0.3722	False
3	5	0.0887	-0.2698	0.4473	False
3	6	0.0833	-0.2752	0.4418	False
4	5	0.0751	-0.2835	0.4336	False
4	6	0.0696	-0.2889	0.4281	False
5	6	-0.0055	-0.364	0.3531	False

Table 15. Feature Set Tukey Test Results, Wind V 18 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.1088	-0.3589	0.1414	False
1	3	-2.2288	-2.479	-1.9786	True
1	4	-2.2035	-2.4537	-1.9533	True
1	5	-2.0103	-2.2605	-1.7602	True
1	6	-2.0247	-2.2748	-1.7745	True
2	3	-2.12	-2.3702	-1.8699	True
2	4	-2.0947	-2.3449	-1.8446	True
2	5	-1.9016	-2.1517	-1.6514	True
2	6	-1.9159	-2.1661	-1.6657	True
3	4	0.0253	-0.2249	0.2755	False
3	5	0.2185	-0.0317	0.4686	False
3	6	0.2042	-0.046	0.4543	False
4	5	0.1932	-0.057	0.4433	False
4	6	0.1788	-0.0713	0.429	False
5	6	-0.0143	-0.2645	0.2358	False

4.2.3.2 36 hours

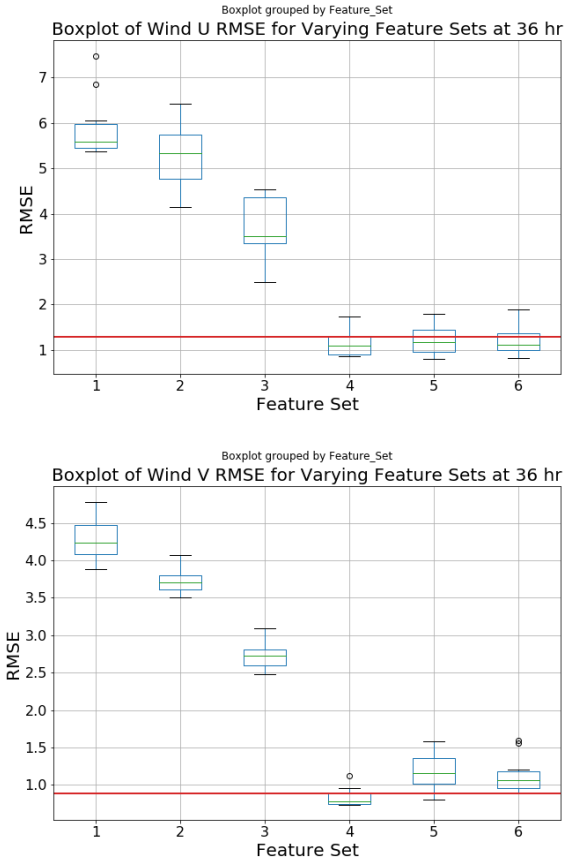

Figure 13. Feature Selection Box Plots, 36hr Results

Figure 13 shows the box plots for u and v, for the +36 hour prediction range, $T = 36$ hours. The red line indicates the 75^{th} percentile for the value with the lowest median, which for both outputs is feature set 4. Using this threshold feature sets 1-3 were eliminated for u and v.

A similar pattern is occurring here as we increase the prediction period. Feature set 1-2 still performed the worst with feature set 2 only a bit better than 1. Feature set 3 decreased in performance once the prediction period was increased beyond the forecast features that it contained. This was expected, although it is shown that having only a forecast feature earlier than the prediction period still helps improve

accuracy more than just the base features or neighbor features. Feature sets 4-6 performed the best with little being shown in u, though in v a larger difference is noticed that could lead to a statistical difference between the sets. While this would be unexpected, the small sampling of 10 runs could being having an effect on the variance.

Results from Tukey's T-test are shown in Table 16 and Table 17. The results for u show a result that is expected, with feature sets 4-6 being statistically similar to each other while being different from sets 1-3. The results from v show a more surprising result. Feature sets 1-3 behave as expected, being statistically different from the rest. The discrepancy occurs in feature sets 4-6, where sets 5 and 6 are statistically similar but different from set 4. As mentioned, this could be due to the small sampling size. Investigating the lower bounds of the CI along with the mean difference shows a relatively small difference. This could be improved by more sampling and won't be used a means to disqualify the groups. Looking at their median values from the box plots, the difference is 0.62 m/s and 0.36 m/s for set 5 and 6 respectively when compared to set 4. Realistically this isn't as huge of a difference as sets 1-3. Therefore the best performing feature sets for $T = 36$ hours are sets 4-6.

Table 16. Feature Set Tukey Test Results, Wind U 36 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.6367	-1.3701	0.0967	False
1	3	-2.2356	-2.969	-1.5021	True
1	4	-4.7607	-5.4941	-4.0273	True
1	5	-4.68	-5.4134	-3.9466	True
1	6	-4.6861	-5.4195	-3.9526	True
2	3	-1.5989	-2.3323	-0.8655	True
2	4	-4.124	-4.8574	-3.3906	True
2	5	-4.0433	-4.7767	-3.3099	True
2	6	-4.0494	-4.7828	-3.316	True
3	4	-2.5251	-3.2586	-1.7917	True
3	5	-2.4444	-3.1778	-1.711	True
3	6	-2.4505	-3.1839	-1.7171	True
4	5	0.0807	-0.6527	0.8142	False
4	6	0.0747	-0.6588	0.8081	False
5	6	-0.0061	-0.7395	0.7273	False

Table 17. Feature Set Tukey Test Results, Wind V 36 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.563	-0.8582	-0.2678	True
1	3	-1.5698	-1.865	-1.2747	True
1	4	-3.4506	-3.7458	-3.1555	True
1	5	-3.1032	-3.3984	-2.808	True
1	6	-3.1487	-3.4439	-2.8535	True
2	3	-1.0068	-1.302	-0.7116	True
2	4	-2.8876	-3.1828	-2.5924	True
2	5	-2.5402	-2.8354	-2.245	True
2	6	-2.5857	-2.8809	-2.2905	True
3	4	-1.8808	-2.176	-1.5856	True
3	5	-1.5334	-1.8285	-1.2382	True
3	6	-1.5789	-1.874	-1.2837	True
4	5	0.3474	0.0523	0.6426	True
4	6	0.3019	0.0068	0.5971	True
5	6	-0.0455	-0.3407	0.2497	False

4.2.3.3 54 hours

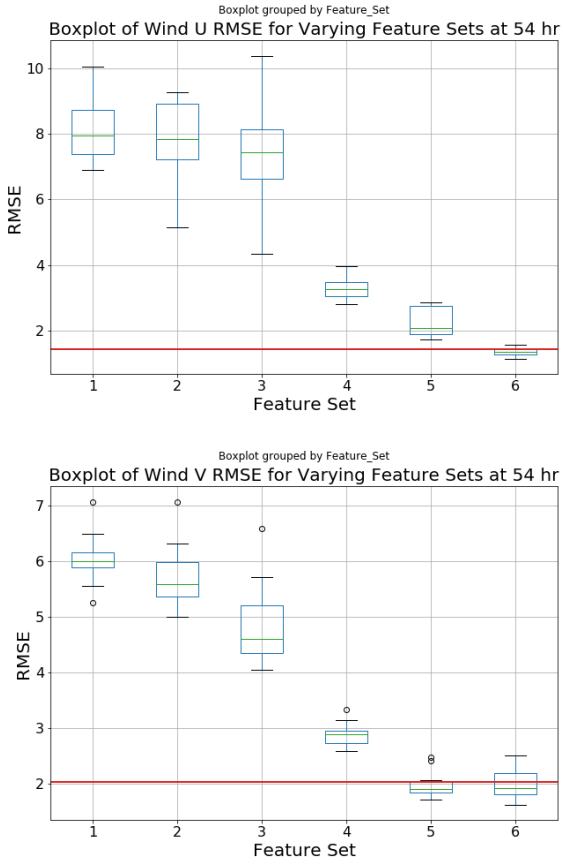

Figure 14. Feature Selection Box Plots, 54hr Results

Figure 14 above shows the box plots for u and v, for the +54 hour prediction range, $T = 54$ hours. The red line indicates the 75^{th} percentile for the value with the lowest median, which for u is feature set 6 and feature set 5 for v. Using this threshold, feature sets 1-4 were eliminated for u and v. Set 5 for u could be eliminated but further examination is needed using the Tukey T-test.

A similar pattern is occurring here as we increase the prediction period. Feature set 1-2 still performed the worst with feature set 3 decreasing even more in performance. Feature set 4 had a drop in performance but it is similar to feature set 3's drop in prediction period $T = 36$ hours. Feature sets 5-6 performed the best with

little being shown in v, though in u a larger difference is noticed that could lead to a statistical difference between the sets. Again, small sampling could be the cause of this which would need to be investigated in future work.

Results from Tukey's T-test are shown in Table 18 and Table 19. The results for u show a result that is expected, with feature sets 5-6 being statistically similar to each other while being different from sets 1-3. The interesting aspect here is set 5 being similar with 4 and 6, but set 4 being different to 6. Set 5 has an overlap with the two which is due to the larger variance apparent in that sample. The results from v show a more consistency with the previous patterns. Feature sets 5-6 are statically similar while different from the rest of the sets. The best performing feature sets for $T = 54$ hours are sets 5-6.

Table 18. Feature Set Tukey Test Results, Wind U 54 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.2855	-1.5652	0.9942	False
1	3	-0.7449	-2.0246	0.5348	False
1	4	-4.8341	-6.1138	-3.5544	True
1	5	-5.8582	-7.1379	-4.5785	True
1	6	-6.7687	-8.0484	-5.489	True
2	3	-0.4594	-1.7391	0.8203	False
2	4	-4.5486	-5.8283	-3.2689	True
2	5	-5.5727	-6.8524	-4.293	True
2	6	-6.4832	-7.7629	-5.2035	True
3	4	-4.0891	-5.3688	-2.8094	True
3	5	-5.1132	-6.3929	-3.8335	True
3	6	-6.0238	-7.3035	-4.7441	True
4	5	-1.0241	-2.3038	0.2556	False
4	6	-1.9347	-3.2143	-0.655	True
5	6	-0.9106	-2.1902	0.3691	False

Table 19. Feature Set Tukey Test Results, Wind V 54 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.3167	-0.9646	0.3312	False
1	3	-1.1636	-1.8114	-0.5157	True
1	4	-3.1414	-3.7893	-2.4936	True
1	5	-4.0426	-4.6905	-3.3947	True
1	6	-4.0543	-4.7022	-3.4064	True
2	3	-0.8468	-1.4947	-0.199	True
2	4	-2.8247	-3.4726	-2.1768	True
2	5	-3.7259	-4.3738	-3.078	True
2	6	-3.7376	-4.3855	-3.0897	True
3	4	-1.9779	-2.6258	-1.33	True
3	5	-2.8791	-3.5269	-2.2312	True
3	6	-2.8907	-3.5386	-2.2428	True
4	5	-0.9012	-1.549	-0.2533	True
4	6	-0.9128	-1.5607	-0.2649	True
5	6	-0.0117	-0.6596	0.6362	False

4.2.3.4 90 hours

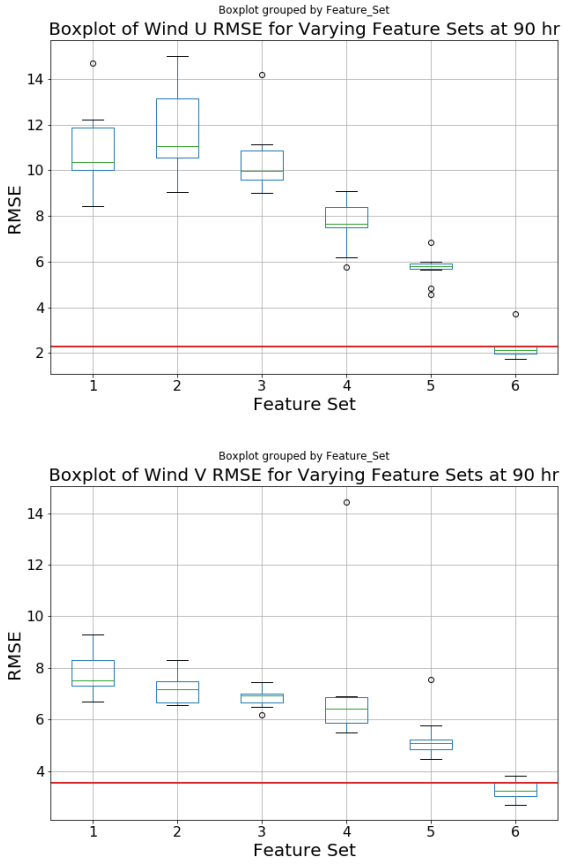

Figure 15. Feature Selection Box Plots, 90hr Results

Figure 15 shows the box plots for u and v, for the +90 hour prediction range, $T = 90$ hours. The red line indicates the 75^{th} percentile for the value with the lowest median, which for u and v is feature set 6. Using this threshold feature sets 1-5 were eliminated for u and v. Steadily increasing the time steps from the forecast feature to the prediction period decreases accuracy until it performs no better than the base set. Having a forecast feature including the prediction period greatly increases accuracy.

Results from Tukey's T-test are shown in Table 20 and Table 21. The results for u and v show that feature set 6 is statistically different from the rest of the sets. This indicates feature set 6 is the best performing set for $T = 90$ hours.

Table 20. Feature Set Tukey Test Results, Wind U 90 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	0.7698	-1.078	2.6176	False
1	3	-0.4596	-2.3074	1.3882	False
1	4	-3.221	-5.0688	-1.3732	True
1	5	-5.2098	-7.0576	-3.362	True
1	6	-8.6789	-10.5266	-6.8311	True
2	3	-1.2294	-3.0772	0.6184	False
2	4	-3.9908	-5.8386	-2.143	True
2	5	-5.9796	-7.8274	-4.1318	True
2	6	-9.4487	-11.2965	-7.6009	True
3	4	-2.7614	-4.6092	-0.9136	True
3	5	-4.7502	-6.598	-2.9024	True
3	6	-8.2193	-10.0671	-6.3715	True
4	5	-1.9888	-3.8366	-0.141	True
4	6	-5.4578	-7.3056	-3.61	True
5	6	-3.469	-5.3168	-1.6213	True

Table 21. Feature Set Tukey Test Results, Wind V 90 hours

Feature Group 1	Feature Group 2	Mean Difference	Lower	Upper	Reject
1	2	-0.5821	-2.2066	1.0424	False
1	3	-0.9162	-2.5407	0.7083	False
1	4	-0.683	-2.3075	0.9415	False
1	5	-2.4915	-4.116	-0.867	True
1	6	-4.468	-6.0925	-2.8435	True
2	3	-0.334	-1.9585	1.2905	False
2	4	-0.1009	-1.7254	1.5236	False
2	5	-1.9094	-3.5339	-0.2849	True
2	6	-3.8859	-5.5104	-2.2614	True
3	4	0.2331	-1.3914	1.8576	False
3	5	-1.5754	-3.1999	0.0491	False
3	6	-3.5518	-5.1763	-1.9273	True
4	5	-1.8085	-3.433	-0.184	True
4	6	-3.785	-5.4095	-2.1605	True
5	6	-1.9765	-3.601	-0.352	True

4.2.3.5 Feature Selection Conclusion

Since feature set 6 was included in the best performing sets for all prediction periods, this set is used for training the final model. This set includes the base features, along with the deterministic forecast features from the +18, +36, +54, and +90 hour forecast.

4.2.4 LSTM Final Architecture

With the large amount of parameter options looked at for LSTM architecture tuning, an iterative process helped eliminate options that did not perform well. The first parameter investigated was the optimizer employed. Figure 16 shows the results for the validation MSE from an aggregated list of options. The first noticeable thing is RMSprop consistently performs poorly when paired with the sigmoid and tanh activation functions. While having 2 layers reduces this only for the larger layer widths, it does not correct it.

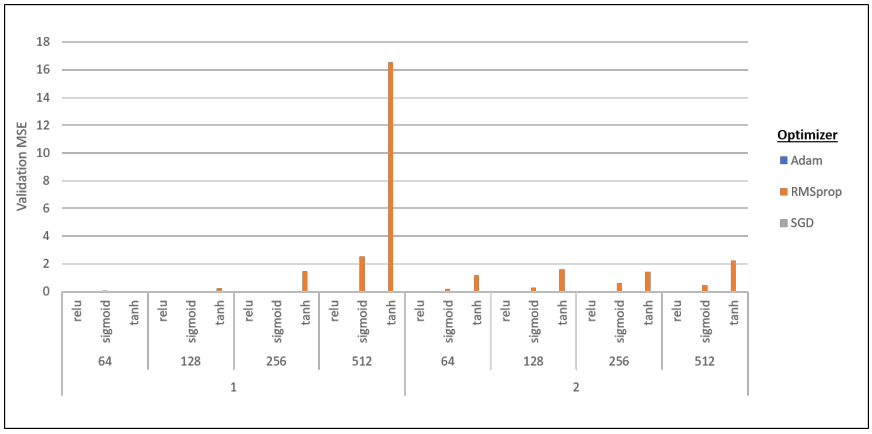

Figure 16. Architecture Selection Initial Results

Activation functions tanh and sigmoid were filtered out to further explore the RMSprop optimizer. These results are shown in Figure 17. The ReLU activation function seems to fix the issue RMSprop was having with the training data. This

piece of evidence indicates that the combination of sigmoid and tanh with RMSprop was running into the vanishing gradient problem. This is an issue corrected with the ReLU function. Also, the other optimizers have varying built in features to mitigate this issue. Since RMSprop did not perform better than Adam even with a ReLU activation function, RMSprop was removed from consideration.

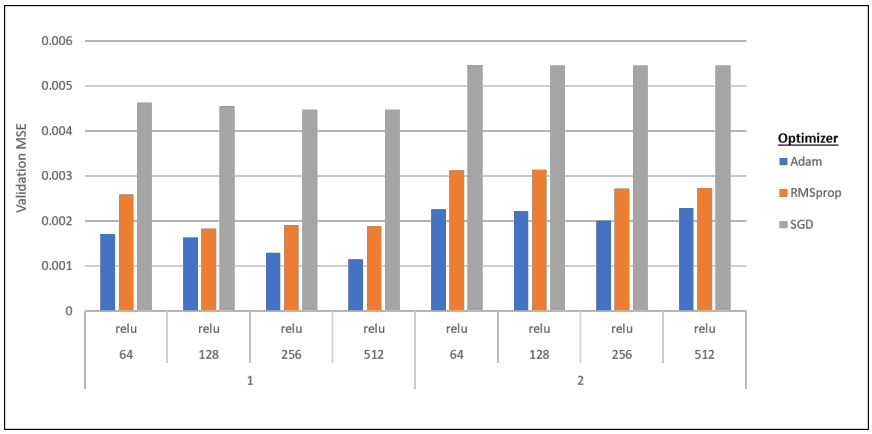

Figure 17. Architecture Selection, ReLU results

Figure 18 shows the filtered results with RMSprop removed. The SGD optimizer consistently under performs Adam for every option combination without dispute. Due to that, SGD was removed leaving Adam to be the best performing optimizer.

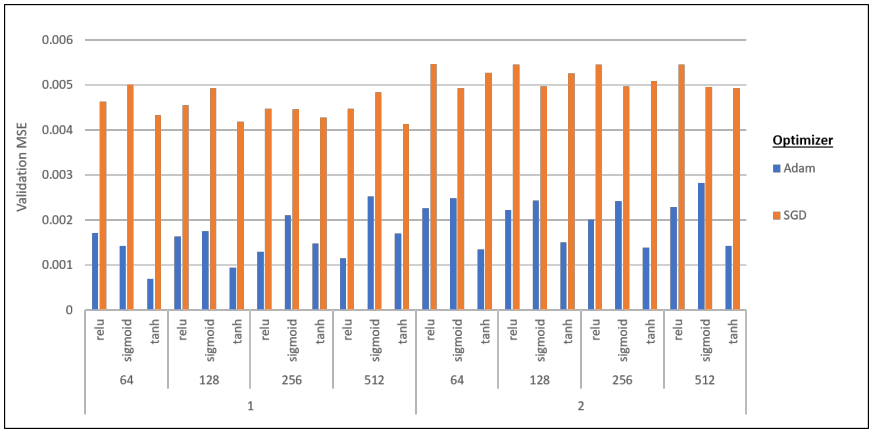

Figure 18. Architecture Selection, SGD and Adam Results

Activation functions were explored next. Table 19 shows results filtered to include

only the Adam optimizer with all the rest of the parameter options. The tanh function is the best performing activation function here. ReLU comes the second closest, and only beats it on one layer networks when the layer width grows larger 128 nodes. On the 2 layer network, this does not happen and tanh out preforms ReLU on all explored layer widths. With this all other activation functions except tanh were removed, and learning rates were explored next.

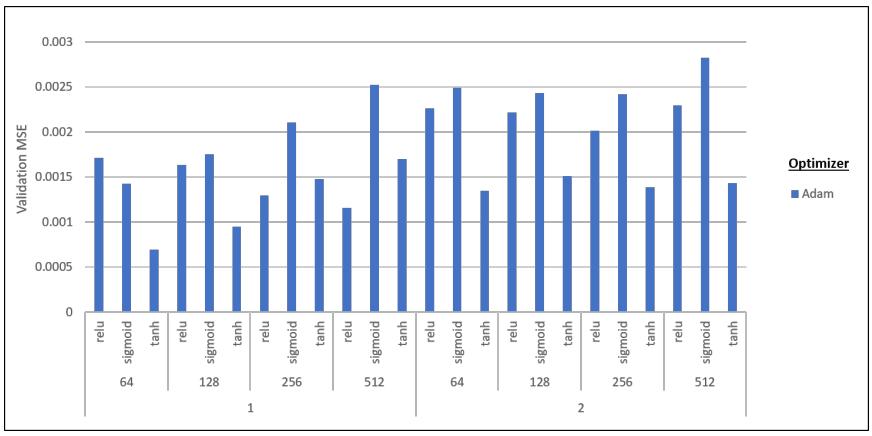

Figure 19. Architecture Selection, Adam Results

Figure 20 shows the filtered results with the Adam optimizer and tanh activation function. Exploring the learning rates, there is a consistent pattern of the largest learning rates having the highest error while the opposite is true for the lower learning rates. Based on that evidence a learning rate of 0.001 was chosen for the final model. The last step was to explore the number of nodes and layers.

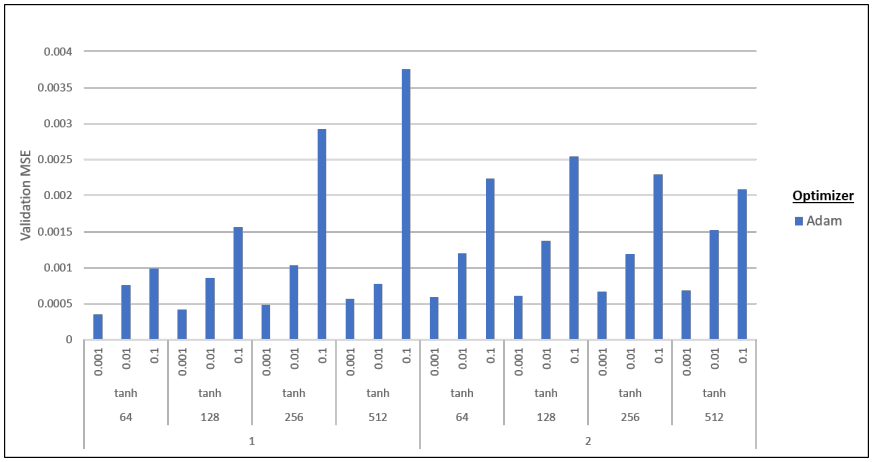

Figure 20. Architecture Selection, Adam and tanh Results

Figure 21 shows the filtered results to explore the number of nodes and layers that perform best for the model. The results are not sorted even though they have that look. The performance of the model seemed to steadily decrease as the layer width increased, and with the addition of the extra layer. This could be due to the small sample size. Although the patience is meant to help correct over fitting, the speed at which it may occur with the addition of extra layers and increased layer width may be the cause of the increased error. The final model will go with one layer with 64 nodes.

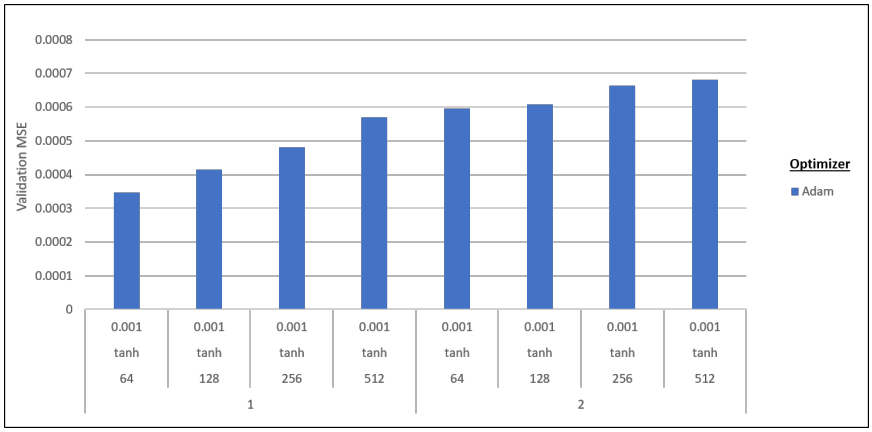

Figure 21. Architecture Selection, Learning Rates Results

The only thing not shown in these graphs was patience. This parameter did not

seem to have a large effect past 50 epochs, so the patience was set to 50 epochs. The final LSTM architecture parameters are displayed in Table 22.

Table 22. Final LSTM Architecture Parameters

Parameter	Value
Layers	1
Layer Width	64
Activation Function	tanh
Optimizer	Adam
Learning Rate	0.001
Patience	50

4.3 Model Design Choice

The final LSTM architecture was tested against an ensemble network of six LSTM models. Both were trained against C_0, with the individual ensemble members being trained on six random coordinates from C_i. The test was done against 4,000 randomly sampled coordinates, with the results shown in Figure 22. The plots are in order from $T = 18$ hours to $T = 90$ hours. Table 23 displays the median and mean values for each scenario.

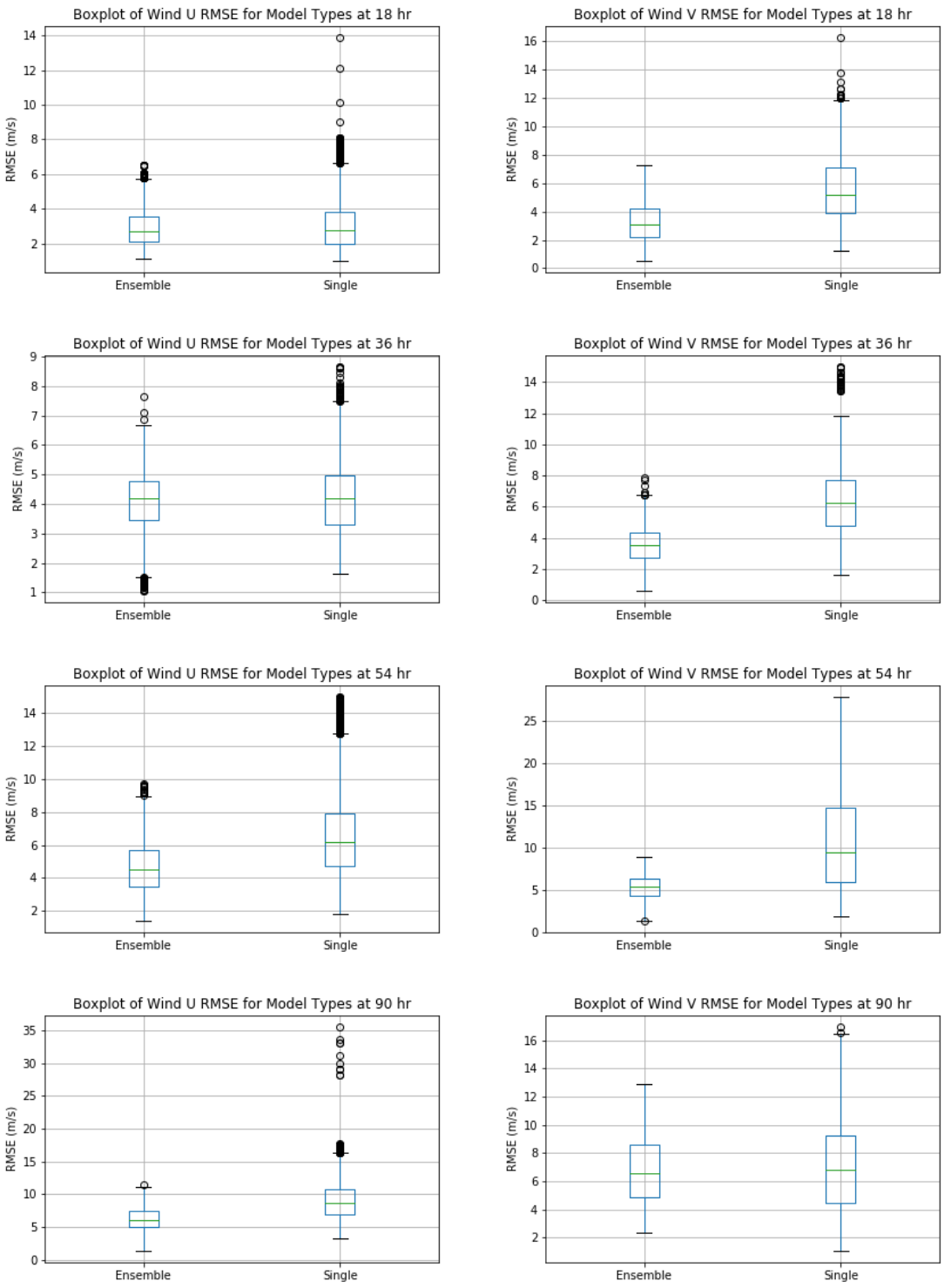

Figure 22. Model Selection Box Plot Results

Table 23. Model Selection Numeric Results

Prediction Period	Model	*u* Mean	*u* Median	*v* Mean	*v* Median
18 hrs	Ensemble	2.927	2.719	3.241	3.088
	Single	3.116	2.763	5.593	5.139
36 hrs	Ensemble	4.073	4.184	3.475	3.570
	Single	4.177	4.189	6.370	6.248
54 hrs	Ensemble	4.679	4.538	5.219	5.335
	Single	6.627	6.189	10.770	9.489
90 hrs	Ensemble	6.225	5.979	6.743	6.560
	Single	9.207	8.632	7.200	6.795

Initial investigation of the box plots reveals the medians for both model types being slightly close, with the ensemble edging out each time. The single model tends to suffer from more outliers and higher variance. Investigation of the outliers revealed a handful of coordinates responsible for the behavior. They were not removed since the ensemble model was also tested against the same coordinates, but produced better results and less outliers than the single LSTM model. The difference in performance became more apparent as the prediction period increased towards $T = 90$ hours. The exception being v at $T = 90$ hours for the single model LSTM displayed performance that was closer to the ensemble than in the previous prediction period. To test for statistically significant differences between the model types, a T-test was performed. The results for u are displayed in Table 24, and the results for v are displayed in Table 25.

Table 24. Model Selection T-Test Results, Wind U

Prediction Period	Mean Difference	Lower	Upper	Reject
18 hour	-0.4572	-0.6080	-0.3065	True
36 hour	-0.1036	-0.1653	-0.0420	True
54 hour	-1.9478	-2.2042	-1.6914	True
90 hour	-4.1205	-4.9489	-3.2920	True

Table 25. Model Selection T-Test Results, Wind V

Prediction Period	Mean Difference	Lower	Upper	Reject
18 hour	-2.4621	-2.7132	-2.2110	True
36 hour	-2.8947	-3.0554	-2.7340	True
54 hour	-5.4937	-6.5842	-4.4032	True
90 hour	-0.6101	-1.0013	-0.2190	True

These results indicate that a statistical difference exist between the ensemble model and the single LSTM model at a 95% confidence level. Sample size is not an issue here since 4,000 samples were used to generate the results. Going from the mean or median values the ensemble performed better in each metric. Therefore the ensemble model was chosen as the final model to test against the deterministic forecast.

4.4 Deterministic Forecast Comparison

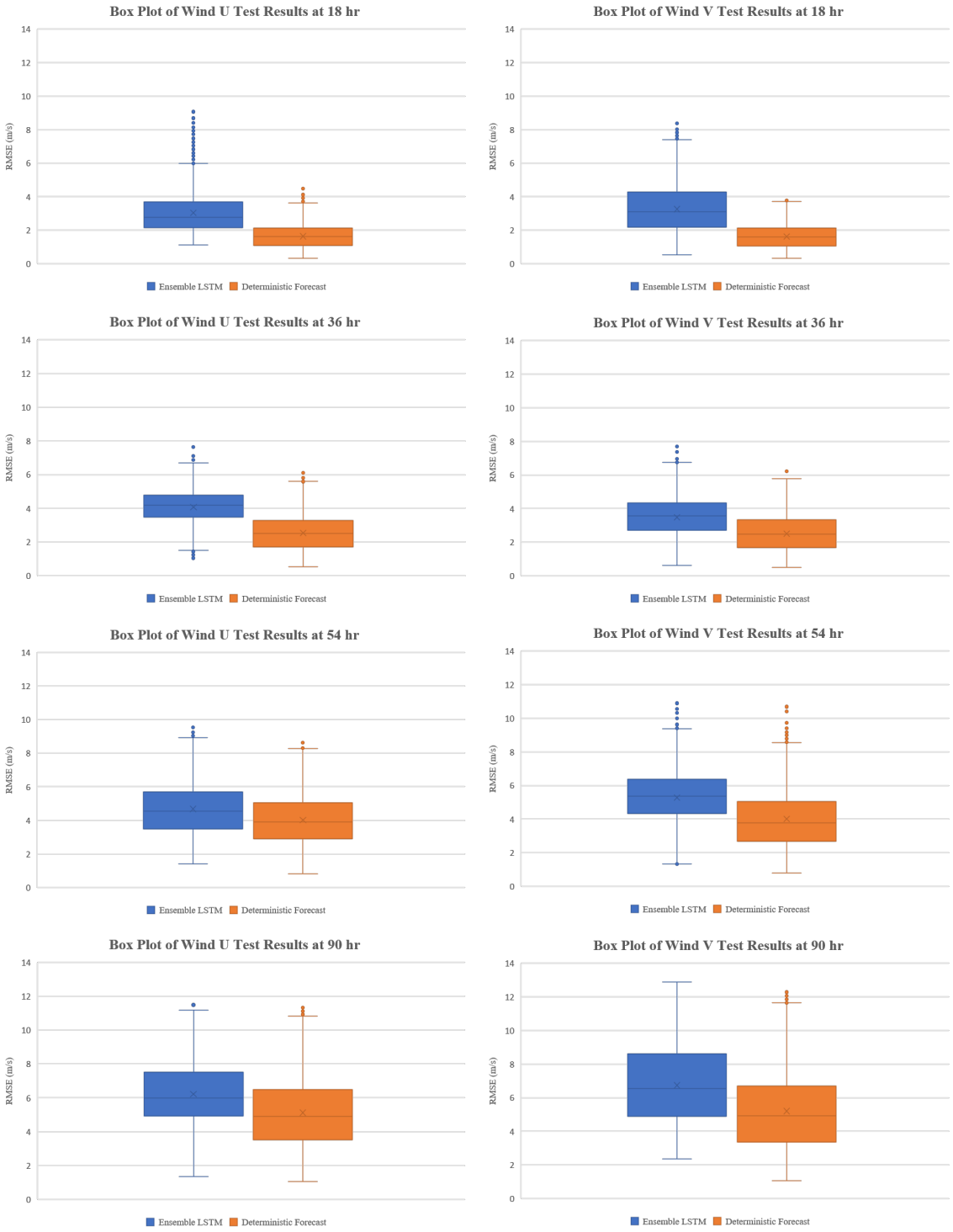

Figure 23. Deterministic Comparison Box Plot Results

Figure 23 displays the box plot results for the test between the deterministic forecast and the ensemble LSTM for 10,000 randomly sampled coordinates. The deterministic forecast appears to be better, but only marginally. This is more distinctive in the larger prediction periods where the uncertainty increases and the models appear to be very similar. In the smaller prediction periods the deterministic forecast seems to have an advantage in terms of a tighter variance. The numeric results for the mean and median are shown in Table 26.

Table 26. Deterministic Comparison Numeric Results

Prediction Period	Model	*u* Mean	*u* Median	*v* Mean	*v* Median
18 hrs	Deterministic	1.642	1.615	1.627	1.599
	LSTM	3.037	2.764	3.268	3.098
36 hrs	Deterministic	2.535	2.498	2.515	2.494
	LSTM	4.073	4.184	3.475	3.570
54 hrs	Deterministic	4.009	3.899	4.011	3.789
	LSTM	4.679	4.538	5.277	5.368
90 hrs	Deterministic	5.123	4.886	5.202	4.938
	LSTM	6.225	5.979	6.743	6.560

A t-test was conducted to examine statistical differences in the two models. Table 27 and Table 28 show the results for u and v, respectively. The results show that the difference between the models is statistically significant to a confidence level of 95%. This difference can be put into perspective with the margin of error. The LSTM ensemble model's error was on average 1.18 m/s for +18 hours, 1.25 m/s for +36 hours, 0.97 m/s for +54 hours, and 1.32 m/s for +90 hours higher than the deterministic forecast. While the model is not statistically similar, the difference in error is small. Also, in approximately 11% of points the LSTM ensemble model outperformed the deterministic forecast by an average of 0.67 m/s. This indicates

the model is benefiting from the addition of the forecast features to learn the true wind speed values as opposed to being bound by the value of the forecast feature. This is further seen in Figure 24 where the predicted wind speeds for a random point is shown against the truth values at varying altitudes. While the LSTM may not be outperforming the deterministic model, it is still showing its ability to learn the complex underlying wind dynamics.

Table 27. Deterministic Comparison T-Test Results, Wind U

Prediction Period	Mean Difference	Lower	Upper	Reject
18 hour	1.3947	1.3479	1.4414	True
36 hour	1.5385	1.4898	1.5872	True
54 hour	0.6704	0.5661	0.7747	True
90 hour	1.1015	0.9388	1.2642	True

Table 28. Deterministic Comparison T-Test Results, Wind V

Prediction Period	Mean Difference	Lower	Upper	Reject
18 hour	1.6410	1.5676	1.7144	True
36 hour	0.9600	0.8999	1.0202	True
54 hour	1.2660	1.1405	1.3915	True
90 hour	1.5404	1.3167	1.7641	True

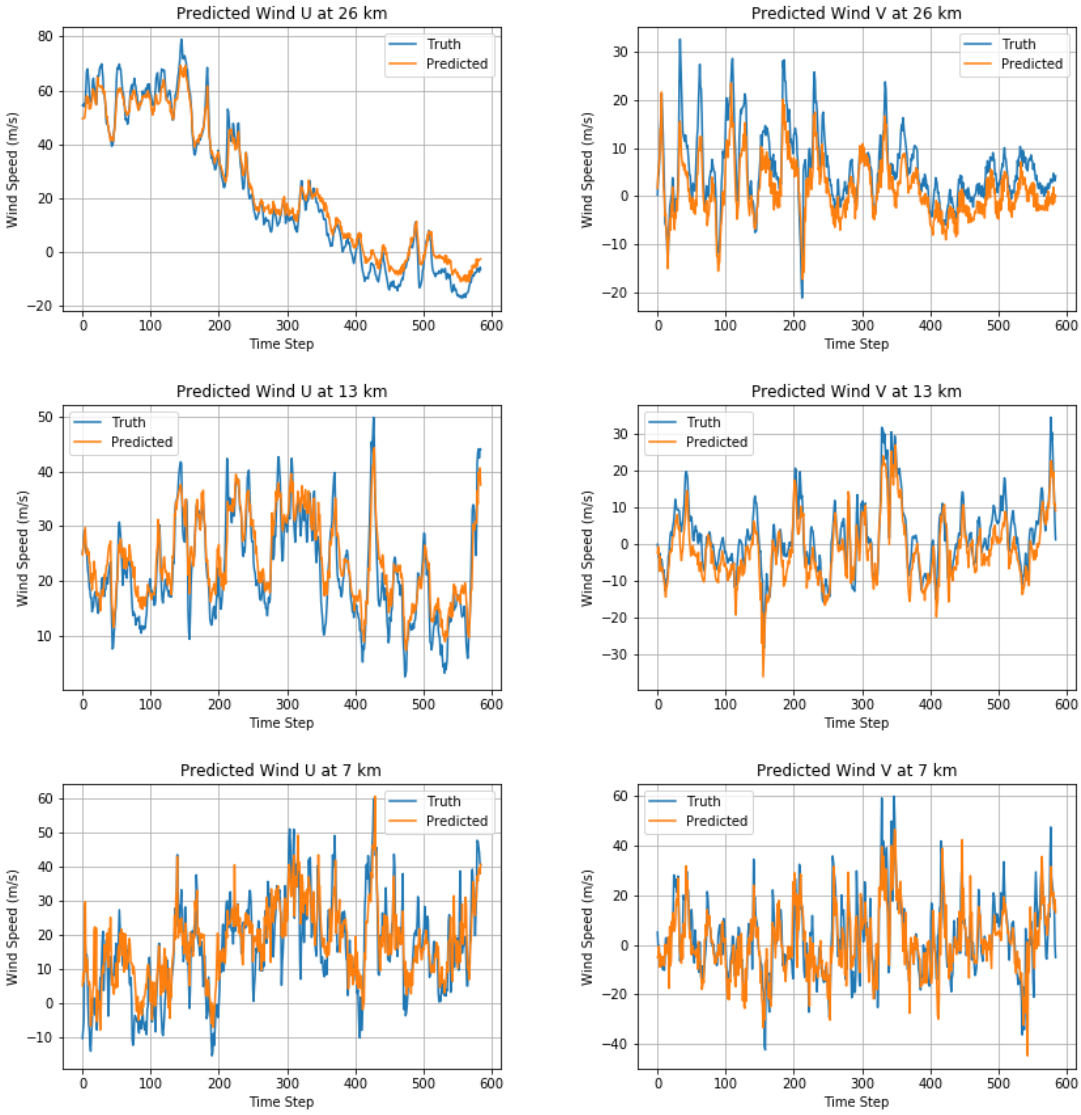

Figure 24. Predicted Wind Components at Coordinate (52°, 13°)

4.5 Flight Path Results

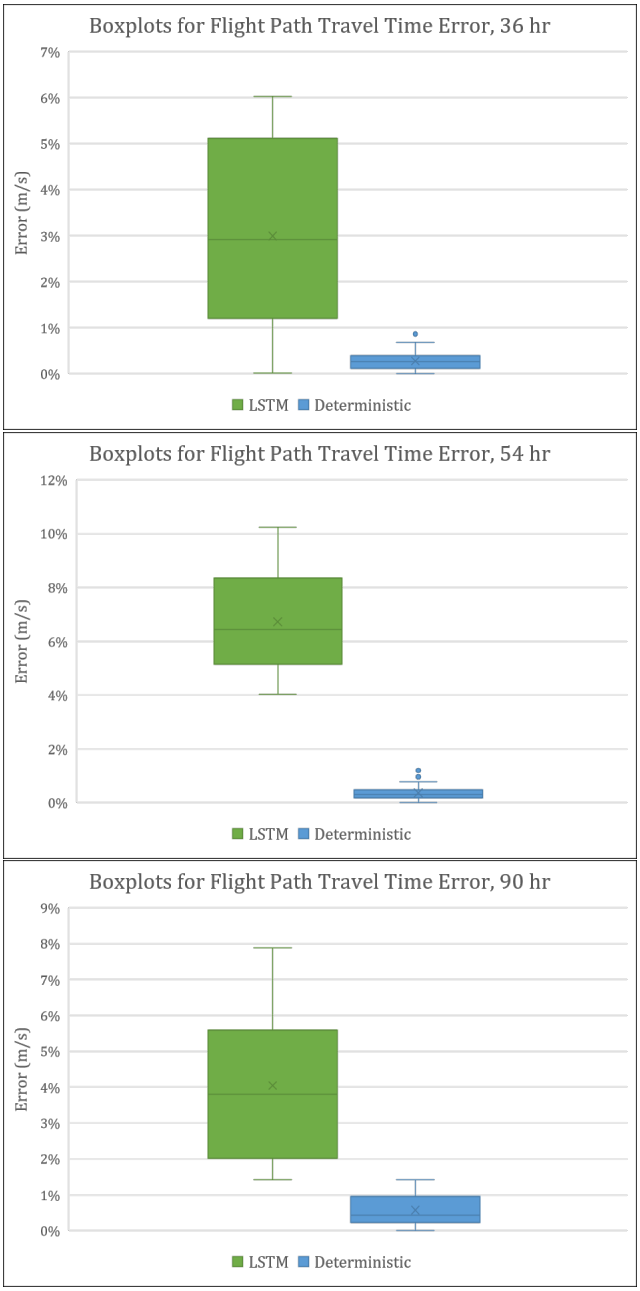

Figure 25. Percent Error for Flight Path Travel Time

Table 25 displays the results for the percent error of the average predicted travel time from McGuire AFB to Ramstein AFB for the deterministic NWP and the LSTM

model. It is not surprising that the deterministic model has lower error than the LSTM. The previous sections already highlighted this difference in accuracy. Considering this, the actual percent difference is relatively quite small for the LSTM model. Table 29 shows the numeric results for both models while Table 30 shows the predicted travel times for this particular route. The GFS represents the best estimate using the truth information.

Table 29. Numeric Results for Flight Path Travel Time Error

Prediction Period	Model	Mean	Stdev
36 hrs	Deterministic	0.28%	0.20%
	LSTM	3.00%	2.06%
54 hrs	Deterministic	0.37%	0.29%
	LSTM	6.72%	1.89%
90 hrs	Deterministic	0.57%	0.44%
	LSTM	4.05%	1.96%

Table 30. Predicted Flight Path Travel Time

Prediction Period	Model	Average Travel Time	Error
36 hrs	LSTM	7 hours 42.7 mins	13.3 mins
	Deterministic	7 hours 29.4 mins	1.3 mins
	GFS Truth	7 hours 29.5 mins	-
54 hrs	LSTM	8 hours 0.3 mins	30.1 mins
	Deterministic	7 hours 29.9 mins	1.7 mins
	GFS Truth	7 hours 30.2 mins	-
90 hrs	LSTM	7 hours 48.5 mins	18.1 mins
	Deterministic	7 hours 30.5 mins	2.6 mins
	GFS Truth	7 hours 30.4 mins	-

With a percent error ranging from 3-7% for the LSTM model, this accounts to only 13-30 min or error in the predicted travel time for an approximately 7 hour and

30 min flight. The trade-off that needs to be considered here is if the decrease in accuracy for the LSTM is worth the benefit of decreased computation times. While the 36 hour LSTM model maybe be off by 13 min, it will give an answer in a couple seconds compared to the couple hours the deterministic NWP takes to run. **This significant decrease in computation time for the LSTM model is worth consideration for all future flight planning needs**.

Table 31. Model Computation Times

Model	Approximate Run Time
LSTM	1.85 seconds
Deterministic NWP	3 hours

V. Conclusions and Future Research

5.1 Conclusion

This study examined the potential for LSTMs to model and predict upper atmospheric wind speeds for any given latitude and longitude coordinate. The prediction periods examined were $T = (18, 36, 54, 90)$ hours with the results being compared against the GFS deterministic weather model for the same forecast periods. The goal was to reduce the error in the deterministic model, thereby giving users of the forecast data more accurate information. A secondary goal was to develop a model that computed faster than current methods with a reasonable level of accuracy. This would be usable by AMC for aircraft mission planning to build more fuel efficient routes, and reduce fuel usage thereby saving money to the AF for fuel procurement. A faster model would also increase productive time for mission planners, and allow them to respond rapidly to building flight plans.

An important insight from this study was the importance of model tuning and the effect it can have on results. Keying in on the LSTM parameters, the difference between pairing a tanh activation function with the RMSprop optimizer as opposed to the Adam optimizer had stark differences with this data set. The Adam optimizer significantly outperformed the RMSprop optimizer when paired with the tanh or sigmoid activation function. Exploring layer width and learning rates had similar insights. While higher learning rates were expected to not perform well, more unexpected was that a larger layer width also did not perform as well. This can be an overcapacity to the network where the model over fits faster than the validation methods can terminate the training at an appropriate point. Other interesting insights came from the features used in the model. While it was expected that adding the forecast features would help, having a forecast feature further away in the past

from the predicted period provides negligible help. The range seems to grow as the prediction period extends out.

While the ensemble LSTM model was the best performing of the architectures tested, it did not beat the deterministic model on a majority of occasions. Although the difference between the two for error was small, with the ensemble LSTM on average off by 1.18 m/s for $T = 18$ hours, 1.25 m/s for $T = 36$ hours, 0.97 m/s for $T = 54$ hours, and 1.32 m/s for $T = 90$ hours. On 11% of tested coordinates the ensemble LSTM outperformed the deterministic model by an average of 0.67 m/s. These results demonstrate a viability for this technique exist, and can be improved upon if certain shortfalls and limitations are addressed for further research.

Even though the LSTM did not achieve its accuracy goal, its error was relatively close. Removing the forecast features increased the error a small amount, but allowed the LSTM model to function without being reliant on results from the deterministic NWP model. In this instance instead of acting as a post processor for error, it was used as a means to achieve similar predictive results at an incredibly faster rate. The LSTM was able to optimize flight paths between two points with an average error of 4.6% to the travel time using predictions from +36, +54, and +90 hours out. The computation time was approximately 1.85 seconds compared to the deterministic NWP which takes approximately 3 hours. That is 7,200 times faster than current methods for approximate solutions that are close to truth. This is a major benefit for flight planners, as it increases productive time and allows rapid generation of flight plans. It also does not require anything more than a basic computer or laptop, and as such enables its ability to be deployed in a myriad of situations and environments.

5.2 Limitations

There were some limitations in this study. The biggest limitation was the size of the data. While the results of this study were promising, it was trained on a small set representing six months or half a season of weather. This is not enough to fully explore the ability of the LSTM to learn the weather dynamics, and generalize them. The smaller data size also contributed to limited feature selection. While a multitude of weather variables existed to include in the model, limited features were chosen as to not incur a dimensionality issue while training the data.

A second limitation presented itself in the complex nature of sequential weather data. Training the models takes an extensive amount of time, upwards to weeks to fully complete parameter tuning. This prevented a full exploration of parameter values and feature variables that could improve the network. It also deterred from using more complex LSTM variant networks, which could possibly assist in improving accuracy.

5.3 Future Research

This study serves as a starting point for a number of future projects. With a benchmark established for ensemble LSTM performance, the next steps would look to improve upon the existing architecture or available data. A reforecast data set exist from NOAA which contains data from 1984 with readings done every day instead of the six hour intervals used here. This set offers the potential to test longer series of weather data, and would allow the exploration of a deeper LSTM network. With more data available, more features could be explored for adding into the model since dimensionality would be less on an issue. Some of these features could include relative humidity, max ground wind speeds, convective potential energy, potential vorticity

surfaces, etc.

Another area to explore would be the flight planning itself and the path optimization. Building a more complex time dynamic three dimensional network could offer interesting insights into the best flight path that maximizes tailwinds. The existing LSTM architecture could be used for this as it shows relatively small error and provides wind speed predictions at an incredibly fast rate.

Future work can be done studying the coordinate regions. With over 2 billion coordinates, a deeper study could investigate the relationships between different global regions and their ability to model other regions. This could lead to developing of regional LSTM models built from coordinates that characterize that area, and generalize to similar areas around the globe. These regional weather similarities can be of importance to many national weather agencies.

Appendix A. Weather Truth Data Pre-Processing MATLAB Code

```
c = {}; % create empty cell to hold specific variables from data
countF = 0; % count number of actual files opened
time = [0, 6, 12, 18]; % array of time intervals

for k = 8:9 % year
    for j = 1:12 % months
        for i = 1:31 % days
            for n = 1:4 % time intervals
                try
                    % create filename to open
                    if j < 10
                        month = strcat(num2str(0),num2str(j));
                    else
                        month = num2str(j);
                    end

                    if i < 10
                        day = strcat(num2str(0),num2str(i));
                    else
                        day = num2str(i);
                    end

                    if n < 3
                        timeS = strcat(num2str(0),num2str(time(n)));
                    else
                        timeS = num2str(time(n));
                    end

                    tic
```

```
30
31                  date = strcat('201',num2str(k),month,day);
32                  file = strcat('G:\Thesis\Deterministic\',date,'\
    ',date,timeS,'\gfs_3_',date,'_',timeS,'00_000.grb2');
33                  ds = ncgeodataset(file);% open file
34                  var = ds.variables;% save variable data
35
36                  %len = length(var);
37                  %meta = ds.metadata;
38
39                  countF = countF + 1;
40
41                  % Temperature data
42                  c{countF,1} = ds.data(var{11});
43                  c{countF,1} = squeeze(c{countF,1});
44
45                  % Wind-U data
46                  c{countF,2} = ds.data(var{37});
47                  c{countF,2} = squeeze(c{countF,2});
48
49                  % Wind-V data
50                  c{countF,3} = ds.data(var{46});
51                  c{countF,3} = squeeze(c{countF,3});
52
53                  % Geopotential Height data
54                  c{countF,4} = ds.data(var{74});
55                  c{countF,4} = squeeze(c{countF,4});
56
57                  % Pressure Layer data
58                  c{countF,5} = ds.data(var{108});
59                  c{countF,5} = squeeze(c{countF,5});
60
```

```
61
62                      toc
63
64                  catch
65
66                  end
67              end
68          end
69      end
70  end
71
72  % Parse out all data into separate time series files for each
        coordinate
73
74  for lat = 1:181
75      for long = 1:360
76          for alt = 1:31
77              tic
78              data = zeros(length(c),7);
79
80              for loc = 1:length(c)
81
82                  tempArr = c{loc,1};
83                  WindUArr = c{loc,2};
84                  WindVArr = c{loc,3};
85                  GeoArr = c{loc,4};
86                  PresArr = c{loc,5};
87
88                  data(loc,1) = lat - 91;
89                  data(loc,2) = long - 180;
90                  data(loc,3) = PresArr(alt);
91                  data(loc,4) = GeoArr(alt,lat,long);
```

```
92                  data(loc,5) = tempArr(alt,lat,long);
93                  data(loc,6) = WindUArr(alt,lat,long);
94                  data(loc,7) = WindVArr(alt,lat,long);
95
96              end
97
98              data = data(any(data,2),:);
99              fileName = strcat('D:\Thesis\Points\data_all_',num2str(
    lat),'_',num2str(long),'_',num2str(alt),'.txt') %Enter your own
    file destination
100             writematrix(data, fileName);
101
102             toc
103         end
104     end
105 end
```

Appendix B. Weather Forecast Data Pre-Processing MATLAB Code

```
c = {}; % create empty cell to hold specifc variables form data
countF = 0; % count number of actual files opened
time = [0, 6, 12, 18]; % array of time intervals
forecast = [18, 36, 54, 90]; % forecast to pull

for h = 1:length(forecast)
    for k = 8:9 % year
        for j = 1:12 % months
            tic
            for i = 1:31 % days
                for n = 1:4 % time intervals
                    try
                        % create filename to open
                        if j < 10
                            month = strcat(num2str(0),num2str(j));
                        else
                            month = num2str(j);
                        end

                        if i < 10
                            day = strcat(num2str(0),num2str(i));
                        else
                            day = num2str(i);
                        end

                        if n < 3
                            timeS = strcat(num2str(0),num2str(time(n
    )));
                        else
```

```
29                          timeS = num2str(time(n));
30                      end
31
32                      date = strcat('201',num2str(k),month,day);
33                      file = strcat('G:\Thesis\Deterministic\',
    date,'\',date,timeS,'\gfs_3_',date,'_',timeS,'00_0',num2str(
    forecast(h)),'.grb2');
34                      ds = ncgeodataset(file);% open file
35                      var = ds.variables;% save variable data
36
37                      %len = length(var);
38                      %meta = ds.metadata;
39
40                      countF = countF + 1;
41
42                      % Temperature data
43                      c{countF,1} = ds.data(var{12});
44                      c{countF,1} = squeeze(c{countF,1});
45
46                      % Wind-U data
47                      c{countF,2} = ds.data(var{54});
48                      c{countF,2} = squeeze(c{countF,2});
49
50                      % Wind-V data
51                      c{countF,3} = ds.data(var{63});
52                      c{countF,3} = squeeze(c{countF,3});
53
54                      % Geopotential Height data
55                      c{countF,4} = ds.data(var{101});
56                      c{countF,4} = squeeze(c{countF,4});
57
58                      % Pressure Layer data
```

```
                        c{countF,5} = ds.data(var{154});
                        c{countF,5} = squeeze(c{countF,5});

                        % Note that for the forecast, the variables of interest are located in different indices of the ds.data array


                    catch

                    end
                end
            end
            toc
        end
    end
end

for h = 1:length(forecast)
    for lat = 1:55
        tic
        for long = 1:360
            for alt = 1:31
                data = zeros(length(c),7);

                for loc = 1:length(c)

                    if loc ~= 225

                        tempArr = c{loc,1};
                        WindUArr = c{loc,2};
                        WindVArr = c{loc,3};
                        GeoArr = c{loc,4};
```

```
90                          PresArr = c{loc,5};
91
92                          data(loc,1) = lat - 91;
93                          data(loc,2) = long - 180;
94                          data(loc,3) = PresArr(alt);
95                          data(loc,4) = GeoArr(alt,lat,long);
96                          data(loc,5) = tempArr(alt,lat,long);
97                          data(loc,6) = WindUArr(alt,lat,long);
98                          data(loc,7) = WindVArr(alt,lat,long);
99
100                     end
101                 end
102
103                 data = data(any(data,2),:);
104                 fileName = strcat('D:\Thesis\Points_54hr\data_all_',
    num2str(lat),'_',num2str(long),'_',num2str(alt),'_0',num2str(
    forecast(h)),'.txt');
105                 writematrix(data, fileName);
106
107             end
108         end
109         toc
110     end
111 end
```

Bibliography

1. Christopher Olah, "Understanding lstm networks," 2015, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

2. U.S. Department of Defense, "Fiscal Year 2019 Operational Energy Budget Certification Report Assistant Secretary of Defense for Energy, Installations, and Environment," Tech. Rep., 2018.

3. U.S. Department of Defense, "United States Department of Defense Fiscal Year 2020 Budget Request," Tech. Rep., 2019.

4. Jonathan Lovegren and R. John Hansman, "Estimation of Potential Aircraft Fuel Burn Reduction in Cruise Via Speed and Altitude Optimization Strategies," *Department of Aeronautics and Astronautics*, , no. February, 2011.

5. National Centers for Environmental Prediction, "Ensemble Prediction Systems," `https://www.wpc.ncep.noaa.gov/ensembletraining/`, 2006, Accessed: 2019-10-02.

6. NOAA Center for Weather and Climate Prediction, "The Global Forecast System - Global Spectral Model," `https://www.emc.ncep.noaa.gov/GFS/doc.php`, 2016, Accessed: 2019-10-02.

7. Martin Leutbecher and Tim N Palmer, "Ensemble forecasting," *Journal of computational physics*, vol. 227, no. 7, pp. 3515–3539, 2008.

8. Elham Alipourtarzanagh and Mehrdad Boroushaki, "Dynamic Modeling of Wind Speed and Temperature Using Nonlinear Auto Regressive with eXogenous (NARX)," *International Academic Journal of Science and Engineering*, vol. 3, no. 6, pp. 56–73, 2016.

9. Gyanesh Shrivastava, Sanjeev Karmakar, Manoj Kumar Kowar, and Pulak Guhathakurta, "Application of artificial neural networks in weather forecasting: a comprehensive literature review," *International Journal of Computer Applications*, vol. 51, no. 18, 2012.

10. T. N. Krishnamurti, C. M. Kishtawal, Zhan Zhang, Timothy LaRow, David Bachiochi, Eric Williford, Sulochana Gadgil, Sajani Surendran, T. N. Krishnamurti, C. M. Kishtawal, Zhan Zhang, Timothy LaRow, David Bachiochi, Eric Williford, Sulochana Gadgil, and Sajani Surendran, "Multimodel Ensemble Forecasts for Weather and Seasonal Climate," *Journal of Climate*, vol. 13, no. 23, pp. 4196–4216, 2000.

11. Ross Keith, Stephen M. Leyton, Ross Keith, and Stephen M. Leyton, "An Experiment to Measure the Value of Statistical Probability Forecasts for Airports," *Weather and Forecasting*, vol. 22, no. 4, pp. 928–935, 2007.

12. James W Taylor and Roberto Buizza, "Using weather ensemble predictions in electricity demand forecasting," *International Journal of Forecasting*, vol. 19, no. 1, pp. 57–70, 2003.

13. Luciana Bertotti, Jean-Raymond Bidlot, Roberto Buizza, Luigi Cavaleri, and Martin Janousek, "Deterministic and ensemble-based prediction of adriatic sea sirocco storms leading to 'acqua alta'in venice," *Quarterly Journal of the Royal Meteorological Society*, vol. 137, no. 659, pp. 1446–1466, 2011.

14. Nicholas M. Leonardo, Brian A. Colle, Nicholas M. Leonardo, and Brian A. Colle, "Verification of Multimodel Ensemble Forecasts of North Atlantic Tropical Cyclones," *Weather and Forecasting*, vol. 32, no. 6, pp. 2083–2101, 2017.

15. WMO, *Guidelines on Ensemble Prediction Systems and Forecasting*, World Meteorological Organization Geneva, 2012.

16. William Lowrie, *Fundamentals of geophysics*, Cambridge University Press, 2007.

17. Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing and Applications*, vol. 13, no. 2, pp. 112–122, 2004.

18. J. Wang, P. Balaprakash, and R. Kotamarthi, "Fast domain-aware neural network emulation of a planetary boundary layer parameterization in a numerical weather forecast model," *Geoscientific Model Development*, vol. 12, no. 10, pp. 4261–4274, 2019.

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

20. Yoshua Bengio, Patrice Simard, and Paolo Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

21. Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26-03-2020	Master's Thesis	SEP 2018 - MAR 2020

4. TITLE AND SUBTITLE

Predicting Upper Atmospheric Weather Conditions Utilizing Long-Short Term Memory Neural Networks for Aircraft Fuel Efficiency

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Alarcon, Garrett, A. 1st Lt, U.S. Air Force

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-20-M-129

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Aviation fuel is a major component of the Air Force (AF) budget, and vital for the core mission of the AF. This study investigated the viability of LSTMs to increase the accuracy of deterministic NWP models, while also investigating the ability to reduce model generation time. Increased forecast accuracy for wind speeds could be implemented into existing flight path models to further increase fuel eciency, while reduced modeling times would allow flight planners to generate a flight plan in rapid response situations. The most viable model consisted of an ensemble of six LSTMs trained o six coordinates. The model's error was on average +1.2 m/s higher than the deterministic NWP with a computation time of 1.85 s. The LSTM generated a flight path that was on average 14.2 min slower for an approximately 7 hour 32 min flight. This forecast generation took seconds to complete compared to hours from the deterministic model. While the LSTM architecture in this study was not able to increase forecast accuracy, the speed at which it generates an approximately close forecast can be an integral tool for flight planners in the future.

15. SUBJECT TERMS

AI, Neural Networks, LSTM, Aircraft, Flight Planning, Fuel Efficiency, Weather Models, Wind Speeds, Forecasting

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Lt Col Andrew J. Geyer, Ph.D., AFIT/ENS
U	U	U	UU	90	19b. TELEPHONE NUMBER *(include area code)* (312) 785-3636, x4584; andrew.geyer@afit.edu