STUDENT SUMMER INTERNSHIP TECHNICAL REPORT

Processing of Time Series Data in Support of Producing a Climatological Summary of WIPP

DOE-FIU SCIENCE & TECHNOLOGY
WORKFORCE DEVELOPMENT PROGRAM

Date submitted:

December 4, 2020

Principal Investigators:

Gisselle Gutierrez-Zuniga, E.I. (DOE Fellow Student)
Florida International University

Dr. Anderson L. Ward (Mentor)
DOE Carlsbad Field Office

Ravi Gudavalli Ph.D. (Program Manager)
Florida International University

Leonel Lagos Ph.D., PMP® (Program Director)
Florida International University

Submitted to:

U.S. Department of Energy
Office of Environmental Management
Under Cooperative Agreement # DE-EM0000598

FIU Applied Research Center
FLORIDA INTERNATIONAL UNIVERSITY

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, nor any of its contractors, subcontractors, nor their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe upon privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any other agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

ABSTRACT

This report provides an overview of the remote summer internship done by Gisselle Gutierrez-Zuniga with the U.S. Department of Energy's Carlsbad Field Office (CBFO) under the mentorship of Dr. Anderson L. Ward. Over the ten weeks of the internship, Gisselle worked on the first phase of a project aimed at developing a web-accessible public database for meteorological data for the Waste Isolation Pilot Plant (WIPP). Tasks included the development of a Python program to automatically process archived data dating back to the 1970s, parsing metadata, substituting National Oceanographic and Atmospheric Administration (NOAA) data for missing data, and generating ASCII files of observations for storage. Gisselle used this database to calculate statistics, including normals and return intervals, which will be the basis of a Climatological Summary Report for WIPP, of which she will be co-author. This will also serve as a reference for the CBFO and DOE of the historical monthly and annual data of WIPP from its start to present. This program has already found use in automated quality checks of monthly datasets to detect anomalies and missing data.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In the 1940s, transuranic (TRU) waste began accumulating as a result of the start of the nation's nuclear defense program. The federal government was uncertain of where to dispose the TRU waste and thus began the search for an ideal location. In 1956, the National Academy of Sciences recommended salt formations as a suitable medium to permanently dispose of radioactive wastes. Throughout the 1960s and into the 1970s, scientists searched throughout the country for the perfect location and ultimately tested a remote desert area in New Mexico in 1974, where a 2,000 ft thick salt bed was located. Ultimately, in 1979, U.S. Congress approved the Waste Isolation Pilot Plant (WIPP) which still serves as the nation's only deep geologic repository for the disposal of TRU waste.

U.S. Department of Energy's (DOE) owned WIPP is licensed to store TRU waste for 10,000 years and because of this, it is important to monitor any significant climate changes that can potentially impact its integrity and performance. WIPP is located 26 miles (42 km) east of Carlsbad, New Mexico, situated in the northern reaches of the Chihuahuan Desert, in the lower Pecos River Valley. Atmospheric emission is believed to be the only plausible pathway for radionuclide transport during the receipt and emplacement of transuranic waste at WIPP. Regulators require that atmospheric dispersion and dose models be used to predict atmospheric concentration of stack emissions and estimate airborne radionuclide exposures within 80 km of WIPP. The best method to quantify uncertainty in these models is to compare their predictions with environmental and meteorological measurements under conditions similar to those assumed by the model. For this reason, the EPA has established guidelines for the collection and processing of primary meteorological variables (wind direction, wind speed, temperature, humidity, pressure, and radiation) and for processing of derived meteorological variables such as stability, mixing height, and turbulence at nuclear facilities (USEPA, 2000). Over years, much of the early data collected during site selection and characterization at WIPP have been lost and exist only as hard copies in archived reports. Data continue to be collected 24 hours a day, 7 days and week and must be analyzed to identify anomalies and missing data, and to detect instrument malfunctions.

The goal of this internship is to create an automated approach to process archived data, parse metadata, substitute NOAA data for missing data, and generate ASCII files of observations for storage. The database will then be used to calculate statistics and ultimately create a Climatological Summary Report. It will also serve as a reference for the CBFO and DOE of the historical monthly and annual data of WIPP from its start to present.

2. EXECUTIVE SUMMARY

This research work has been supported by the DOE-FIU Science & Technology Workforce Initiative, an innovative program developed by the US Department of Energy's Environmental Management (DOE-EM) and Florida International University's Applied Research Center (FIU-ARC). During the summer of 2020, a DOE Fellow intern Gisselle Gutierrez-Zuniga spent 10 weeks doing a remote summer internship for the DOE Carlsbad Field Office under the supervision and guidance of Dr. Anderson L. Ward, Site Specialist. The intern's project was initiated on June 15, 2020 and continued through August 21, 2020 with the objective of automatically processing archived data dating back to the 1970s, parsing metadata, substituting NOAA data for missing data, and generating ASCII files of observations for storage. This data will be the basis of a Climatological Summary Report for WIPP.

3. RESEARCH DESCRIPTION

Development of a Python Package

In order to develop the Python package, the following packages were downloaded:

- Numpy
- Plotly.express
- Matplotlib
- pandas

The first step was to read in the available monthly files. In this case, the year 2019 was used for creating the python package. After all twelve files were read in,, any missing values were identified and replaced with “NaN” in order to allow for proper computations.

```
# Loop over filenames to import data
#filenames = glob(strpath + '*.csv') # file names have to be sorted correctly..
filenames = [strpath + 'Jan_2019.csv',
             strpath + 'Feb_2019.csv',
             strpath + 'Mar_2019.csv',
             strpath + 'Apr_2019.csv',
             strpath + 'May_2019.csv',
             strpath + 'Jun_2019.csv',
             strpath + 'Jul_2019.csv',
             strpath + 'Aug_2019.csv',
             strpath + 'Sep_2019.csv',
             strpath + 'Oct_2019.csv',
             strpath + 'Nov_2019.csv',
             strpath + 'Dec_2019.csv']
```

Figure 1. Python code detailing the read-in of all monthly data.

```
# Check for missing data
print (df_jan.isnull().sum())
print (df_feb.isnull().sum())
print (df_mar.isnull().sum())
print (df_apr.isnull().sum())
print (df_may.isnull().sum())
print (df_jun.isnull().sum())
print (df_jul.isnull().sum())
print (df_aug.isnull().sum())
print (df_sep.isnull().sum())
print (df_oct.isnull().sum())
print (df_nov.isnull().sum())
print (df_dec.isnull().sum())
```

Figure 2. Python code identifying any missing values in the monthly data.

Following the replacement of the missing data, all monthly files were combined into one annual file and each month was separated into its own data frame for the calculations. The variables of interest for the calculations were wind speed (at 2 meters (m), 10 m, and 50 m), wind direction (at 2 m, 10 m, 50 m), temperature (at 2 m, 10 m, 50 m), relative humidity, dew point temperature, solar radiation, barometric pressure, and precipitation. Each variable was resampled hourly, daily, monthly, and annually for the minimum, maximum, and average values. The purpose of resampling was to aggregate the time series data by a new time period (e.g. daily to monthly).

```
# Resample Wind speed @ 2 m:
WS2m = df_jan['WS2m']
WS_2m_h_min = WS2m.resample('1H').min() #hourly(min)
WS_2m_h_max = WS2m.resample('1H').max() #hourly(max)
WS_2m_h_avg = WS2m.resample('1H').mean() #hourly(avg)

WS_2m_d_min = WS2m.resample('1D').min() #daily(min)
WS_2m_d_max = WS2m.resample('1D').max() #daily(max)
WS_2m_d_avg = WS2m.resample('1D').mean() #daily(avg)

WS_2m_m_min = WS2m.resample('1M').mean() #monthly(min)
WS_2m_m_max = WS2m.resample('1M').mean() #monthly(max)
WS_2m_m_avg = WS2m.resample('1M').mean() #monthly(avg)

WS_2m_a_min = WS2m.resample('1A').min() #annually(min)
WS_2m_a_max = WS2m.resample('1A').max() #annually(max)
WS_2m_a_avg = WS2m.resample('1A').mean() #annually(avg)
```

Figure 3. Resampling of one of the variables of interests, wind speed at 2 m elevation for the month of January 2019.

Development of Plots and Wind Roses

Once all the variables were resampled, the wind roses using the wind direction and wind speeds at 2 m, 10 m, and 50 m were created. In order to create a wind rose, bins were created for wind velocity and wind direction in order to convert the values to cardinal directions (N, NNE, NE, ENE, E, ESE, SE, SSE, S, SSW, SW, WSW, W, WNW, NW, and NNW). Next, a new data frame with all columns was created and an additional column for the frequencies, followed by the grouping of velocity and direction. Finally, all remaining "Norths" were replaced with "N" and the wind rose for 2 m, 10 m, and 50 m was plotted.

```
df = df_wr50m

bins_mag= [0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0]
bins_mag_labels = ['0.0-1.0','1.0-2.0','2.0-3.0', '3.0-4.0','4.0-5.0','5.0-6.0']

bins_dir = [0., 22.5, 45.0, 67.5, 90.0, 112.5, 135.0, 157.5, 180.0, 202.5, 225.0, 247.5, 270.0, 292.5, 315.0, 337.5, 360.0]
bins_dir_labels = ['N','NNE','NE','ENE','E','ESE','SE','SSE','S','SSW','SW','WSW','W','WNW','NW','NNW']

df['mag_binned'] = pd.cut(df['WS50m'], bins_mag, labels = bins_mag_labels)
df['dir_binned'] = pd.cut(df['WD50m'],bins_dir, labels=bins_dir_labels)

# Create new dataframe with all columns and an extra one for frequencies
dfe = df[['mag_binned', 'dir_binned','WS50m']].copy()

# Change name of the last column to frequencies
dfe.rename(columns={'WS50m': 'frequency'}, inplace=True)

# Group by velocity and direction
g = dfe.groupby(['mag_binned','dir_binned']).count() #grouping
g.reset_index(inplace=True)
g['percentage'] = g['frequency']/g['frequency'].sum()
g['percentage%'] = g['percentage']*100
g['Jan 50m Magnitude [m/s]'] = g['mag_binned']

# Replace any remaining Norths with N
g = g.replace(r'North', 'N', regex=True) #replacing remaining Norths with N
print(g)

# Write out data
d = []
d.append(g)
df = pd.DataFrame(g)
df.to_excel("windrose_Jan2019_50m.xlsx")

# Plot the wind rose
fig = px.bar_polar(g, r="percentage%", theta="dir_binned",
                   color="Jan 50m Magnitude [m/s]",template="plotly_dark",
                   color_discrete_sequence= px.colors.sequential.Plasma_r)

fig.show(renderer="svg")
fig.write_image("windrose_Jan2019_50m.svg")
```

Figure 4. Python code creating 50 m wind rose using wind direction (in cardinal directions) and wind velocity for the month of January 2019.

Additionally, all the variable's hourly, daily, and monthly data was plotted by a line graph.

```
#plotting first set of subplots:
#square up dates for x axis
h_dates = pd.to_datetime(T2m_h_avg.index)
d_dates = pd.to_datetime(T2m_d_avg.index)
m_dates = pd.to_datetime(T2m_m_avg.index)

fig,ax = plt.subplots(2, 1,figsize=(8.5,5.5))

# ###########################
# #Wind Speed 2m subplot #
# ###########################
l1a, = ax[0].plot(h_dates,WS_2m_h_avg,c='lightgray',label='hourly',zorder=0)
l2a, = ax[0].plot(d_dates,WS_2m_d_avg,c='dimgray',label='daily',zorder=1)
l3a = ax[0].scatter(m_dates,WS_2m_m_avg,marker='o',s=75,c='r',label='monthly',zorder=2)
legend0 = ax[0].legend(handles=[l1a,l2a,l3a],ncol=3,frameon=True,loc=4)
ax[0].set_ylabel('2 m Wind Speed. [m/s]')
ax[0].set_xlim([h_dates[0],h_dates[-1]])
ax[0].set_ylim([0.,20.])
ax[0].tick_params(axis='x',labelrotation=45)

# ###########################
# #Wind Direction 2m subplot #
# ###########################
l4a, = ax[1].plot(h_dates,WD_2m_h_avg,c='lightgray',label='hourly',zorder=0)
l5a, = ax[1].plot(d_dates,WD_2m_d_avg,c='dimgray',label='daily',zorder=1)
l6a = ax[1].scatter(m_dates,WD_2m_m_avg,marker='o',s=75,c='r',label='monthly',zorder=2)
legend1 = ax[1].legend(handles=[l1a,l2a,l3a],ncol=3,frameon=True,loc=4)
ax[1].set_ylabel('2 m Wind Direction. [deg]')
ax[1].set_xlim([h_dates[0],h_dates[-1]])
ax[1].set_ylim([0.,360.])
ax[1].tick_params(axis='x',labelrotation=45)


plt.tight_layout()

plt.savefig('Jan_fig1.png',dpi=600)
plt.show()
```

Figure 5. Portion of Python code creating subplots of wind speed and wind direction at 2 m elevation for January of 2019.

The final step was to create a climate chart using the Matplotlib package. To create a climate chart, a boxplot was first created using the minimum, maximum, and average values for the temperature data. The rectangle boxes were made by plotting the maximum temperature as the upper boundary limit and the minimum temperature as the lower boundary of the rectangle. The climate chart displayed the maximum and minimum temperature values for each month in 2019. Once this was done, the same code was applied to the remaining months of 2019. This code will ultimately be used for all the available years of the WIPP site and be used towards the development of a Climatological Report of the WIPP.

```
######################
#  Plot Temperature   #
######################
n = 12 #Number of data points, 12 mo in this case
xdata = np.arange(0, n, 1)
xe = np.ones(n)
ydata = Tavg

# Create figure and axes
fig, ax1 = plt.subplots()

# Create box plot from Tmin, Tmean, Tmax data
xe_lo = xe * 0.40
xe_hi = xe * 0.40
ye_lo = Tavg - Tmin
ye_hi = Tmax - Tmin

def make_temp_boxes(ax1, xdata, ydata, xelo, xehi, yelo, yehi,
                     facecolor='r',edgecolor='k', alpha=0.5):

    temp_boxes = [Rectangle((x-xlo, y-ylo),2*xlo, yhi)
           for x, y, xlo, xhi, ylo, yhi in zip(xdata, ydata, xelo, xehi, yelo, yehi)]

    # Create patch collection with specified color/alpha
    pc = PatchCollection(temp_boxes, facecolor=facecolor, alpha=alpha,
                          edgecolor=edgecolor)

    # Add collection to axes
    ax1.add_collection(pc)
#
    # Plot Temperature box using error bar command
    artists = ax1.errorbar(xdata, ydata, xerrl=xe_lo,xerrh=xe_hi, yerrl=ye_lo,
                           yerrh=ye_hi,fmt='None', ecolor='r')
    return artists

# Call function to create Temperature boxes
_ = make_temp_boxes(ax1, xdata, ydata, xe_lo, xe_hi, ye_lo, ye_hi)

# Make the y-axis label
#ax1.set_xlim(0, 11)
ax1.set_ylim(-30,120)
ax1.set_yticks([]) #Turn off LHS y-axis labels
#ax1.set_ylabel('Temperature ($^o$F)',fontsize='12') # LHS Y-axis label
#ax1.set_xlabel('Month', color='r',fontsize='12')

# Annotate temperature boxes with maximum temperatures
for i, v in enumerate(Tmax):
    plt.text(xdata[i] - 0.2, v + 3, str(v),fontsize='12',color='r')
```

Figure 6. Code creating climate chart using Matplotlib for January 2019.

4. RESULTS AND ANALYSIS

Once the code was run, the following figures were obtained for the month of January 2019. The variables of interest (wind speed, wind direction, temperature, relative humidity, dew point temperature, solar radiation, barometric pressure, and precipitation) were all successfully plotted as well as the wind roses for the 2 m, 10 m, and 50 m elevations. The most challenging charts to create were the wind roses and the climate chart due to the fact that there was limited information on creating one, but after several trials and errors, they were created.

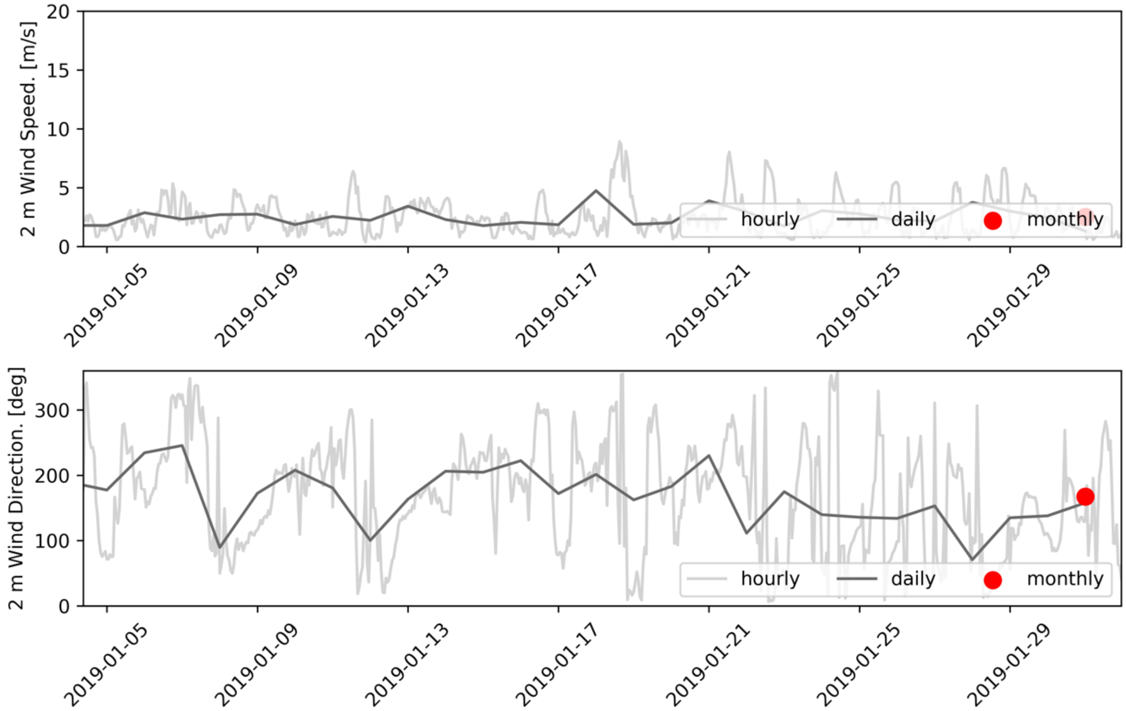

Figure 7. Subplots of 2 m Wind Speed and Wind Direction in January 2019.

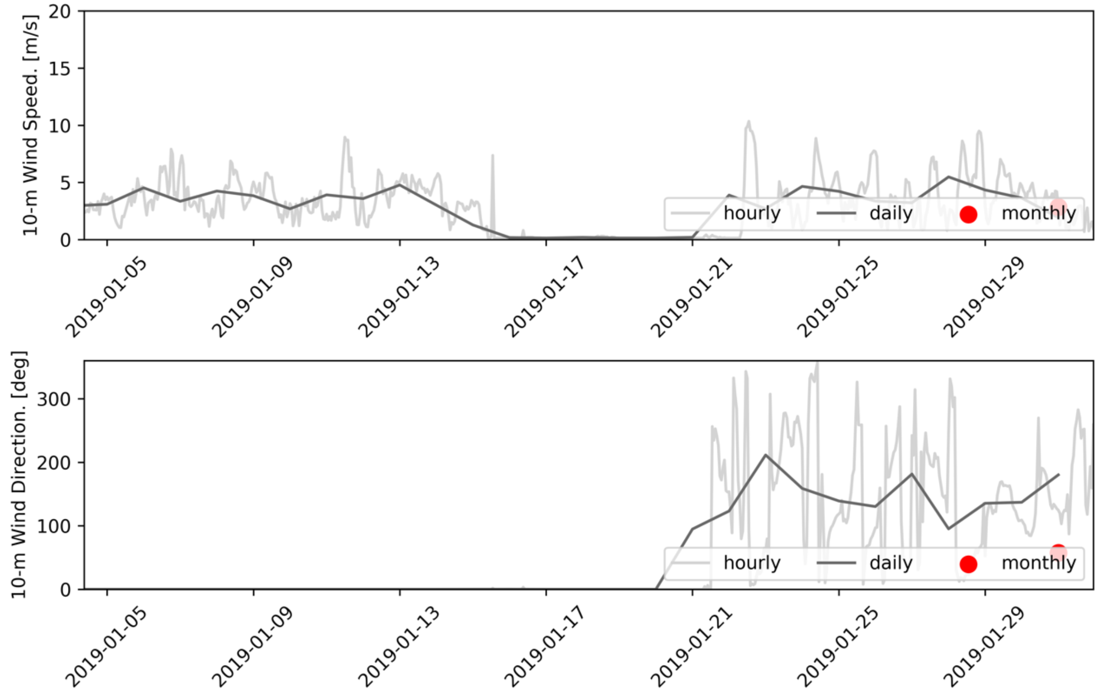

Figure 8. Subplots of 10 m Wind Speed and Wind Direction in January 2019.

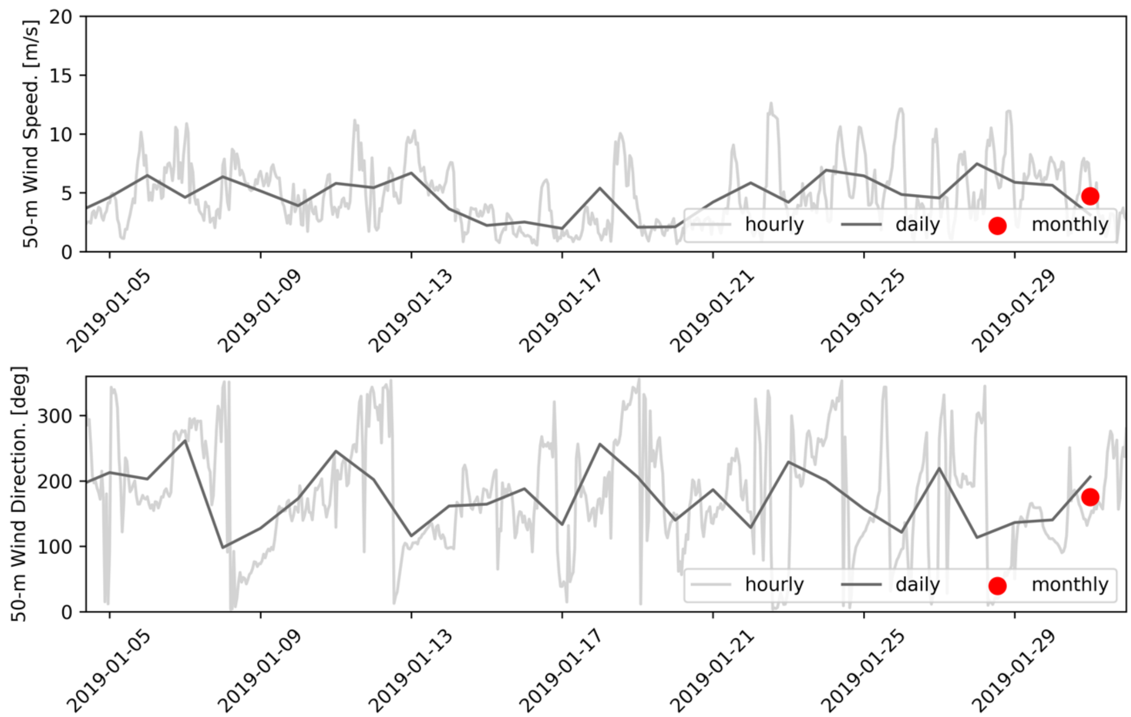

Figure 9. Subplots of 50 m Wind Speed and Wind Direction in January 2019.

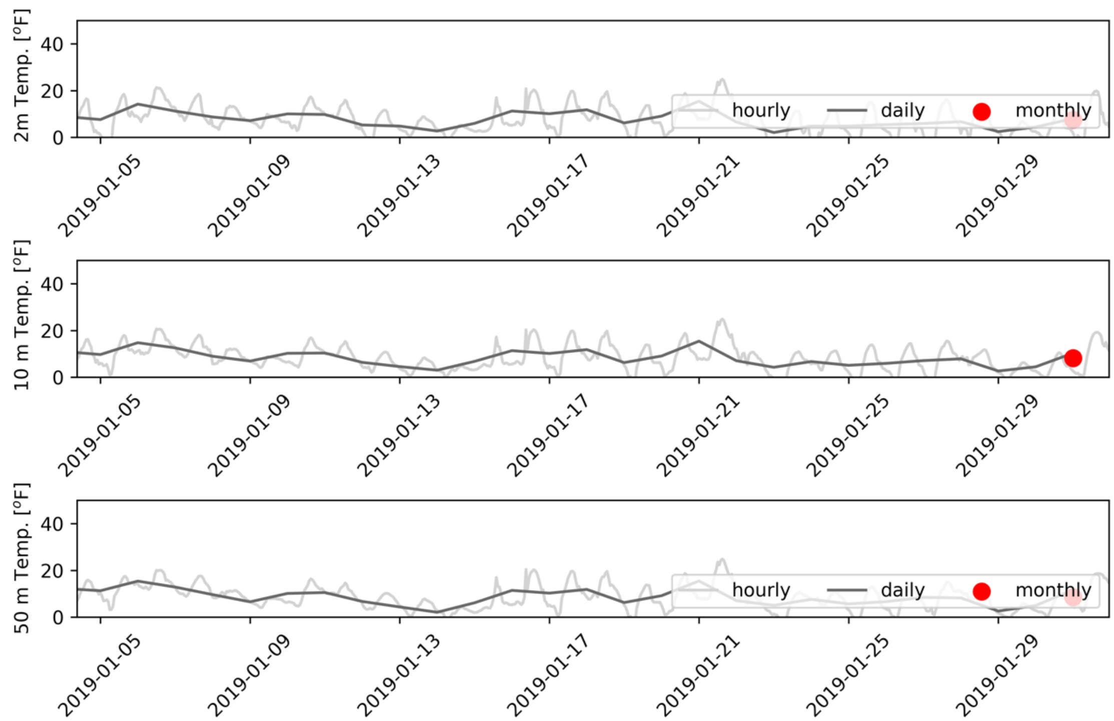

Figure 10. Subplots of 2 m, 10 m, and 50 m Temperature in January 2019.

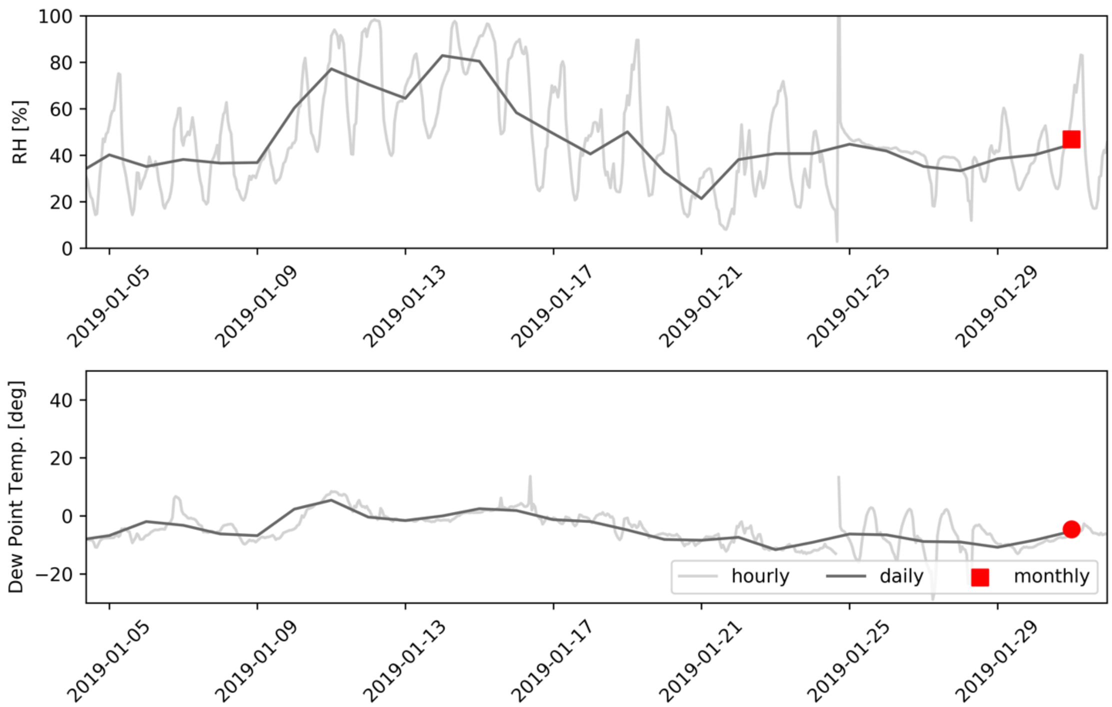

Figure 11. Subplots of Relative Humidity and Dew Point Temperature in January 2019.

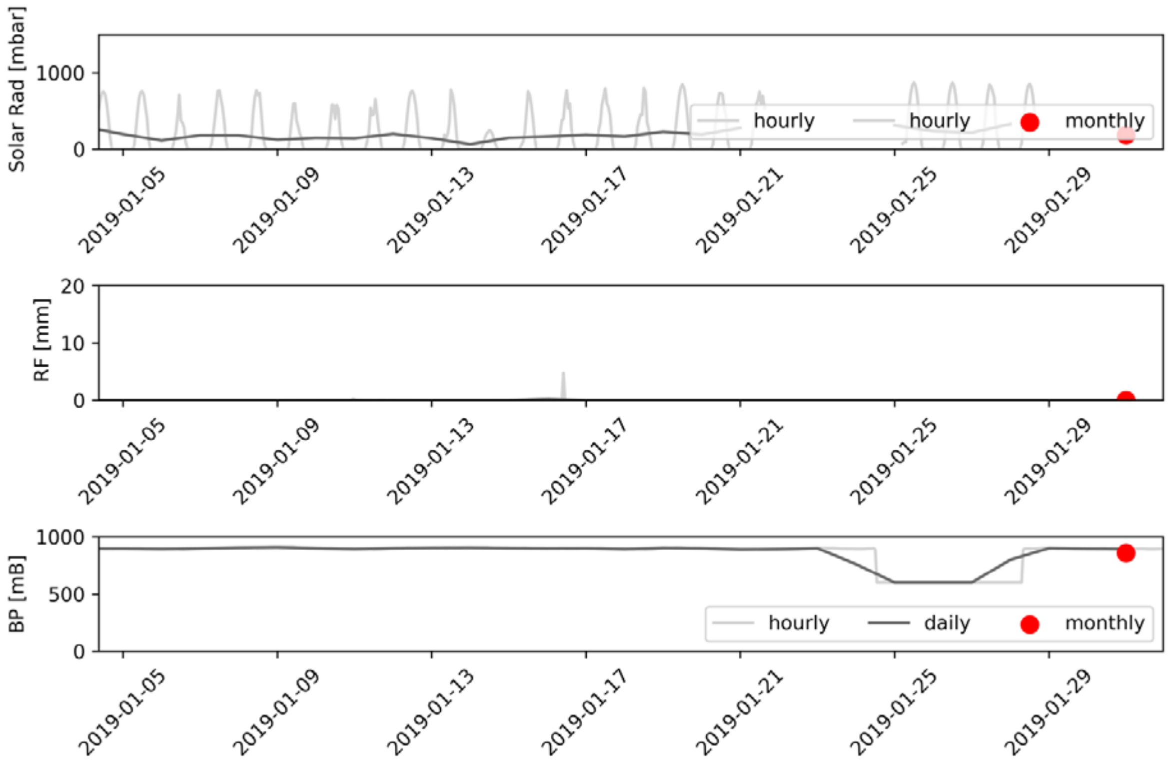

Figure 12. Subplots of Solar Radiation, Rainfall, and Barometric Pressure in January 2019.

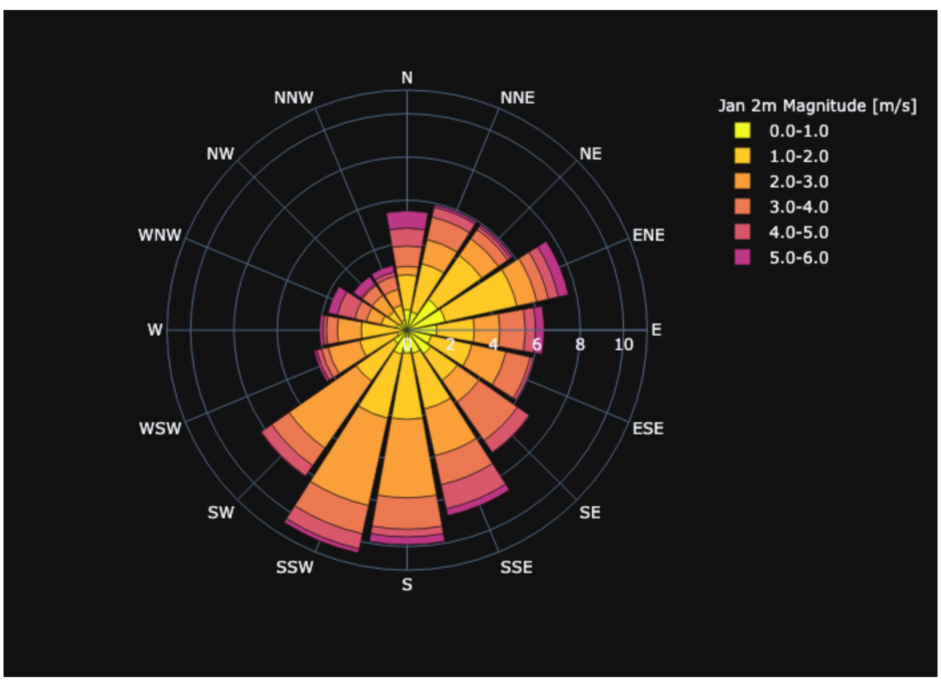

Figure 13. Wind rose of 2 m wind speed and wind direction in January 2019.

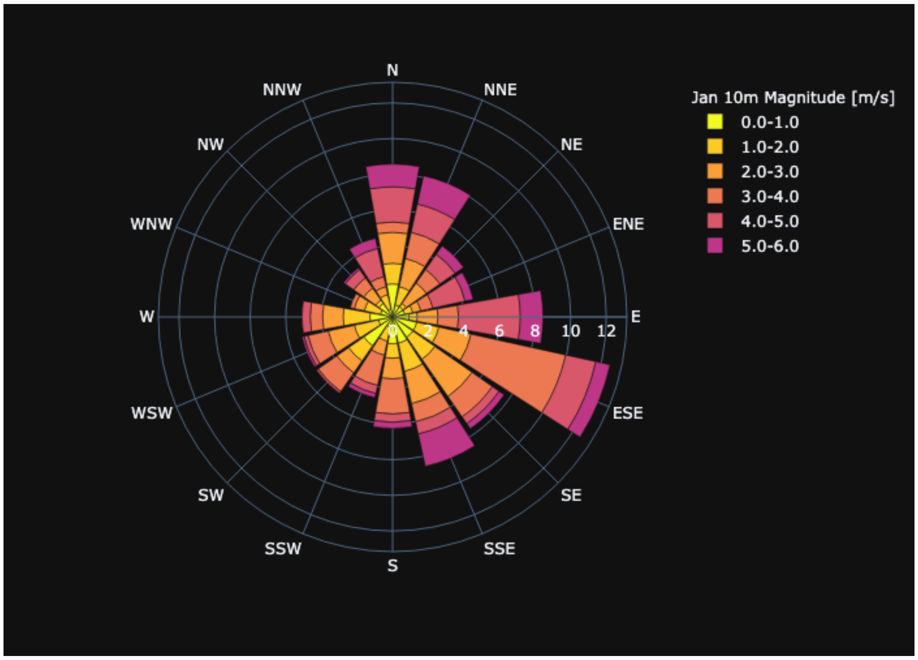

Figure 14. Wind rose of 10 m wind speed and wind direction in January 2019.

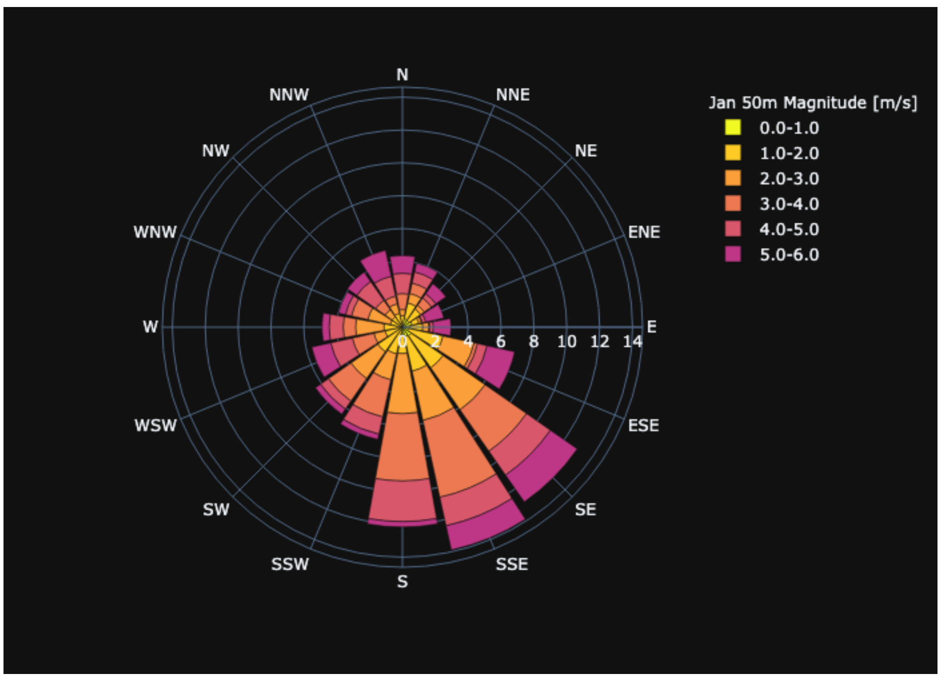

Figure 15. Wind rose of 10 m wind speed and wind direction in January 2019.

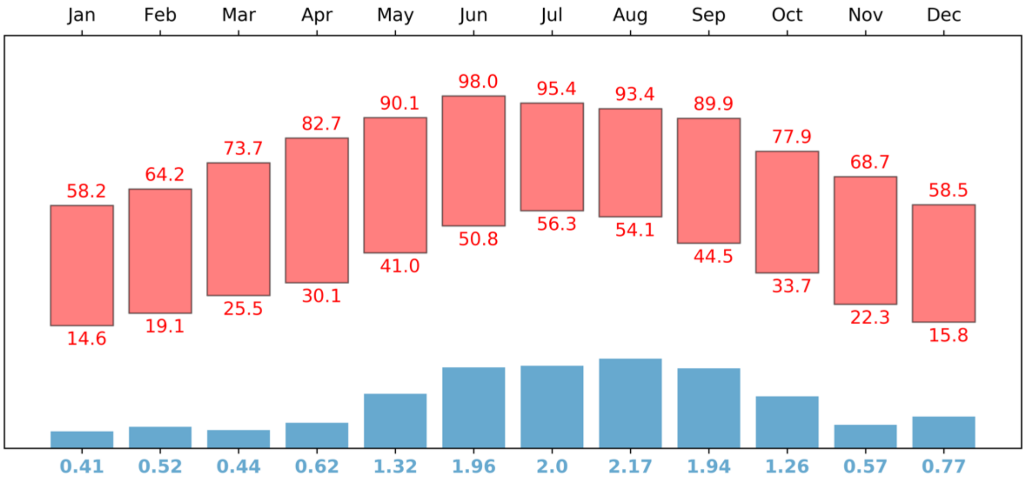

Figure 16. 30-year normal climograph for the WIPP site showing precipitation in inches (blue bars), and temperature, including minima and maxima (red bars).

5. CONCLUSION

The goal of this internship was to create an automated approach to process archived data, parse metadata, substitute available NOAA data for missing data, and generate ASCII files of observations for storage. This will then be used as a database to calculate statistics and create a Climatological Summary Report for the WIPP, which will ultimately serve as a future reference for the CBFO and DOE of the historical monthly and annual data of the WIPP from its start to present. During the time of this internship, a Python package was developed to process monthly data, join monthly data for a year, deal with missing values, and plot the variables of interest. This code will be applied to other available annual data and serve as a reference for the DOE CBFO of the climatological historical data of the WIPP. Future work of this project will consist of further improving the Python code and developing a database with all available years to ultimately create a Climatological Summary Report.

6. REFERENCES

1. U.S. Department of Energy. (n.d.). *Why WIPP* [Factsheet]. https://www.wipp.energy.gov/fctshts/Why_WIPP.pdf
2. U.S. Department of Energy. (n.d.). *WIPP Chronology* [Factsheet]. https://www.wipp.energy.gov/fctshts/Chronology.pdf
3. U.S. Environmental Protection Agency. 2000. Meteorological Monitoring Guidance for Regulatory Modeling Applications (EPA-454/R-99-005).