UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

RWIS BASED ROAD CONDITION PREDICTION USING MACHINE LEARNING ALGORITHMS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

By

MOHAMED E AFIFY
Norman, Oklahoma
2020

RWIS BASED ROAD CONDITION PREDICITION USING MACHINE LEARNING ALGORITHMS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Hazem H. Refai, Chair

Dr. Choon Yik Tang

Dr. Samuel Cheng

Dedication

To my mom and dad,
Eman Labib & Essam Afify
My beloved family
&
Friends.

ALL PRAISE IS TO THE ONE AND ONLY GOD.

Acknowledgements

Foremost, I would like to express my genuine gratitude and sincere thanks to my advisor Dr. Hazem Refai for his support and guidance. Besides my advisor, I would like to state my appreciation to my committee members—*Dr. Samuel Cheng* and *Dr. Choon Yik Tang.* I would also like to thank Michelle Farabough for helping me edit this thesis. Many thanks to my colleagues who participated in this research, especially: Siraj Muhammad, Mohamat Irban bin Ali Kaja Najumudeen, and Abdallah Mousselli. Finally, I want to recognize the support and great love of my family and friends.

Table of Contents

List of Tables

List of Figures

Abstract

The need for a forecasting model of road conditions is becoming evermore critical, given the effects of ever-increasing severity in weather. Drastic changes in weather, especially cold fronts, often lead to dangerous roads. Consequently, traffic efficiencies are diminished and, even worse, accidents resulting in loss of life and property could increase. Across the nation, states are responsible for anticipating inclement weather and treating roads accordingly. Treatment costs can be reduced with more precise road condition predictions. The development of machine learning capabilities has enhanced the utilization of Big Data Systems throughout various sectors, including road climatology, making weather forecasting much more efficient and reliable.

The study reported in this thesis analyzed various road climatology data, including sub-surface temperature at two- and six-inches from Road and Weather Information Systems (RWIS) deployed by Oklahoma Department of Transportation (ODOT) along the I-35 corridor at various road-bridge intersections aimed at producing a reliable and robust forecast model for predicting road surface temperature in the near and distant future. The predicting importance of each factor is analyzed statistically, and then manually, to determine its requirements for the forecast model. The study also determined the best forecast model after comparing a newly developed neural network with common regression techniques previously available through Machine Learning. Results showed that the novel neural network model offered a reliable 12-hour prediction for road surface temperature at a frequency of five minutes, depending on available historical data from RWIS. Two additional classification models provided Road Conditions Classes. The first was based on time series historical data from RWIS, and the second was based on historical and future data from a GFS (Global Forecast System). Together, these models accurately forecast local road surface

temperatures 12-hour in advance of inclement weather in five-minutes frequencies at RMSE of ±1.67. They also accurately classified road conditions at a rate of more than 87.984%.

1 Introduction

Predicting the road surface temperature during the coldest months is essential for determining road conditions and the likelihood of ice formation. According to United States Department of Transportation (USDOT) data for a 10year average, icy pavement was the cause for 156,164 crashes and 521 fatalities in the United States [1]. During the mid-February deep-freeze, billion-dollar disaster of 2021, at least 125 people died as a direct or indirect result of arctic weather [2]. Roadway ice fatalities far outnumber those resulting from severe tornadoes. Winter cold fronts drop temperatures below freezing, causing freezing rain and sleet. As rain falls on surfaces with below-freezing temperatures, roadways form an ice coating that presents a significant hazard to the vehicles. Weather forecasts can be misleading, as a cold front brings a sharp change in temperature—from warm to freezing levels. Although typical forecasting models will predict snowfall, the snow will melt once it reaches the road surface because the roadway does not immediately react to the sudden drop of ambient temperature. Rather, the decrease is sublinear. Hence, if a model is trained using historical surface temperature data along with additional meteorological features, the model can learn how surface temperature (ST) will react to sudden changes in ambient temperatures, and then provide accurate predictions for dangerous road travel due to freezing rain, sleet, and snow fall. Such a model will predict BST separately, as bridges become icy before roadways due to more surface area exchanging energy with the atmosphere and more rapid cool down [3]. RST and BST prediction will assist ODOT with making more cost-effective salting decisions. Because it takes several hours to salt a highway, the proposed model's advance predictions will provide good indication of road surface temperature and aid road maintenance planners in determining the best time to initiate salting the roads.

The proposed models predict RST and BST for the next 12 hours using developed forecasting models that integrate regression kernels and neural networks. Two additional models are used to predict road conditions. The first uses Window Forecasting to predict the next six hours, and a the second makes a 12-hour prediction using a pipeline of GFS forecasting model and the regression model detailed in this thesis. This automated model will send an HTTP request to parse future data from an online server and update RST and BST conditions for the next 12 hours. The GFS forecasting model is parsed by ClimaCell API service, which provides the model developed for this thesis with weather information for next 12 hours at a frequency of one hour.

Various regression models were tested, including a deep neural network regression learning model for predicting road surface temperature based on collected meteorological data from stations deployed by ODOT along the I-35 highway corridor at road-bridge intersections. Models were validated on historical data collected during winter storms and from online data collected by ClimaCell API online server. The work reported here is investigated various models; these are briefly described below. Performance results for each model are based on Root Mean Square Error (RSME). The same model was tested on different stations to validate performance in different locations based on an Ensample model of combined regression models, which included a pipeline of information as part of an international model for the I-35 corridor.

1.1 Thesis Objective

This thesis was developed with the objective of furthering the utilization of RWIS real-time weather data. Analyses focused on road conditions and data forecasting collected during the winter from October 2020 to late March 2021. The analysis is divided into two sections. The first forecasts both the BST and RST temperature for indicating if the road surface temperature is below the freezing level at 32 °F. The second is a classification approach to validate road condition severity

based on previous incidents that occurred between January 2019 and February 2021. This work reported in this thesis explored data acquisition from RWIS system that reports information gathered along highway I-35, including 15 stations that collect the same data in real time. The analysis Data collection utilized restful API and post processed regression and classification model information for road conditions.

The main contributions of this thesis are summarized below.

[1] Describe the RWIS system—number of stations, their locations, type of systems that collect weather data, and type of collected data

[2] Determine the optimal method to prepare RWIS data for sample observations of supervised learning algorithms to achieve data cleaning, pre-processing, and data analysis

[3] Discuss the methodology to prepare data for regression and classification, as well as demonstrate the importance of data engineering to determine road condition class

[4] Train multiple models for supervised learning, neural network algorithms, ensemble models, and comparing model performance using validation equations to determine the most optimal model for road condition forecasting and classification.

As per the aforementioned stated objectives, this thesis is divided into multiple sections. Section 1 introduces the RWIS system, as well as the thesis objectives. Section 2 summarizes the background and various related works that have investigated forecasting road surface temperature. Section 3 details the RWIS system, station number, system design, and data types. Section 4 explains the exploratory data analysis performed on the RWIS data and demonstrates the way in which API acquires data after preprocessing. This section also explains how regression model data is prepared for long sliding window in post-processing and how classification model data is analyzed. Section 5 focuses on various models for regression and classification, and then compares

the proposed model's performance using model validation formulas. Section 6 summarizes the work completed for this thesis and suggests future investigations to enhance model performance.

2 Background

In this section, we will briefly summarize existing research and development in road climatology, focusing on forecast methods. Notably, although researchers have rigorously pursued climatology forecast, attention on road climatology is sparse and lacking.

The thesis focusses on the relation between road surface temperature, moisture caused by dew point and precipitation, as well as road condition. A study spearheaded by the Washington Department of Atmospheric Sciences [4] examined the correlation between the freezing level (i.e., below 32 °F) and moisture caused by frost, fog, melted snow, and freezing snow and rain. Clouds act as blankets on overcast nights, preventing heat radiation loss from the ground surface and causing snow to melt into black ice. Also, strong winds stir up the atmosphere and force the warmer air loft down the ground. As warm air starts to raise road surface temperatures above freezing, snow melts and turns into wet slush. Each of these dangerous conditions must be monitored and studied separately to enhance the quality of road condition prediction, which is an important aspect of traffic safety for all governments. Germany has developed various forecasting models and real-time radars to aid its road condition predictions. [5]. The UK Met Office offers road surface forecasts for 24-hour periods, detailing the likely amount of snowfall [6]. Most models are based on numerical analyses using a significant amount of data, atmospheric variables, and computational fluid dynamics (CFD). Many researchers have performed RST forecasting using various modeling techniques and features other than the meteorological features utilized for the research presented in this thesis. Much published literature details research about statistical models based on government forecasts and manual reports to predict road conditions. Authors in [7] used historical snow event accident data obtained from an ODOT RWIS station. The statistical study reported a high correlation between storms that had road temperature fall below freezing

during snowfall and traffic accidents. Researchers in [8] introduced a numerical model divided into 15 measurement layers that begin with above-the-ground surfaces at 3.5 meters. Using heat transfer equations, model inputs include ambient temperature, relative humidity, wind speed, wave-radiation, and precipitation. When input into the model, the features output road surface temperature, road conditions, traffic index, and surface friction. The model forecasts 24 to 48 hours depending on the amount of available input data. The model is used by the Finnish Meteorological Institute. The Korea Meteorological Administration (KMA) [9] uses a forecasting model developed by researchers at the Korea Institute of Construction and Technology for forecasting temperature from three to 24 hours using real-time data. The model is based on heat transfer between road surfaces and the atmosphere.

Machine learning models have not been implemented or tested by any meteorological institute to predict RST. Rather, heat transfer models have primarily been used to predict RST based on atmospheric temperature predicted by physical models. None of these forecasting models utilized future attributes to predict sudden drop in the temperatures. It is important to note that the state of Oklahoma is frequently exposed to sharp temperature drops following a cold front. For example, on February 21, 2018 [10] Oklahoma experienced a 21-degree drop in temperature in a four-minute time period. followed by severe freezing rainfall that affected major highways. Such sudden drops can misinform a model by giving inaccurate RST estimations, as shown in Figure 1-1. Drop in ambient temperature (AT) did not affect RST and BST. Ambient temperature remained at a freezing temperature, while RST and BST were almost 8°F above freezing. The proposed model can predict the difference between the BST and RST primarily because a bridge always freezes before the roadway, as explained in section I.

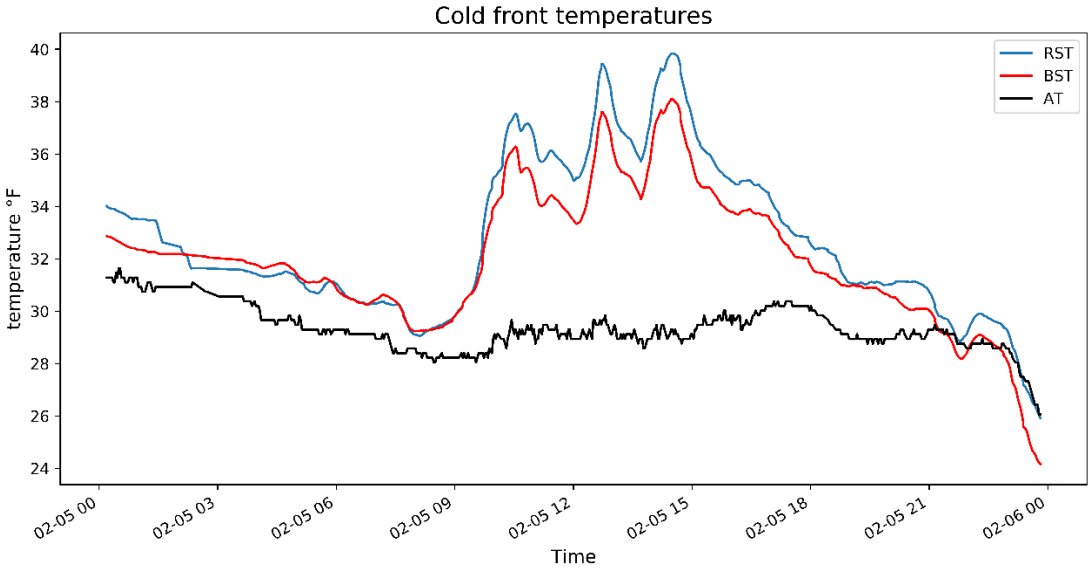


Figure 2-1. Cold-front temperature values on February 5, 2020.

3 Road Weather Information System (RWIS)

The ODOT Road Weather Information System (RWIS) is an automated with a communication system for data transfer. The station measures real-time data, some of which is used by the United States National Weather Service (NWS) to assist with forecasting in specific regions.

RWIS was designed and developed for ODOT by Dr. Hazem Refai's research team at OU-Tulsa. The system collects specific weather data for monitoring highway road weather conditions. Fifteen stations are deployed along 209 miles of the Oklahoma's I-35 corridor, starting from Tonkawa city in Kay County and ending at Thackerville in Love County. The distance between each station is approximately 16.5 miles (See Figure 3-1). Each station has multiple sensors for collecting various data..

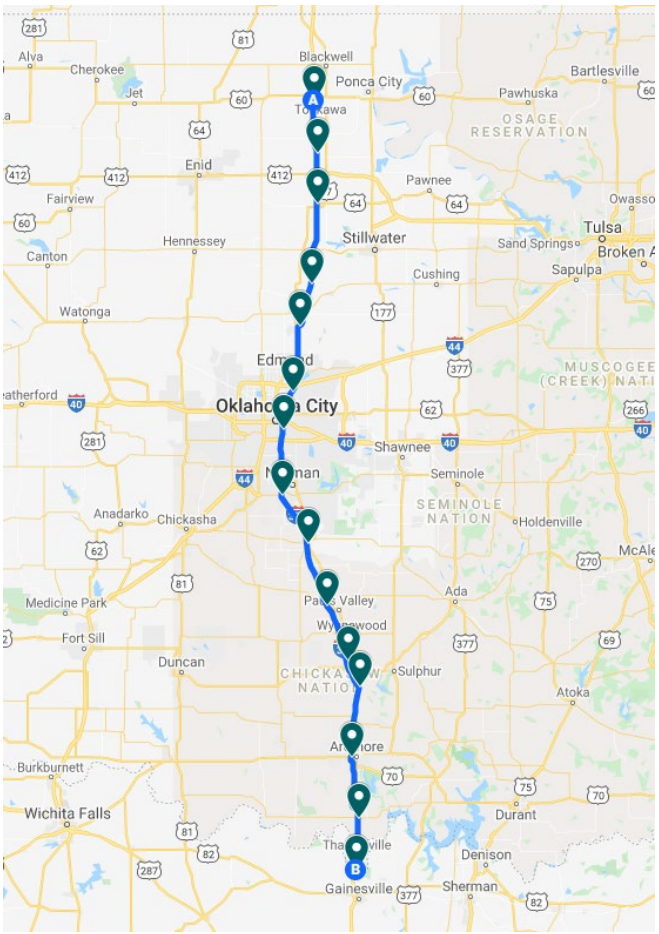


Figure 3-1. RWIS locations.

Figure 3-2 shows RWIS station locations with six components on the mast. The first component is identified as THESIS CLIMA [11] and collects weather data, including wind speed and direction, gust-wind speed and direction, temperature, humidity, air pressure, brightness and brightness direction, precipitation event, as well as precipitation intensity and type. The second component houses the Surface Sentinel sensors [12], which collects surface temperature. One sensor collects road surface temperature, and another collects the bridge Surface temperature. The third component houses two sub-surface temperature probes [13]: one located two inches below the surface level and the other six inches below the surface. The fourth component is a live camera that records road conditions in real time and archives a photo in the system every five minutes. The fifth component is a cabinet that controls the system by collecting data and sending it to the server. Figure 3-3 shows the cabinet schematic, including the power controller; dataloggers for controlling each system and parsing data from it; and the Roadside Embedded Extensible Computing Equipment (REECE) [14]—a Linux-based embedded system for collecting data and sending it to the server via a Sierra cellular gateway [15] to a cloud server. The final component is the solar-based power circuit (See Figure 3-4), which has two 100W solar panels, two 100 AH batteries, and a controller.

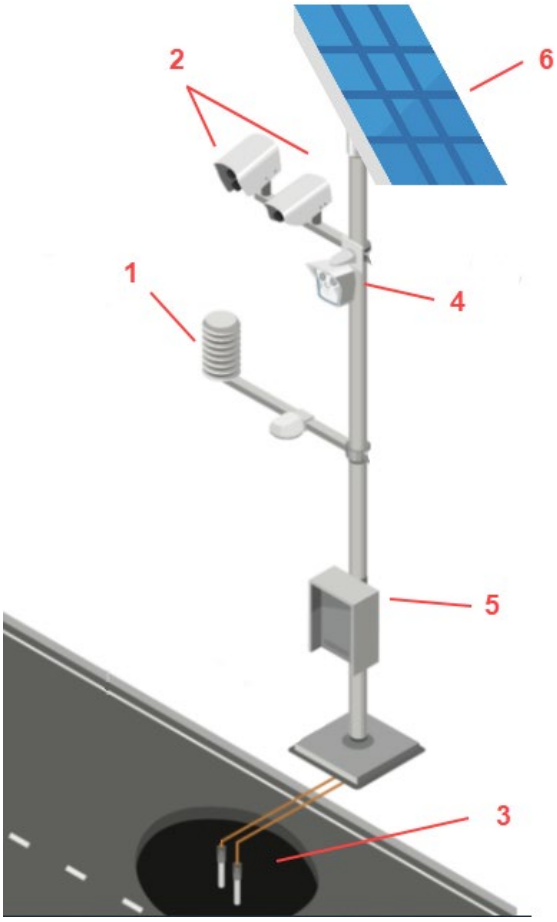


Figure 3-2. RWIS station illustration.

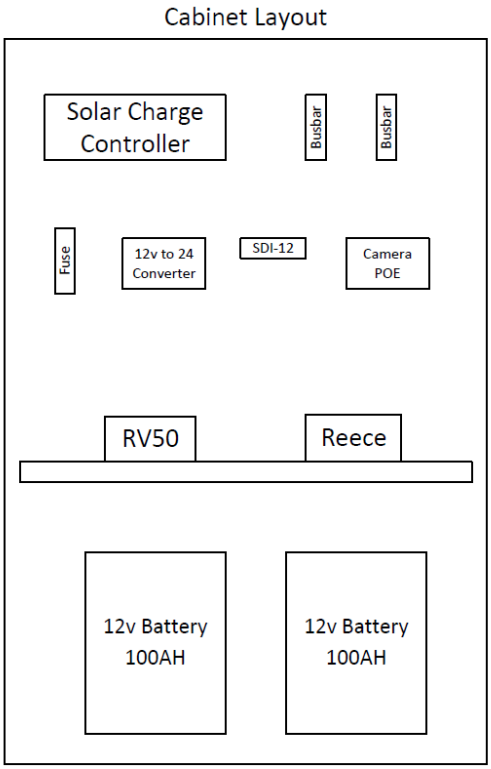


Figure 3-3. RWIS cabinet schematic.

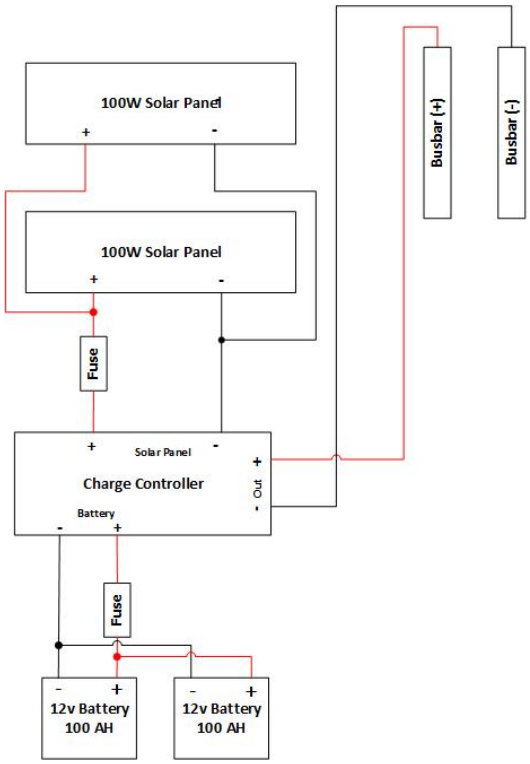


Figure 3-4. Solar panel circuit.

4 RWIS Data Analysis

Data collected from all 15 stations is sent to a local server via JSON packets, as shown in Figure 4-1, and saved in an SQL database. A user dashboard [16] will parse data and plot it into line and bar graphs to visualize current data. Hence, the graphical user interface (GUI) can be manipulated to select a specific range of dates (See Figure 4-2). The user dashboard has a sidebar showing all stations and a map illustrating current weather conditions for each station location. When a user selects the station with a full report, the conditions during that previous 24 hours is displayed (See Figure 4-3).


```
{
  "message_type":2,
  "message_sequence": 123,
  "message_body":
  {
    "station_id":"00ST000",
    "wind_speed": 3.36,                          // mph
    "wind_speed_of_gust": 3.808,                 // mph
    "wind_direction": "196",                       // degree
    "wind_direction_of_gust": 185,               // degree
    "temperature": 48.38,                        // Fahrenheit
    "relative_humidity": 33,                     // %
    "air_pressure": 1046.9,                      // hPa
    "brightness_north": 5619,                    // lux
    "brightness_east": 5422,                     // lux
    "brightness_south": 6235,                    // lux
    "brightness_west": 6158,                     // lux
    "brightness": 6235,                          // lux
    "brightness_direction": 0,                   // degree
    "precipitation_event": 0,                    // 0 or 1
    "precipitation_intensity": 000.000,          // mm/h
    "total_precipitation": 0.00,                 // mm/d
    "precipitation_type": 00,                    // (see table below)
    "latitude": 36.099793,
    "longitude": -095.924920,
    "altitude": 0221,
    "sun_elevation": 032.7,
    "sun_azimut": 206.3,
    "date": "08.11.19",
    "time": "13:38:51",
    "checksum": "*02",
    "ir_temp1_surface_temperature": 79.7,        // Fahrenheit
    "ir_temp1_air_temperature": 50.1,            // Fahrenheit
    "ir_temp1_relative_humidity": 33.5,          // %
    "ir_temp1_dew_point": 22.3,                  // Fahrenheit
    "ir_temp1_error_code": "00",
    "ir_temp1_fan_output": "5",                  // 5: fan on, 6: no start, 7: error, 8: off
    "ir_temp2_surface_temperature": 079.7,       // Fahrenheit
    "ir_temp2_air_temperature": 050.1,           // Fahrenheit
    "ir_temp2_relative_humidity": 33.5,          // %
    "ir_temp2_dew_point": 22.3,                  // Fahrenheit
    "ir_temp2_error_code": "00",
    "ir_temp2_fan_output": "5",                  // 5: fan on, 6: no start, 7: error, 8: off
    "temp_probe_1": 36.58,                       // Fahrenheit
    "temp_probe_2": 34.20                        // Fahrenheit
  }
}
```


Figure 4-1. Weather information message.

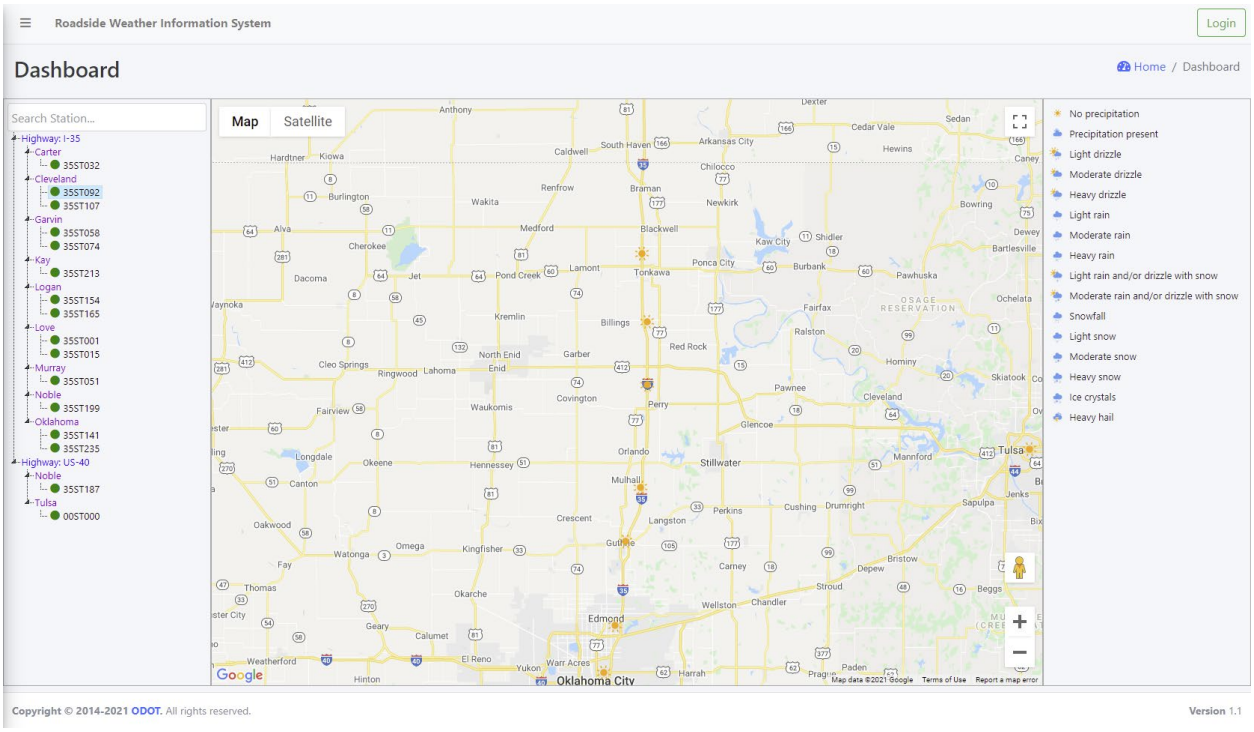


Figure 4-2. Main graphical user interface (GUI) dashboard.

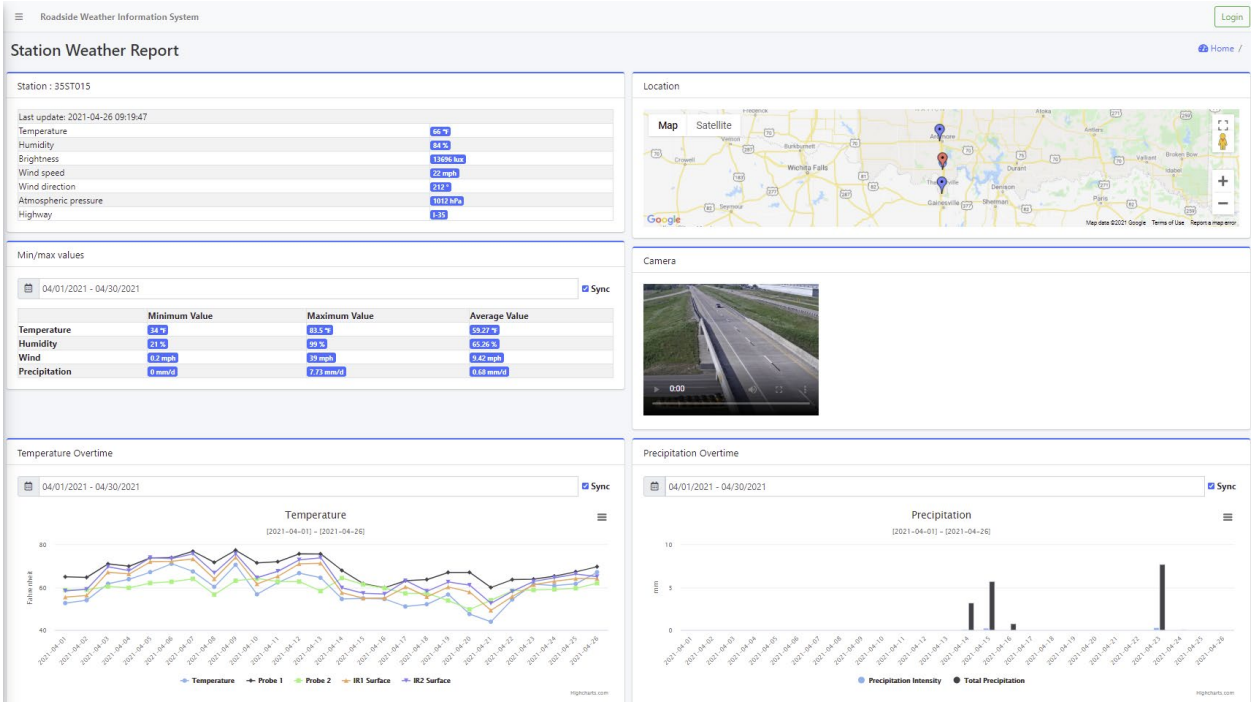


Figure 4-3. Station data dashboard.

The collected data shown in Figure 4-4 were generated from three systems—CLIMA, IR, and probes. Although more than 30 data attributes are sent to the database (See Section III), only 22 attributes were used for the research reported in this thesis.

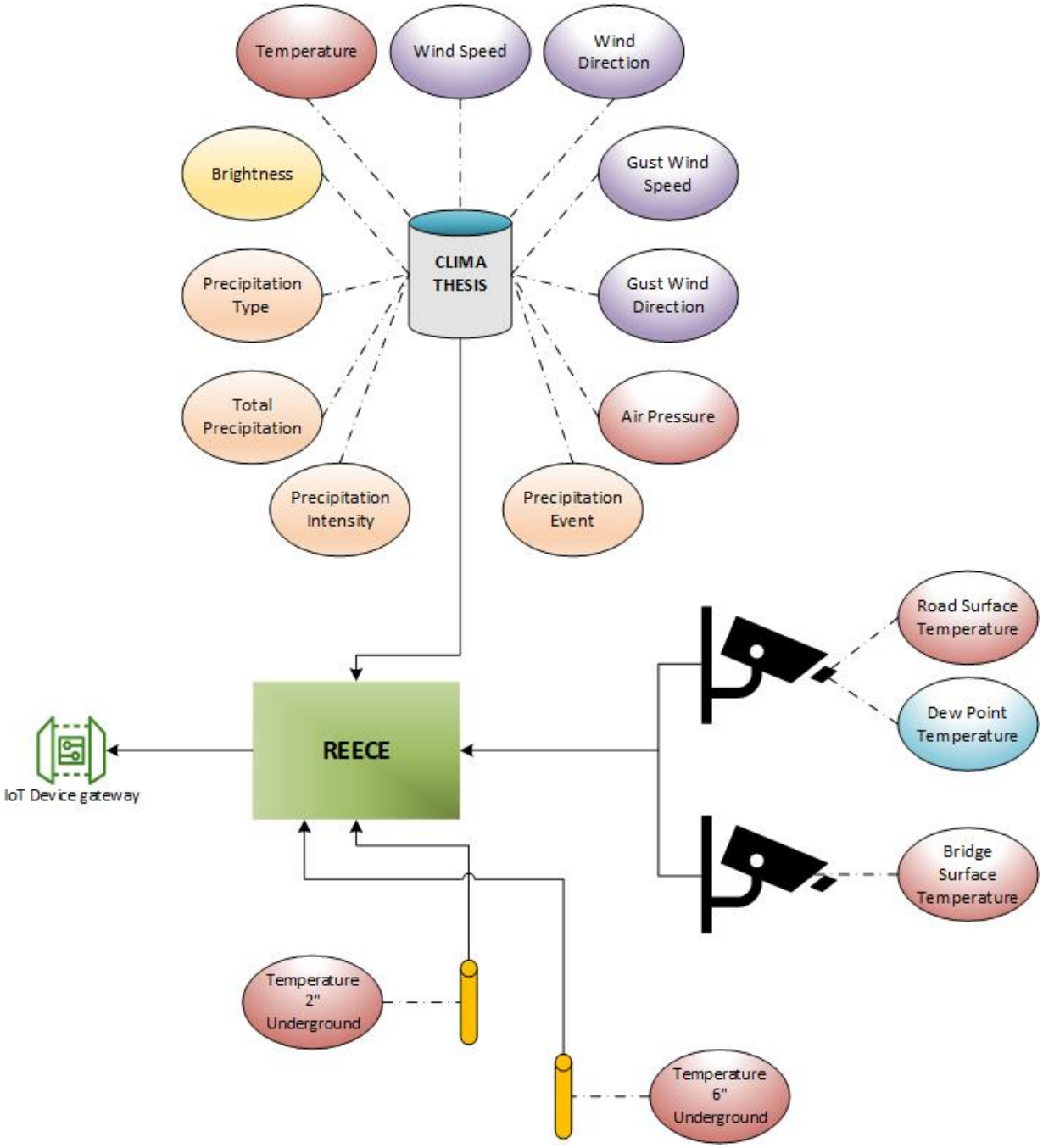


Figure 4-4. RWIS data attributes.

4.1 Data Preprocessing and Preparation

Before exploring the data and developing models, a number of features must be preprocessed and analyzed to detect erroneous readings. Data collection rate varies from 30 seconds to five minutes, depending on approaching weather events. Faster data collection is automatically triggered when the temperature is dropping rapidly. For a forecasting model, the time-series data must be spaced in equal time steps. Data was resampled into three datasets—5-minutes, 15-minutes, and 1-hour time periods.

Stations might experience occasional failure (i.e., stop collecting data). When batteries are not charged by the sun for a long period of time, data loss is experienced and could range from one hour to a few days. To overcome this performance shortcoming in the modeling, days with more than four hours of data loss and days with data loss resulting from sensor failure were removed. To ensure that the data is sampled in a correct sequence, stations with minimum data loss and good data quality without faulty sensing or inaccurate position oriented were picked for the modeling process. Figure 4-5 shows that stations 235, 213, 199, 51, 32, and 1 had the lowest number of down days, making them the best stations to use for this thesis.

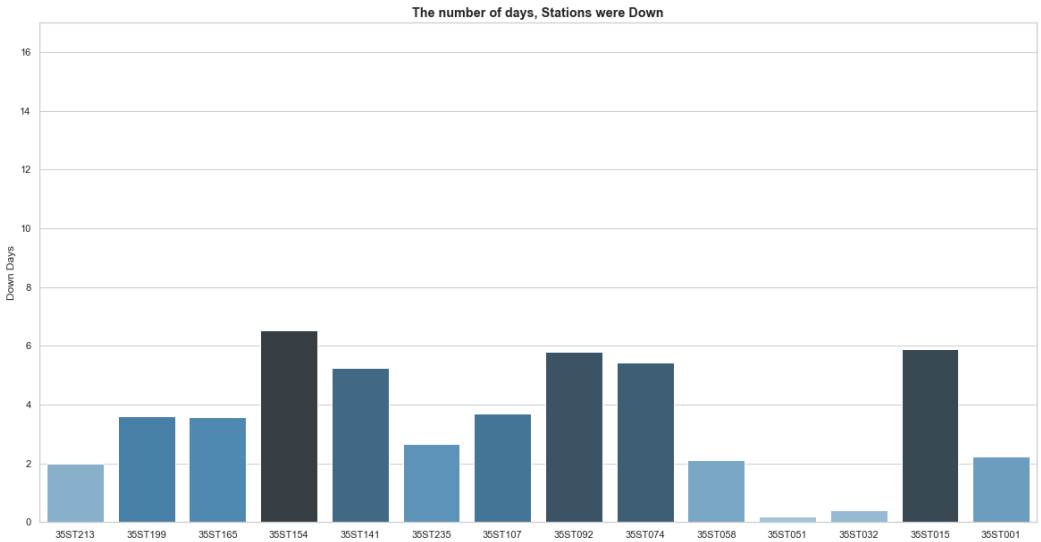


Figure 4-5. Number of days stations were offline.

Data were analyzed for outliers, NaN, and inf values. The dataset was first cleaned from NaN values by removing rows that contains any NaN values. CLIMA system data were then analyzed by comparing values with a valid range. A query function was performed on all attributes to remove any data point outside a valid range. Second, a sensor with processed data was processed is the IR sensors; infrared is sensitive to the outdoor atmosphere and may mislead the users to erroneous results [17]. For an IR sensor to measure road surface, vehicles travel on the road must pass through the measuring range of the sensor. Their reflective surface emits less infrared energy than normal objects, causing the sensor to deliver inaccurate readings. Figure 4-6 shows an example of reading errors (See sharp drops in the chart). To overcome this issue, a query function was used to correct temperature from 0 to 130 °F. Figure 4-7 demonstrates that this process did not correct the problem. IR readings are not smooth, and the figure still shows minor drops in temperature readings that can't be removed by adding a data range. Hence, a smoothing filter method was employed. A one-dimensional Gaussian Kernel [18] generated an impulse response function with standard deviation of 10 pixels. Array function *f* and kernel *g* indicated that *f* has a length of *n* and *g* has a length of *m.* The convolution function *f***g* of *f and g* is defined in function 1.

$$(f * g)(i) = \sum_{j=1}^{m} g(i) \cdot f\left(i - j + \frac{m}{2}\right) \quad (1)$$

This same function is applied to RST and BST; successfully output smoothed data can then be used in the forecasting model. Figure 4-8 shows results after data processing. The remaining sensors serve as underground probes. Probe's data were accurate and did not require conditioning.

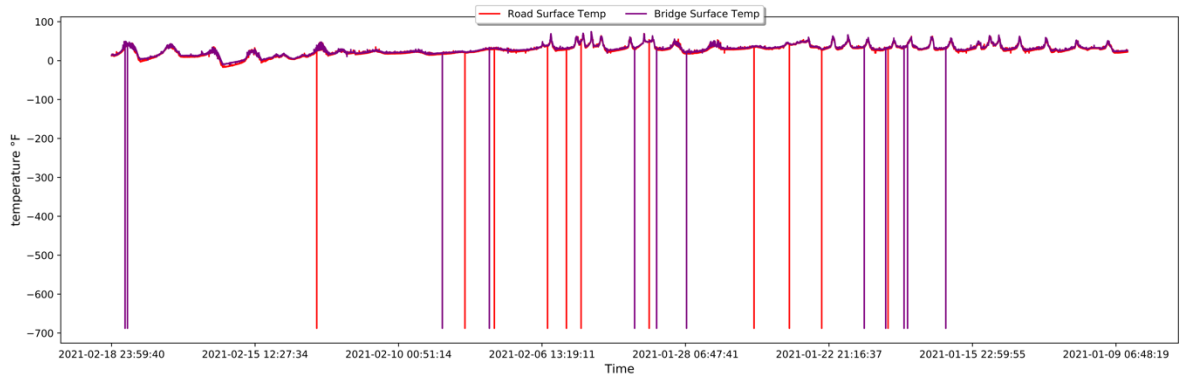


Figure 4-6. Raw RST and BST data output.

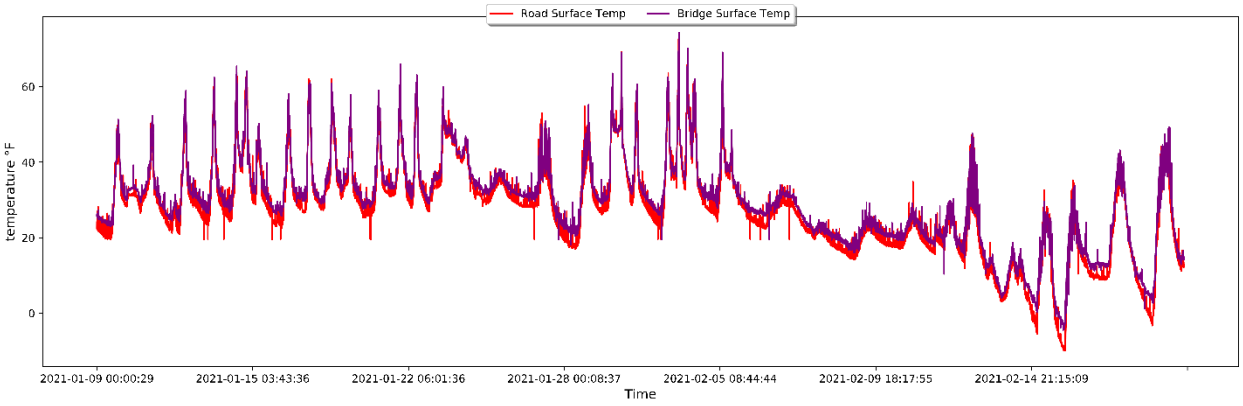


Figure 4-7. RST and BST data after outlier removal.

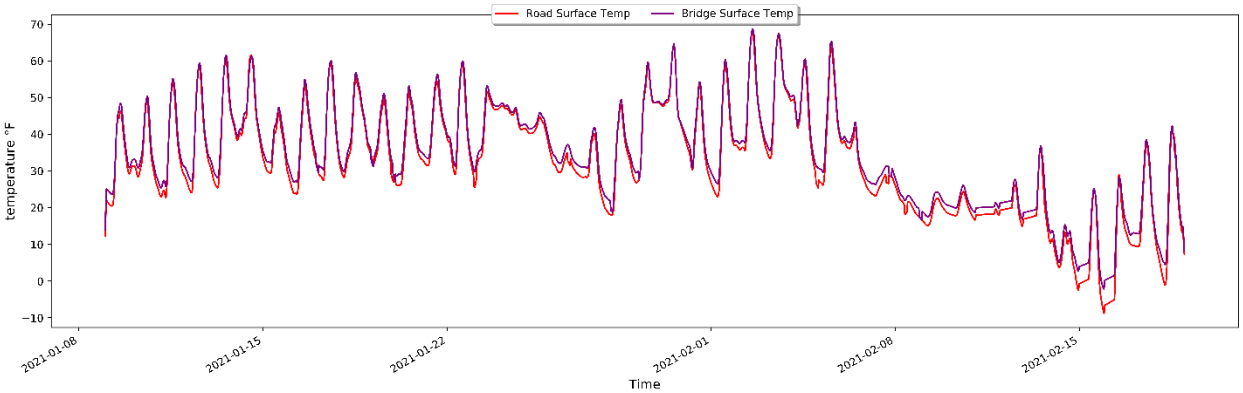


Figure 4-8. RST and BST data after convolution filtering.

A python code (i.e., functions) was constructed using preprocessed data and a RESTful API for parsing data using an HTML request. As mentioned earlier, data was resampled into three datasets. Likewise, three functions were developed to call the datasets. The first will call the

dataset of specific station; the second will call datasets with a resampled size based on user choice; and the third will parse each attribute from all stations and combine them into one dataset. The third function has a nested loop to parse data from north to south. The loop will also verify if a station is offline. Given that the statement is true, the loop will use data from the previous station.

Function 1: Parse data of a specific station into a dataset

Function Get_RWIS (Station ID, From, To)
    Request: HTTP Data
    Append: Attributes
    X → Length of JSON message
    For each X
        Add Append: Attributes
    DF → sum of attributes
    Remove: Nan
    Set Index: Date
    Query: Dataset
    Convolve Filter → IR Data
    Return DF

Function 2: Parse data of a specific station into a dataset with a desired resample size

Function Get_RWIS_Resampled (Station ID, From, To, Sample Size)
    Request: HTTP Data
    Append: Attributes
    X → Length of JSON message
    For each X
        Add Append: Attributes
    DF → sum of attributes
    If Sample Size **is** 5 **then**
        Resample_5: DF
    Elif Sample Size **is** 15
        Resample_15: DF
    Else Sample Size **is** 1H
        Resample_1H: DF
    End If

    Remove: Nan
    Set Index: Date
    Query: Dataset
    Convolve Filter → IR Data
    Return DF

Function 3: Parse data in real time from all stations for a specific attribute

Function Get_RWIS_Real (Data)
    Request Datetime.now
    Request HTTP Data
    Append Attributes
    X → Array of stations ID's
    If Data **is** Name of variable **then**
        For each X
            Add Append: Stations
            N → length of Attributes
            If not Stations
                Stations **is** Null
            End If
            For each N
                Add Append: Attributes
            If not Attributes
                Attributes **equal previous** Attributes
            End If
            Mean Attributes **equal mean** Attributes
            Delete Attributes
    DF → Mean Attributes

Running Function 1 assists modeling and simplifies the steps by merely applying station name and date duration. A user can plot the data and visualize the fetched data. Table 4-1 shows chosen attributes from Function (1), shown below, with the data index selected by the user. Data were selected using the Function (2). This function parses all current values within the last hour of each station, calculates the mean value, and adds it to the dataset, as shown in Table 4-2. Output data is cleaned and ready for a user to import into the model. Figure 4-9 shows preprocessed data in a plot of Ambient, RST, BST, and 2- and 6-inch probe temperature data with black line indicating freeze level. In this figure, data indicates that surface temperature crossed the freezing level while ambient temperature was still below the freezing level.

```
data = Get_RWIS_Resampled('35ST213','2021-01-09','2021-02-18',15)
```
(1)

```
data = Get_RWIS_Real('RST')
```
(2)

Table 4-1
Parsed Data from Function 1

Index	Wind Speed	Wind Direction	Precipitation	Temperature	…	RST	BST	2” UG	6” UG
2021-01-09 00:00:00	1.98	120.0	0.0	11.87	…	7.29	8.16	23.95	29.42
2021-01-09 00:15:00	1.14	232.67	0.0	11.79	…	8.48	9.54	24.13	29.41
…	…	…	…	…	…	…	…	…	…
2021-02-18 23:30:00	2.68	321.0	0.0	24.65	…	14.59	16.47	35.33	37.13
2021-02-18 23:45:00	1.64	259.52	0.0	25.70	…	12.39	13.96	35.49	37.21

Table 4-2
Parsed Data from Function 2

Station ID	Value
213	24.36
199	23.78
187	23.78
…	…
31	31.87
15	32.14
1	32.08

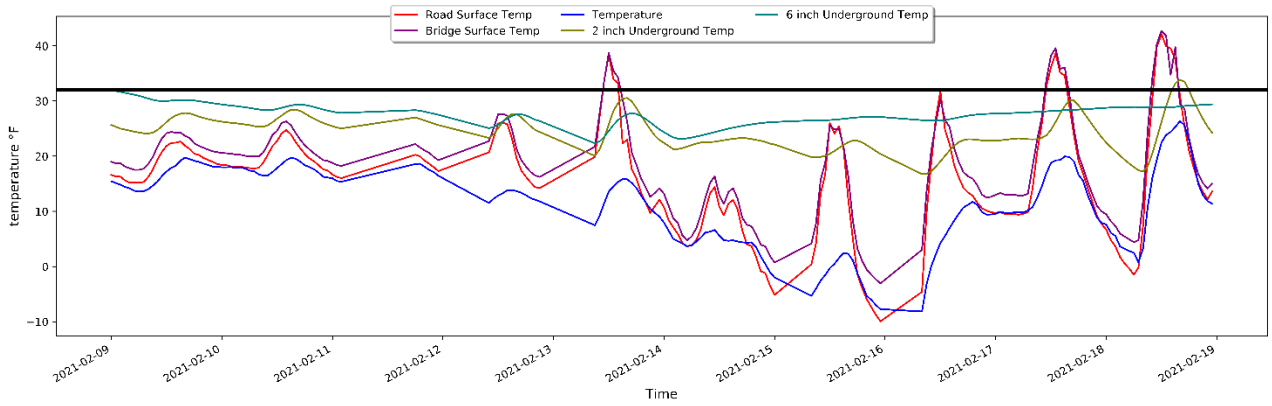


Figure 4-9. API function temperature plot.

4.2 Road Conditions Classification

A roadway has several types of conditions that can be classified as safe or hazardous. Snowstorms bring various snow precipitation types. Some wintery precipitations are significantly dangerous for vehicles traveling on the road and have the potential for causing deadly accidents. Furthermore, temperature, per se, is not a significant factor on road condition classification. For example, it is possible for temperature to drop below freezing level during a precipitation event and the road surface to remain safely drivable. Hence, it is important to investigate data patterns to assist with model development.

During the winter months, the state of Oklahoma typically experiences four types of precipitations: rain, freezing rain, sleet, and snow [19]. The current mechanism for predicting these four types is examining all atmospheric layers. Meteorological events usually occur only in the troposphere layer, which reaches from four to 12 miles above the earth surface. 70 to 80% of the earth's atmosphere falls within this layer [20], and temperature may drop to as low as -60°F.

The change variation in temperature through the troposphere layer aids in predicting precipitation type. A constant decrease in temperature as we move away from the earth surface is ideal. However, according to atmospheric thermodynamics, this is not always the case. The atmosphere layer is a non-equilibrium system, wherein a buoyant force causes warm, less dense air rises and forms condensation in the upper layers. The European Centre for Medium-Range Weather Forecasts explains that in an atmospheric thermodynamics study [21], sun radiation causes the release of heat energy into the atmosphere, making the air rise above the cold air mass. Figure 4-10 shows a cold-front system affecting Oklahoma and causing convection dynamics. Figure 4-11 illustrates the way in which a warm air mass rises and flows into the convective core, causing air saturation and forming a severe thunderstorm.

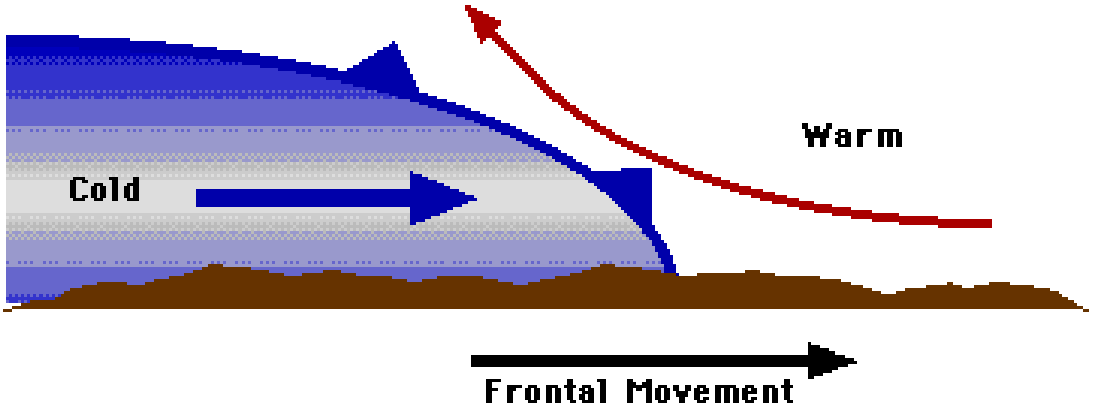


Figure 4-10. Side view of a cold front [22].

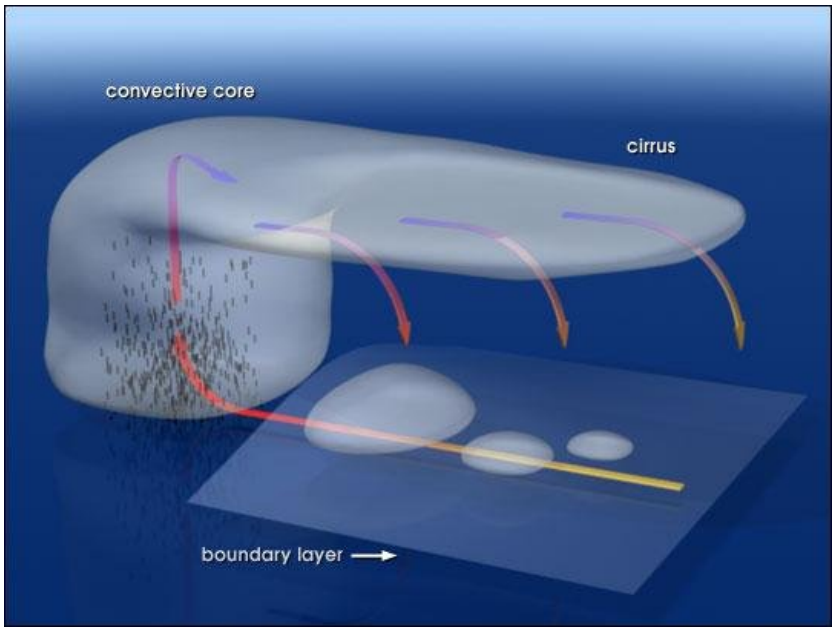


Figure 4-11. Convection formation [23].

Air dynamics demonstrates how air mass can cause temperature change in atmospheric layers wherein clouds and precipitation forms. Meteorologists use an energy diagram (i.e., Temphigram) to obtain a clear picture of expected precipitation. The graph collects data along all atmospheric layers as dewpoint, wind direction and speed, temperature, and moist adiabatic lapse rate (i.e., cooling rate of rising saturated air).

The skew-T graph in Figure 4-12 illustrates the entire atmospheric elevation, which reaches up to 25 km. Temperature lines run from southwest to northeast across the diagram, and dashed lines represent saturation wherein a mass of water vapor is divided by a mass of dry air. The y-axis shows air pressure in millibar; the x-axis shows wind bars, indicating wind speed and direction. The graph is characterized with three-line plots: leftmost is the cape, middle is dewpoint temperature, and rightmost is the environmental temperature. Once dew and environmental temperature lines cross each other, a saturation level occurs and, with the help of the cape, cloud formation starts. After two lines diverge, as illustrated at 300 mb, cloud formation stops. Precipitation point temperature is just above freezing from surface, which can completely melt frozen precipitation before it reaches the earth's surface. In such a case, rain will form.

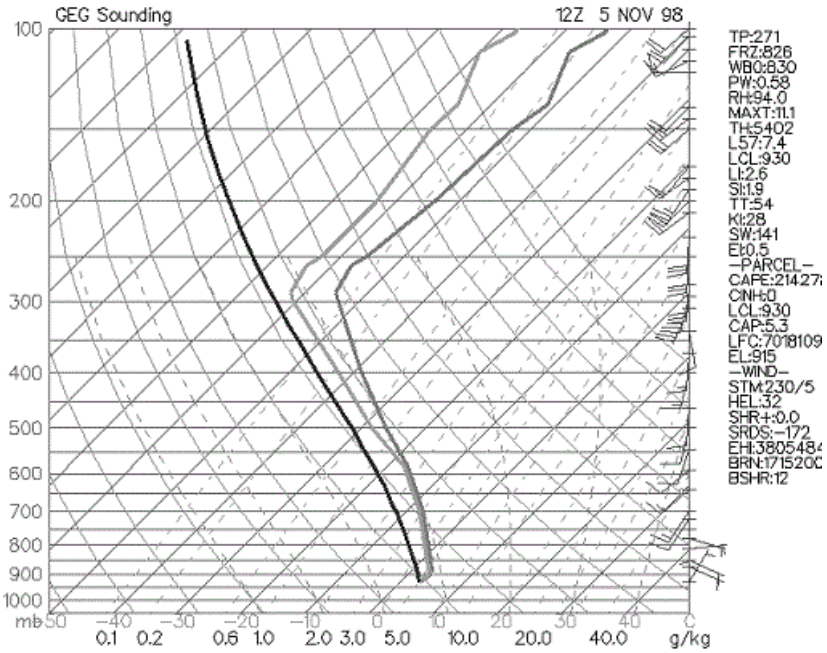


Figure 4-12. Cold rain sounding skew-T graph [24].

As mentioned earlier, different road conditions can occur during a winter event; three types of winter events precipitations are possible (See Figure 4-13). Oklahoma typically experiences all three types of wintry precipitation each winter. For normal snow (type one) to occur, surface level through saturation level temperatures must remain below the freezing level. Sleet (type two), in

which snow melts and then freezes on the surface, occurs when the atmosphere experiences a thin, warm air mass in the upper level between saturation and surface levels. Freezing rain (type three) occurs when a shallow freezing layer forms just above the surface. During this condition, raindrops freeze at the point of contact with the surface layer, creating extremely hazardous driving conditions.

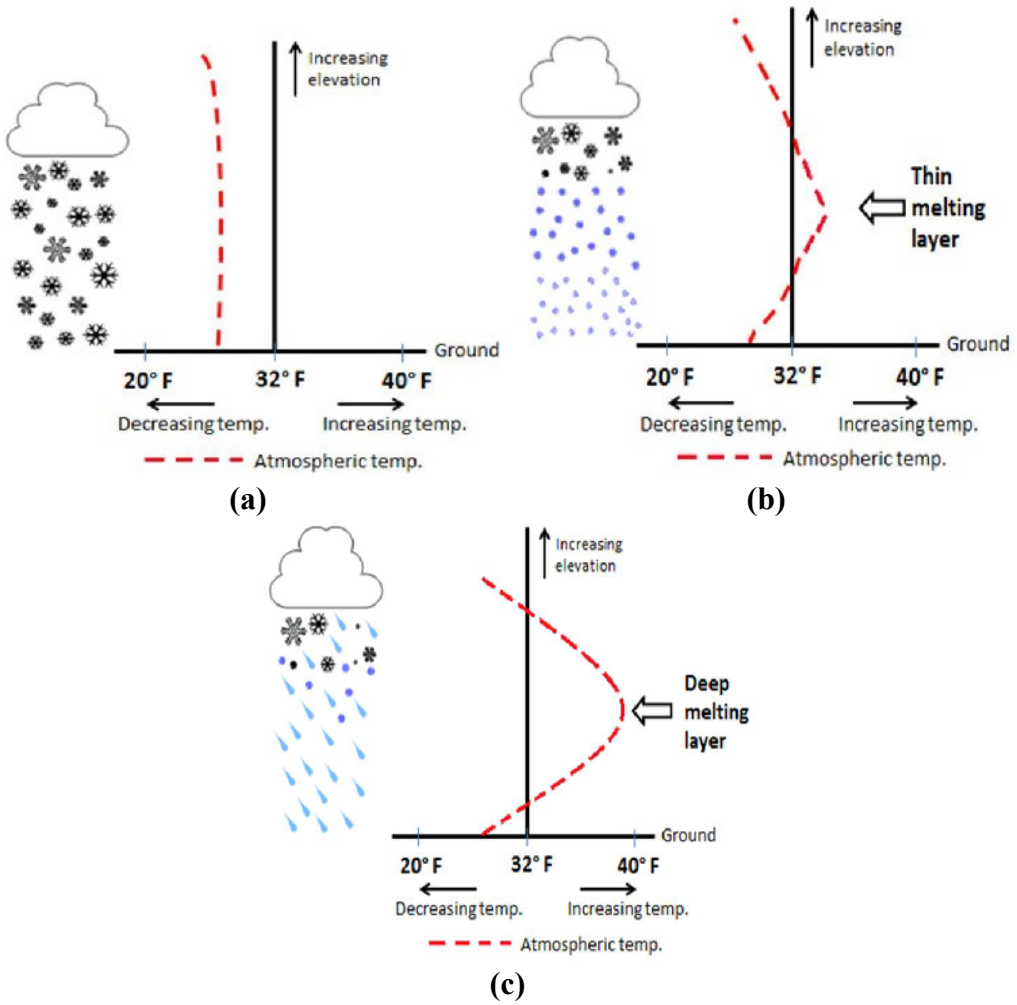


Figure 4-13. Three types of wintry precipitation: (a) snow (b) sleet (c) freezing rain [25].

These three weather classes must be accurately added to the dataset for indicating hazardous or safe driving conditions from an upcoming wintry event. The Oklahoma National Weather Service collects historical data of precipitation events from images captured by maintenance trucks

traveling on state roadways and saves them on updated road condition maps [26]. Snow and wet conditions were indicated as safe road conditions; slick, freezing rain is indicated as hazard. Some snow events were later recorded as hazardous due to a mix of air masses that caused snow to melt and then freeze again, which caused hazard driving conditions. For the sake of classification, a dataset obtained between October 1, 2020 and February 16, 2021 for Oklahoma county was manually parsed and added to Station 235, located in Oklahoma county. See Table 4-3 for a summary of events per class.

Hazard events are indicated by blue bars in Figure 4-14. The red line represents BST; green line indicates dew point temperature; and yellow line indicates precipitation intensity. Note that each drop in BST and a precipitation event is followed by a hazard condition. The study reported in this thesis focuses on four events that occurred in October 2020, December 2020, January 2021, and February 2021.

Table 4-3
Precipitation Event Count Recorded in Historical Dataset

Class event	Event count
Dry & Wet	36805
Snow	341
Sleet & Freezing rain	4402

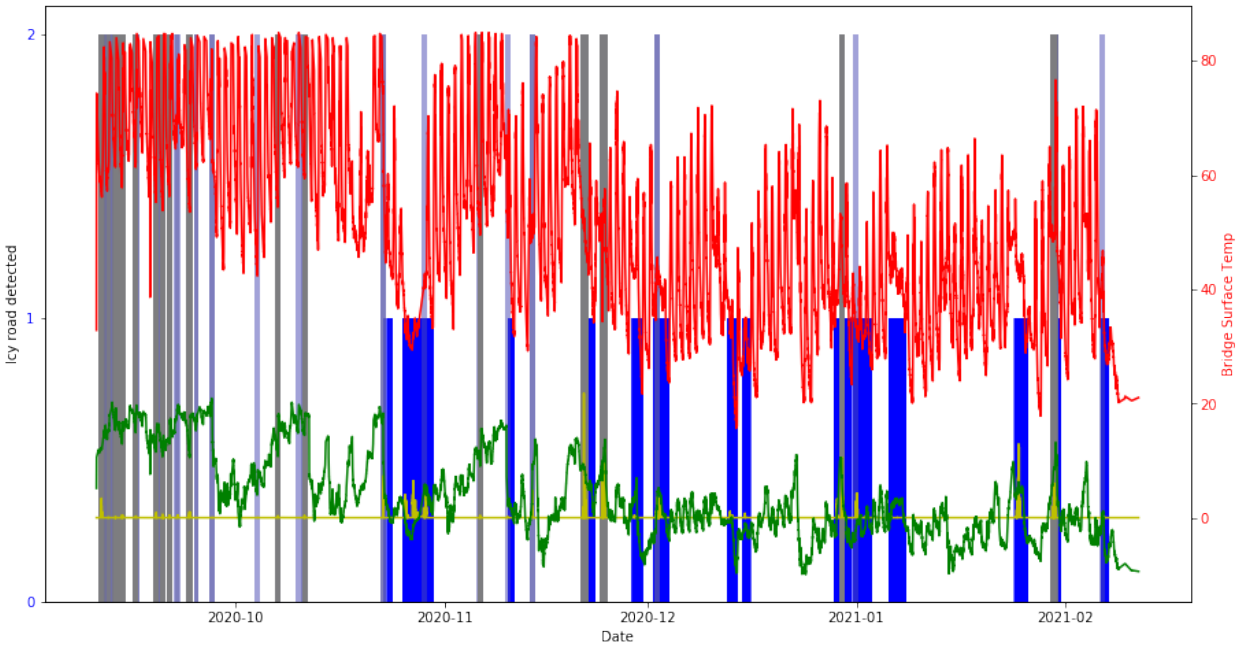


Figure 4-14. Road condition classification graph.

4.3 GFS data Parsing

Section 4.2 explained why it is important to gather information about changes in higher layers of the atmosphere. Figure 4-15 illustrates how the RWIS monitors only a small region of the overall atmosphere and that temperature is irrepressibly changing as elevation increases. Abrupt changes in top atmosphere layers are the primary cause for severe weather conditions and precipitation type indications. RWIS collected data is unable to forecast such changes due to lack of information about changes in these upper atmospheric layers. As such, a future indicator of the changes is needed to link with RWIS data and increase road condition prediction accuracy.

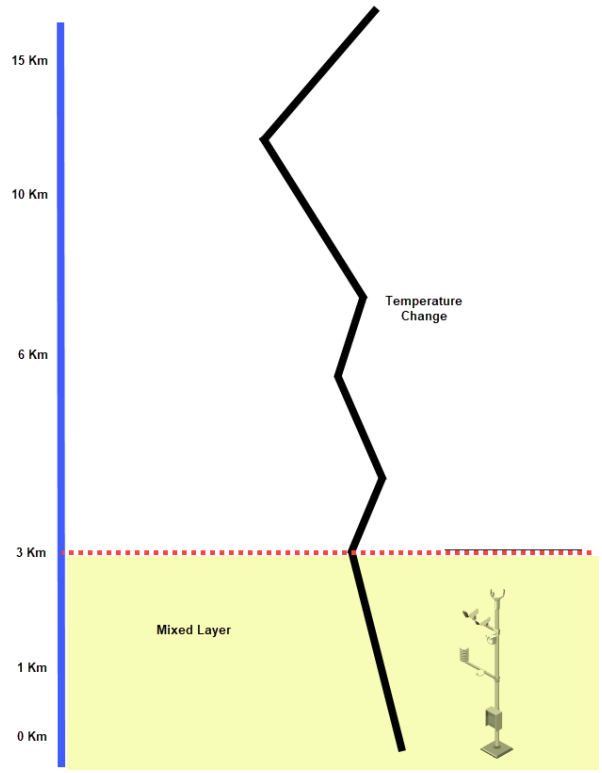


Figure 4-15. Atmosphere temperature changes in upper layers.

A variety of models are used to forecast the weather (e.g., GFS, ECMWF, CMC, JMA and HWRF). GFS has a highly accurate forecast for 72+ hours and is regularly used by NOAA. The agency recently upgraded a higher version (e.g., HRRR) for local regions inside the US. This thesis utilizes an API restful data function operated by ClimaCell [27] to parse temperature and precipitation data for two cases that are highlighted in Sections 5.1.4 and 5.2.3. Data location is required for parsing the data Figures 4-16 and 4-17 illustrate how nine blocks were added along the I-35 corridor, each block's precipitation or temperature data will be parsed and averaged, and then add to a data frame with projected time series for a 24-hour future forecast. A function has been built to parse data from the nine regions and add it to the model scheme. The only requirement is region ID.

Figure 4-16. Forecasted atmosphere temperature.

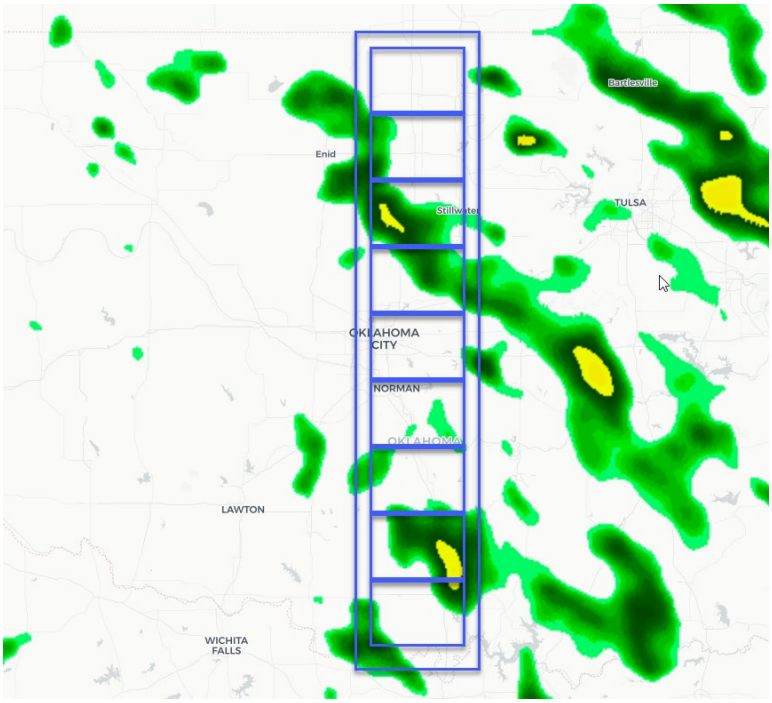


Figure 4-17. Forecasted total precipitation.

Function 4: Parse data of GFS Model

Function Get_RWIS_Resampled (Zone ID)
ZZ = Get_Corr (Zone ID**)**
Request: HTTP Data (Attributes, ZZ)
Append: Attributes
X → Length of JSON message
For each X
Add Append: Attributes
DF → sum of attributes
Set Index: Date
Return DF

4.4 Exploratory Data Analysis

To better understand acquired data used in this thesis, the information was processed and scripted to generate an exploratory data analysis (EDA). The main goal was using data to predict RST/BST and classify road conditions. Hence, regression and classification models were employed.

Weather data has a seasonal shape dependent upon max value temperature at noon and min value at night. Simple forecasting models can easily predict and feed this information into the model based on historical data. The research presented in this thesis focuses on cold fronts that force the seasonal shape into non-uniform shapes. Figure 4-18 illustrates the period from October 21 to 25 (i.e., 1 and 2), demonstrating a seasonal shape. Note that temperature has max and min peaks until a cold front arrives on October 26, which forces the temperature to display a flat line of data, as indicated in 3. The goal of the proposed model is to predict the timing of the drop, and then indicate whether or not the line will cross the freezing level. Figure 4-19 shows dew point drops during the cold front to the point a level of saturation is reached in the upper level as a result of warm air mass upward movement. This phenomenon strongly indicates the formation of precipitation prior to its occurrence. Figure 4-20 shows the low levels of brightness during the cold front, which proves to be a huge factor on road surface temperature. In such a case, the roadway absorbs solar energy, melts wintry precipitation while ambient temperature remains below freezing

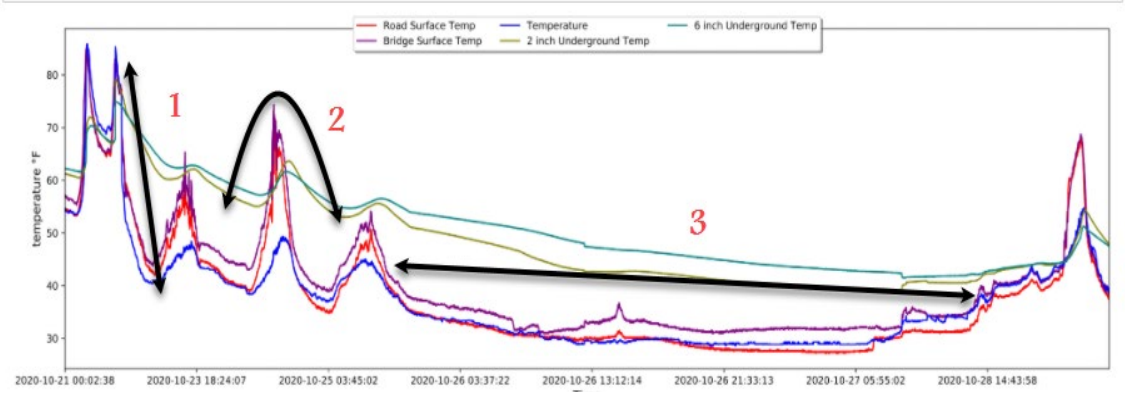


Figure 4-18. Temperature readings during a cold front.

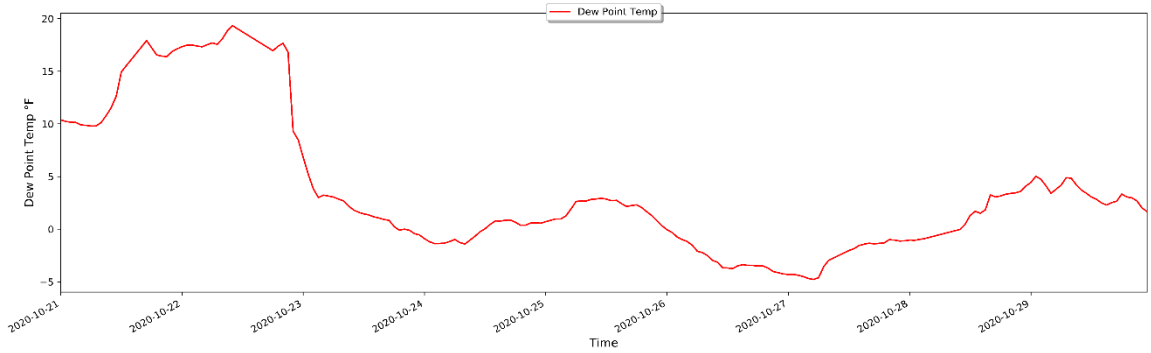


Figure 4-19. Dew point temperature during a cold front.

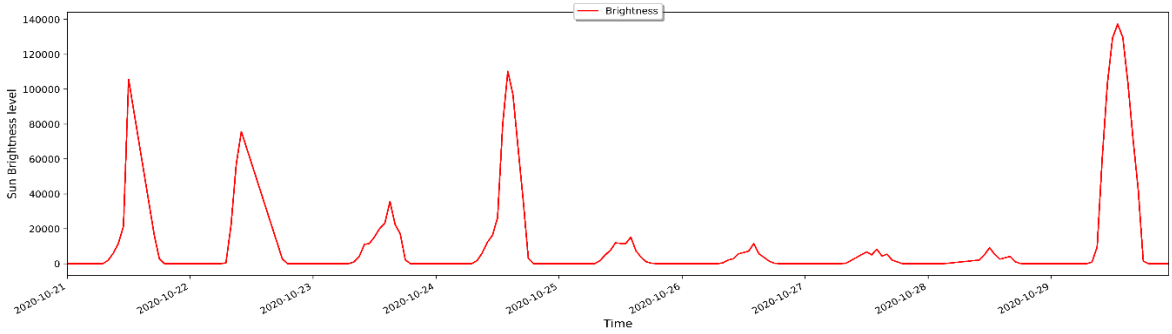


Figure 4-20. Brightness readings during a cold front.

For regression modeling, the correlation of each individual attribute and correlation between each attribute and the remaining attributes is required. To measure such linearity, a Pearson correlation coefficient (r), was used in Equation 2. The x and y features were utilized to calculate the amount of linear correlation between equal length arrays and output a value between -1 and 1. Figure 4-21

shows correlation results. Note a correlation between wind direction and pressure due to the fact that the cold front changes wind direction after it enters the region. Brightness also has high correlation as a consequence of cloudy weather during wintery events. These correlation factors are important for regression models and are significant for predicting target BST and RST when using a series of supervised data sequences.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (y)^2]}} \quad (2)$$

To visualize the relationship between the array of each feature and prove linear relationship in each feature data, each attribute was visualized to provide a general overview of the observed value with class, trend, seasonal, residual, histogram, boxplot, and auto and partial correlation. Notably, autocorrelation is a lag plot that ensures data is free from random structure versus time series. Lag is a fixed time displacement wherein each point in the data array is plotted against the point following it. ACF at lag *k* of the time series is defined in Equation 3, where n is the total number of values inside the array and y_k is the covariance of y_i and y_k. Plot variation ranges between -1 and 1, which describes correlation, respectively. Number of lags is limited to 40 for eliminating noise

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i\,, \quad S_k = \frac{1}{n}\sum_{i=1}^{n-k}(y_i - \bar{y})(y_{i+k} - \bar{y}) \quad (3)$$

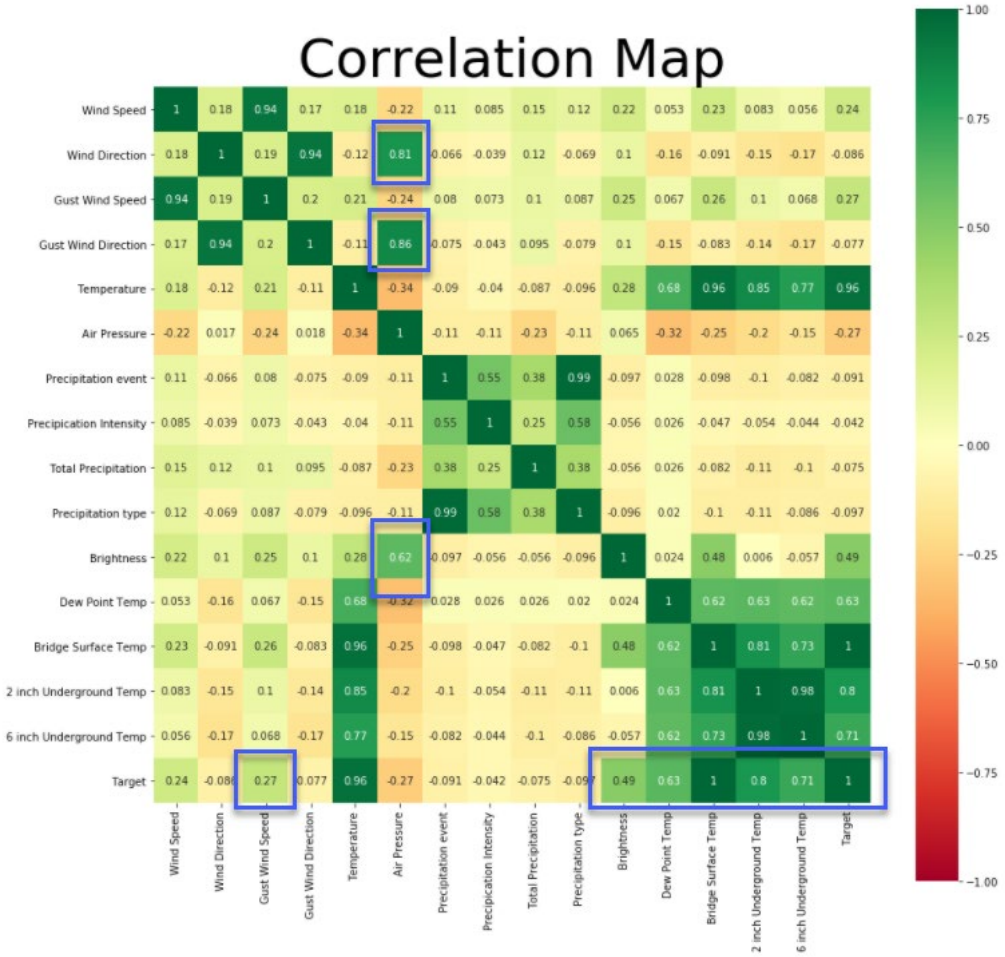


Figure 4-21. Correlation map of data features.

Subplot figures show time series exploratory analysis of each feature. First left is a line plot of the value. Red indicates safe road conditions; green indicates hazard conditions. First right is a trend value (i.e., additive time series). The array will combine to create a time series. An increase indicates the same peak size with relative changes. Second left is seasonal with a short-term cycle in series. Second right is residual, representing leftovers after fitting a model. This amount is equal to the difference between observations and corresponding fitted values. Third left is the histogram, showing distribution and value variation inside the array. Third right is a box plot showing max, min, and mean values. The last two are auto and partial autocorrelation. Figures 4-22 to 4-27 show that all attributes have a strong indicator that aids regression predictions alongside strong indicators

following hazard conditions as temperature drops and wind direction rotates. The one exception is total precipitation value, which has a very random occurrence. For this reason, the GFS model is useful for the modeling process (See Section 5).

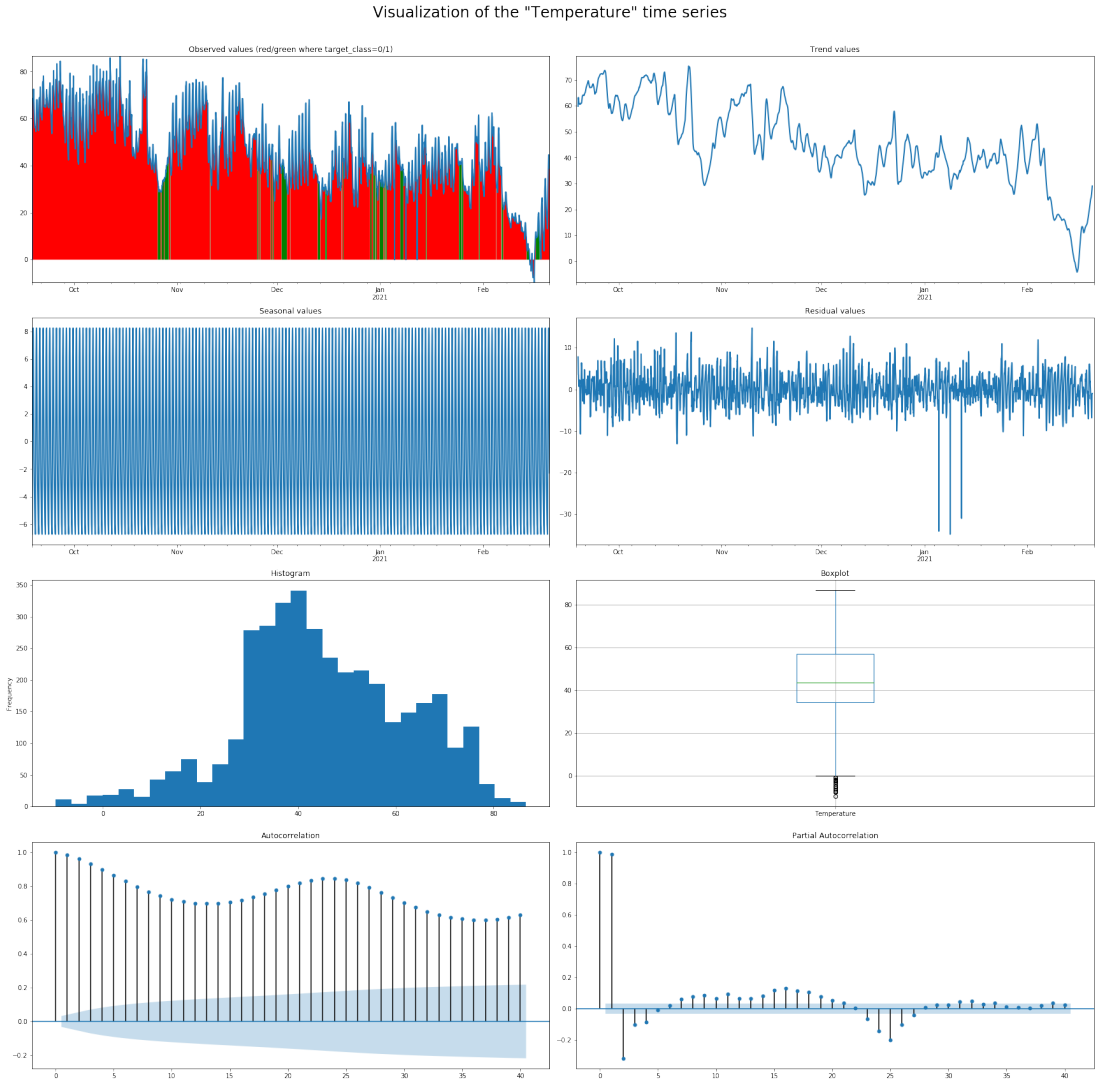


Figure 4-22. Temperature attribute EDA.

Visualization of the "Road Surface Temp" time series

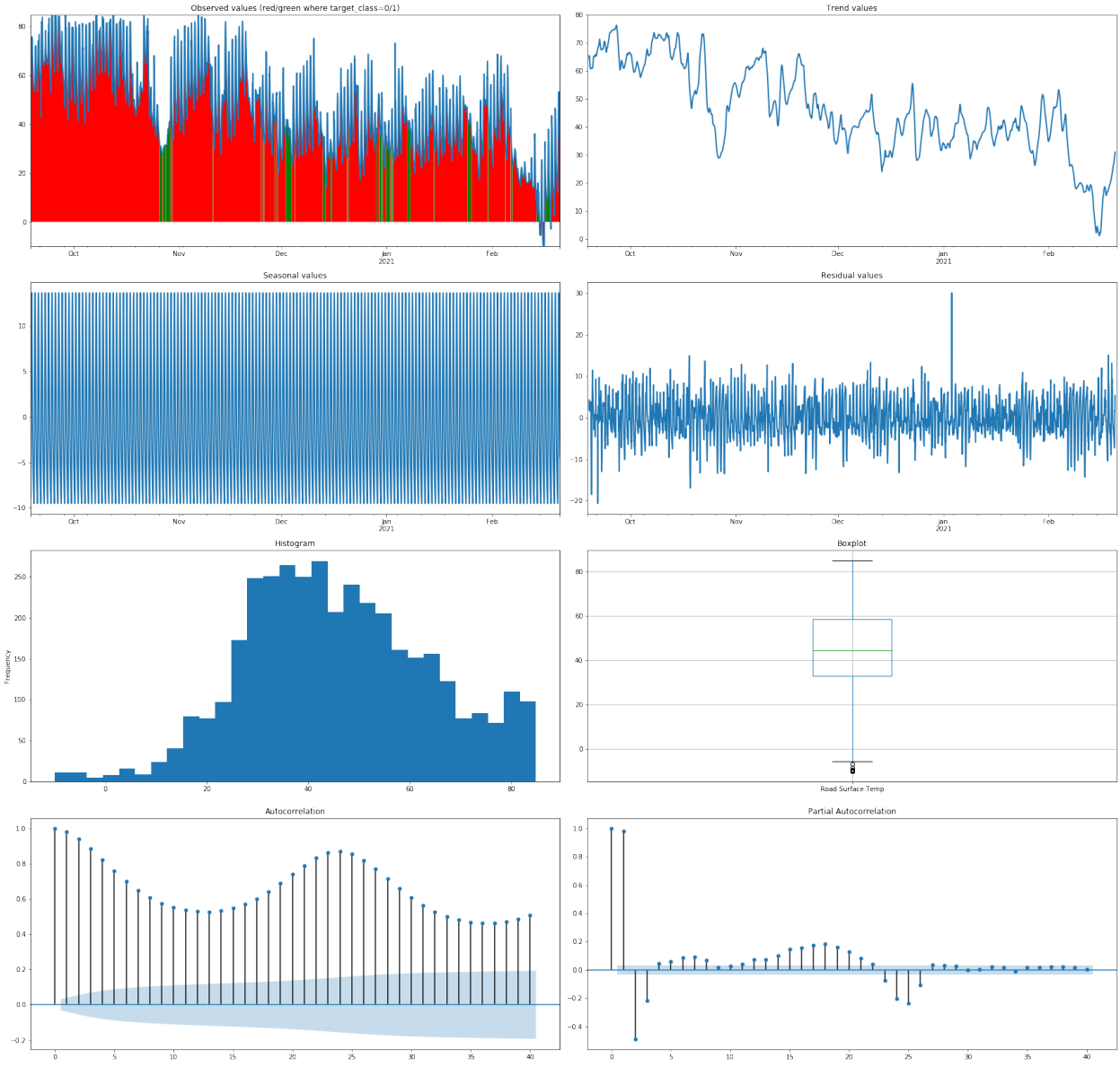


Figure 4-23. Road surface temperature attribute EDA.

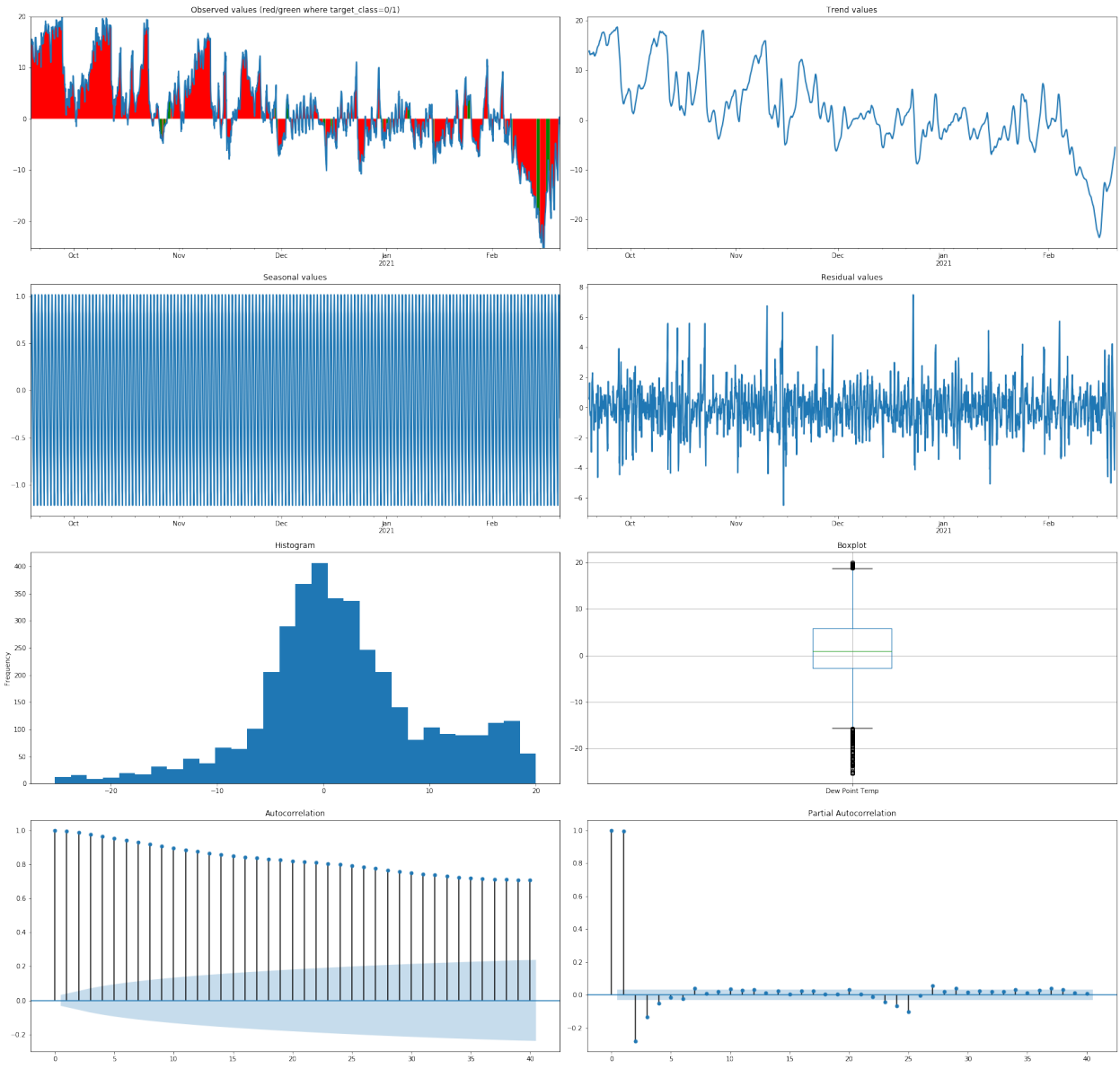


Figure 4-24. Dew point temperature attribute EDA.

Visualization of the "Total Precipitation" time series

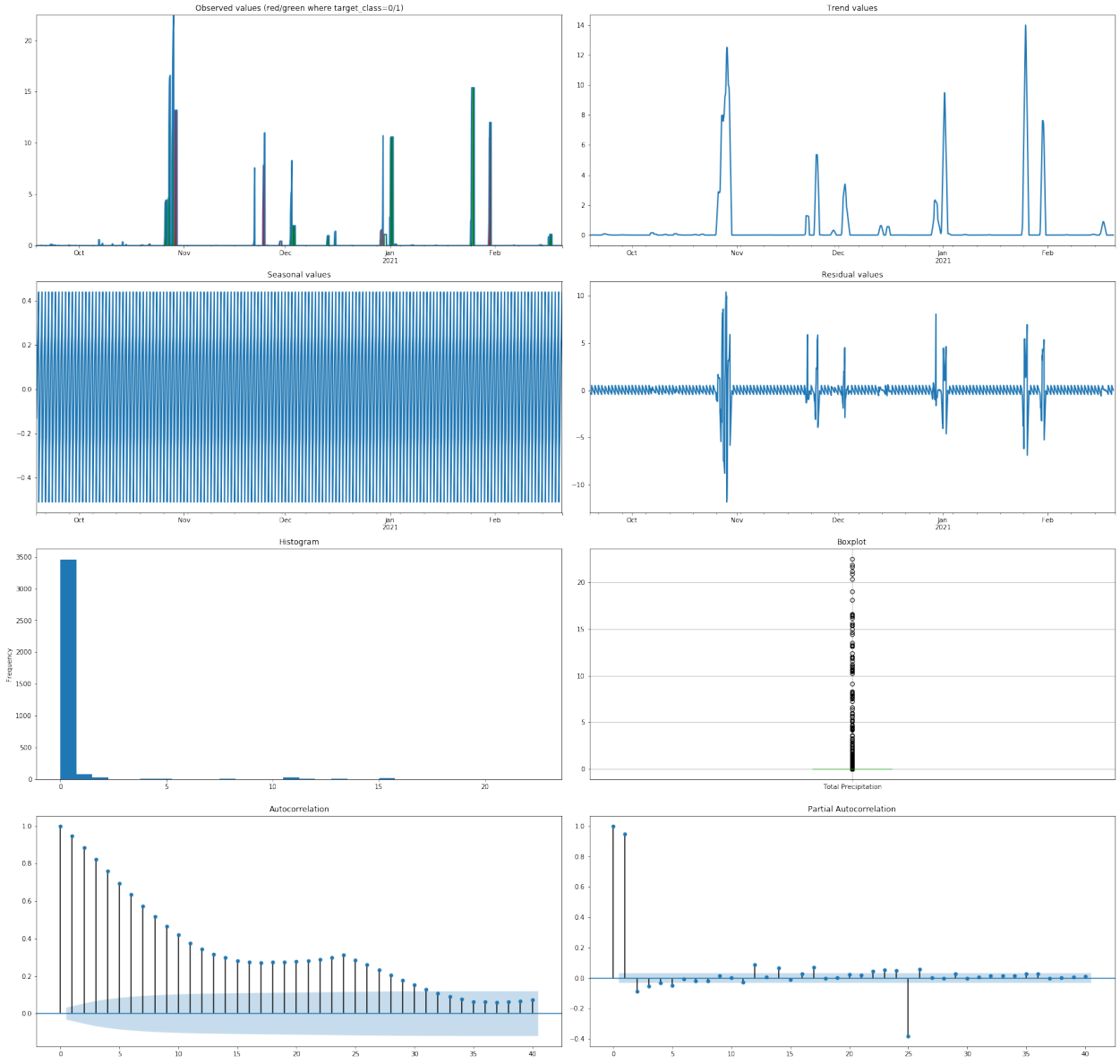


Figure 4-25. Total precipitation attribute EDA.

Visualization of the "Gust Wind Speed" time series

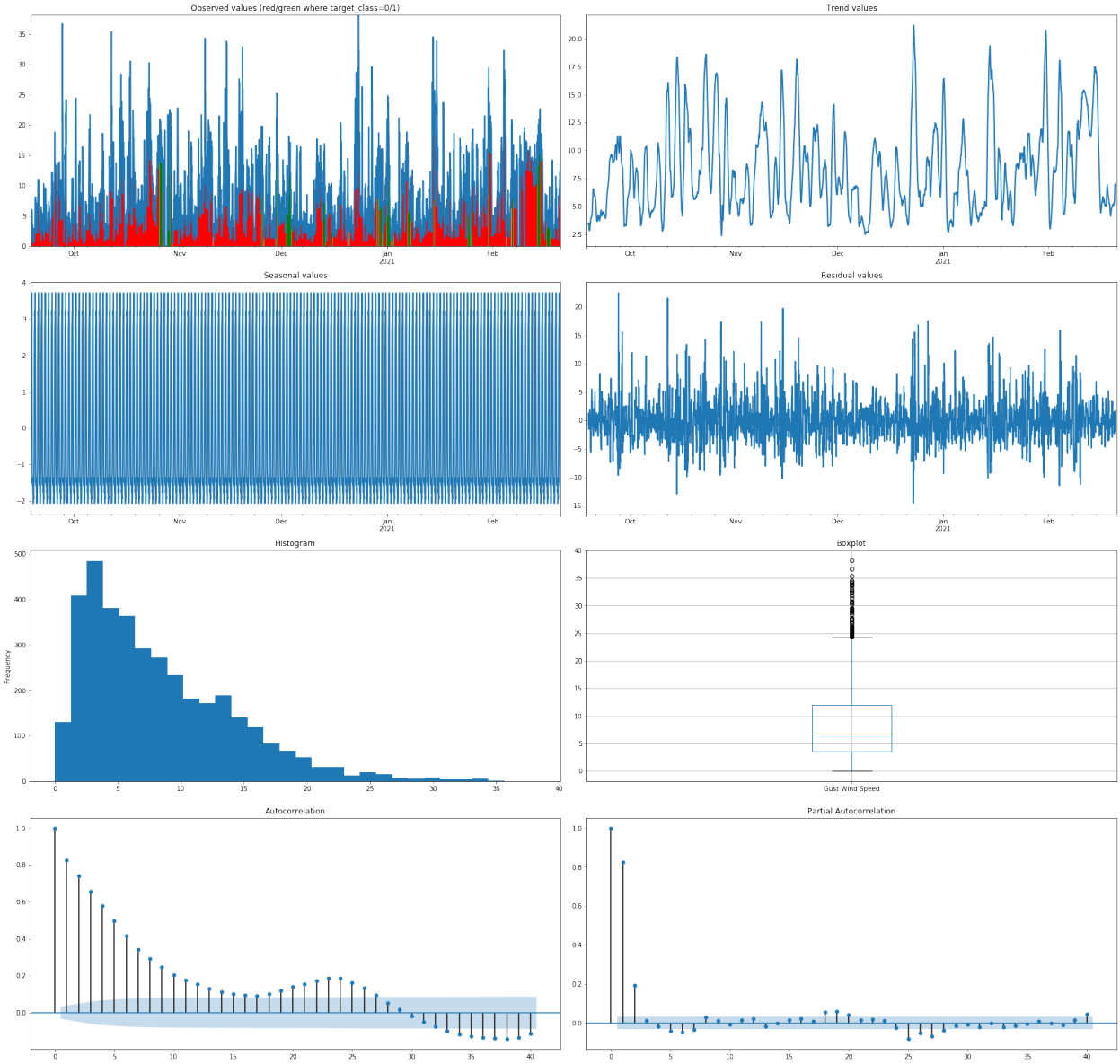


Figure 4-26. Gust wind speed attribute EDA.

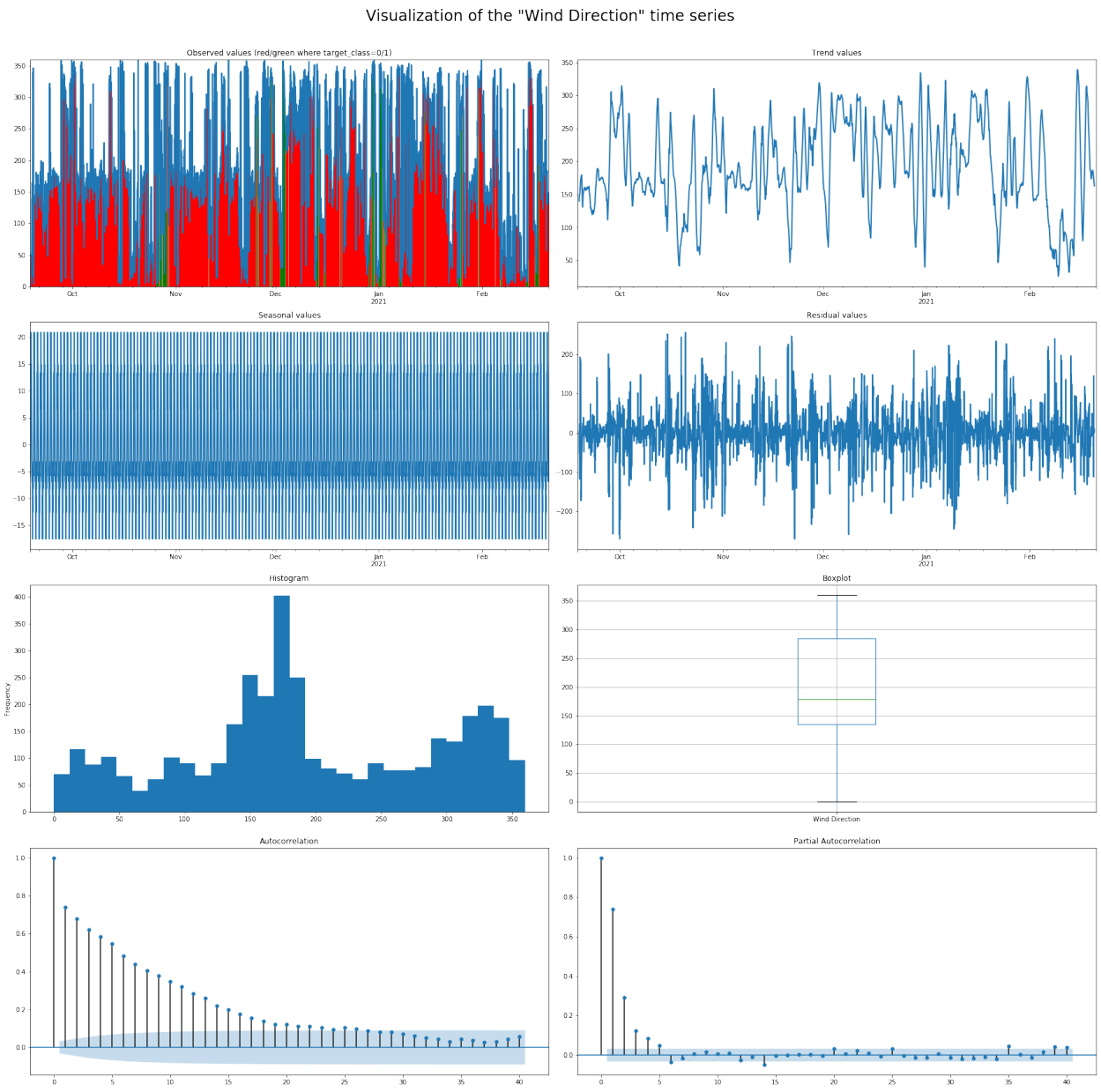


Figure 4-27. Wind direction attribute EDA.

High correlation between temperature values can affect the modeling process negatively. This multicollinearity [28] effect occurs when one predictor variable in a regression model can be linearly predicted from others with a high degree of accuracy. Results can be misleading, however. For the research reported in this thesis, temperature declines are monitored only during cold fronts. Correlation value will be low, as surface temperature will not follow other temperature sensors.

Figure 4-28 demonstrates how the correlation pattern does not work for a winter event. On February 18^{t}, 2021, BST and RST crossed the freezing level (See black line) and ambient temperature did not (See blue area). Snow melted even though ambient temperature indicated perfect conditions for snow accumulation.

At 6 a.m. on February 11, 2020, a deadly 133-car pileup occurred in north Texas on I-35 due to slick road conditions [29]. Figure 4-29 shows Love County Station 1 temperature data. BST and RST crossed the freezing line for few hours while clear skies allowed solar energy to melt an ice layer, making road surfaces very slippery. Ambient temperature remained within a range of 20° to 29°F., This phenomenon indicates the importance of data and the ability to capture temperature differences during potentially hazardous conditions.

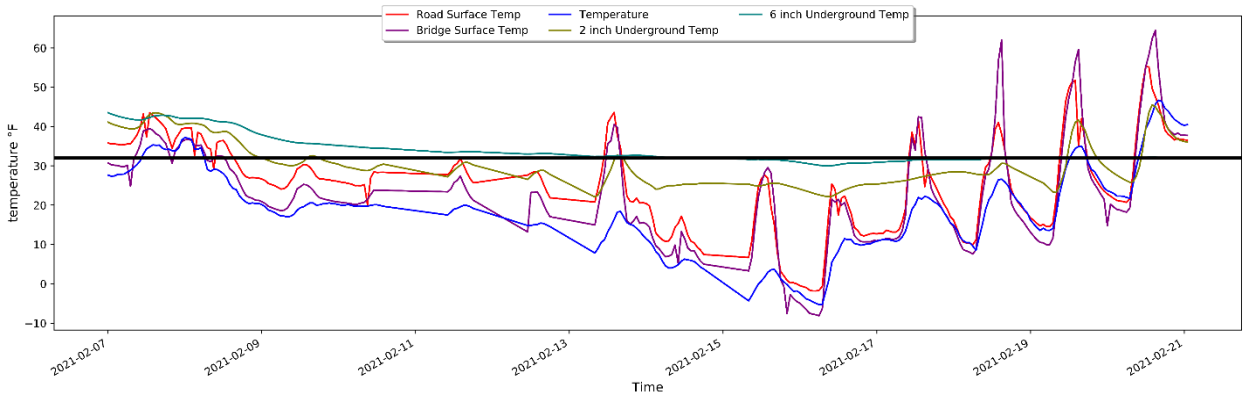


Figure 4-28. Temperature readings during February 2021 blizzard (ST235).

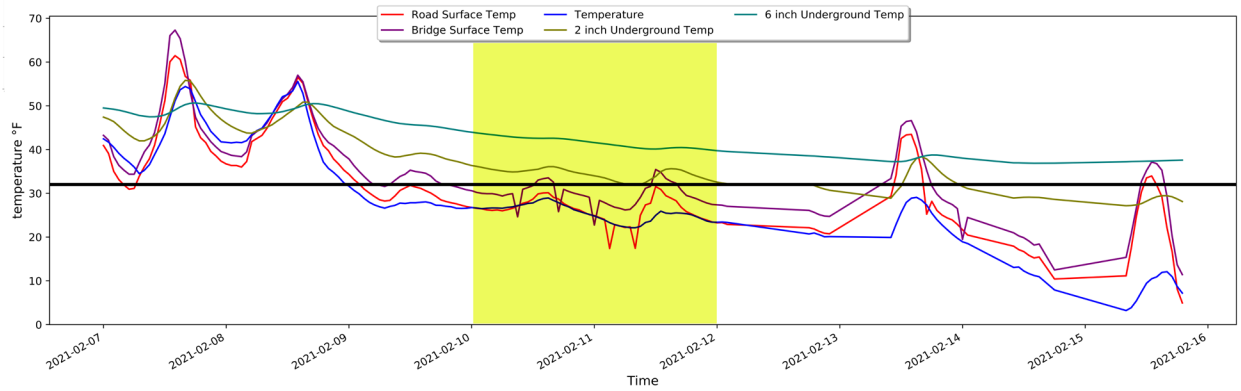


Figure 4-29. Station 1 temperature readings during February 2021 blizzard.

Figure 4-30 shows typical surface temperature distribution along the I-35 highway corridor. Temperatures increase as long as vehicles travel from north to south. This scenario is very different during a wintry event (See Figure 4-31). Snow and ice accumulation varied on each road segment and affected road surface temperatures accordingly. The highlighted square in Figure 4-31 shows two neighboring segments wherein temperature differs nearly 12° F. Figure 4-32 offers an explanation of the difference, as Segment ST154 is cleared from snow while ST165 has snow accumulation remain on the road.

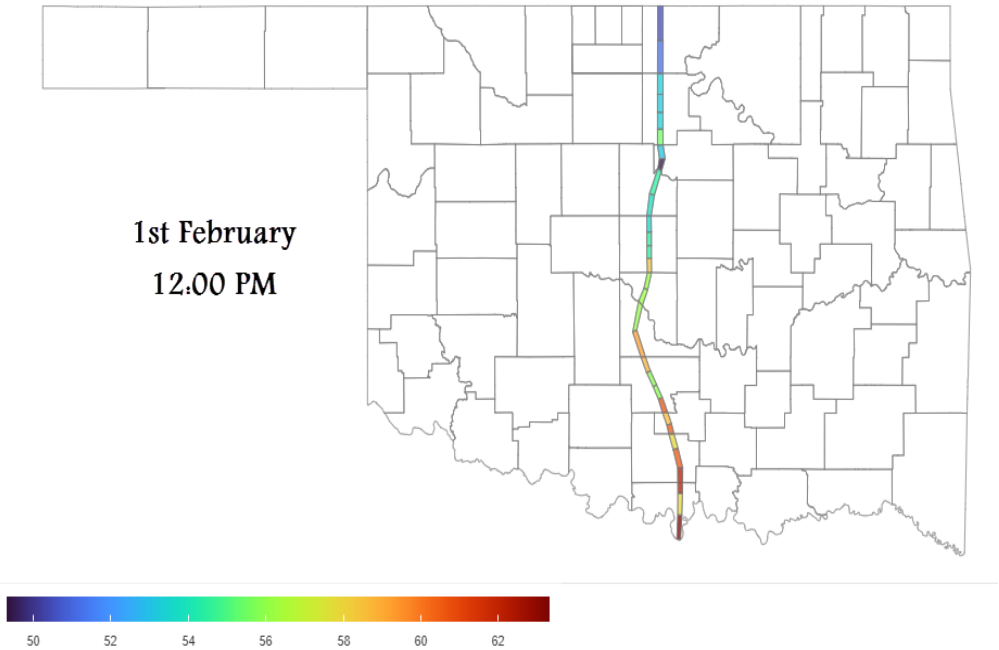


Figure 4-30. Surface road temperature on a typical day.

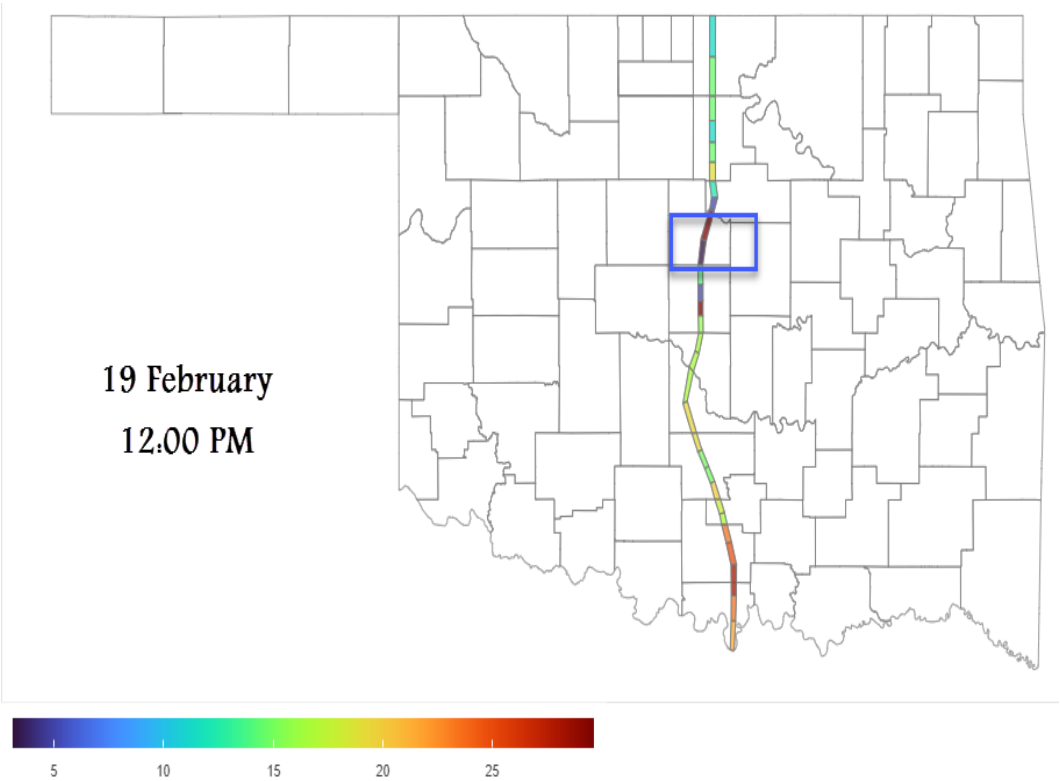


Figure 4-31. Surface road temperature during a winter event.

Figure 4-32. Road view difference between neighboring stations 154 and 165.

4.5 Feature Extraction

Data must first be processed for uniformity in robust time series analysis, as the model only analyzes data by according to row, not column. As such, data must be reshaped so that the model can analyze it throughout a time series. Data engineering must be applied to convert a time series

into a supervised algorithm that a model can developed. Two types of modeling (e.g., regression and classification) are examined in this thesis. Each model must have data reshaped in a specified format to predict temperature value and road condition class.

4.5.1 Regression

In a normal regression problem, a row of historical data is entered into the model in real time for predicting future output based on data array values. Based on the data examined for the research reported herein, output had to be predicted, even though a historical input sequence was not available. Hence, a converted supervised learning sequence was utilized to overcome this issue. Figure 4-33 shows the dataset. The blue columns represent variables, and the green columns represents target value. V(t) is the current time. the range from v(t) to v(t+n) represents the future, while v(t-n) represents the past. It is important to note that the machine learning model only learns sequence from rows, not columns. A regression model must learn attribute trends to overcome this issue.

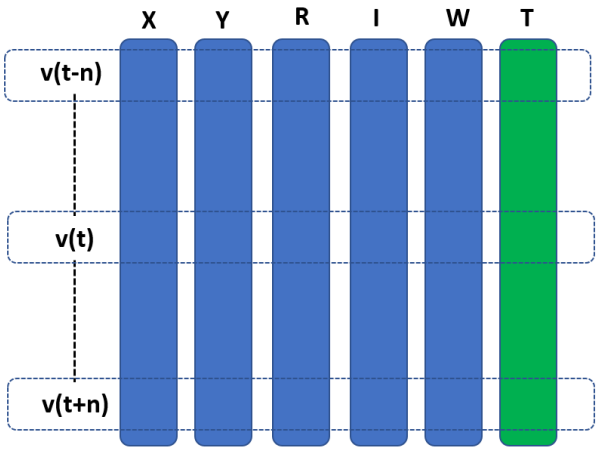


Figure 4-33. RWIS data layout.

To accommodate this, the data frame was changed to the sequence shown in Figure 4-34. All input variables were added in one row in a repeated sequence for a specific past range to ensure that the model captured the past data trend. The goal was obtaining the BST or RST future attributes for a

specific range in one row. Each row had one step forward to make available additional data and training sets.

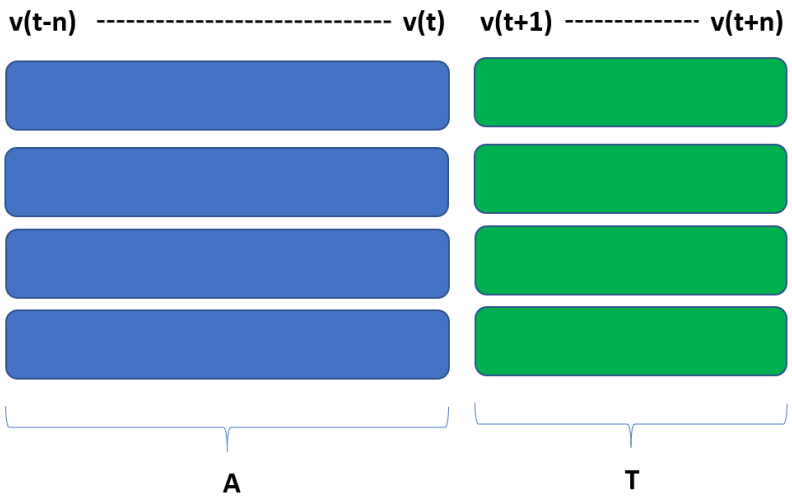


Figure 4-34. Time series transformed into supervised learning dataset.

The more futures target attribute is added, the less accurate the result will be; therefore, various inputs and outputs were used to study model performance. The following Functions 5 and 6 served as data adjustment. Function 5 split the data based on the amount required for prediction need to predict (i.e., one to 12 hours. Function 6 shift the data based on given hours in 5-minute windows.

```
n_in, n_out = SV_5min (6) (3)
```

```
df = shift_data (data = 'values', n_in=n_in, n_out=n_out) (4)
```

Function 5: Calculate range of historical and future timeseries data

Function SV_5min (value)
Define: Switcher from 1 to 12
Define 1: n_in=16, n_out=13
Define 2: n_in=28, n_out=25
…..
Define 12: n_in=148, n_out=145
Return n_in, n_out

Function 6: Convert timeseries data into supervised dataset

Function Shift_data (DF, From, n_in, n_out)
If DF **is** List
Else DF **is** shape [1]
Append: col list, names list
For each *i* **in range** n_in, 0, 1
Add Append: cols **Shift** DF **in** *i*
V **(**t-(*j+1, i)*
For each *i* **in range** 0, n_out
Add Append: cols **Shift** DF **in** -*i*
V **(**t+(*j+1, i)*
DF → concat col
Remove: Nan
Set Index: names
Return DF

Table 4-4
Precipitation Event Count Recorded in Historical Dataset

Index	V1(t-20)	…	V1(t)	V2(t-20)	**…**	V2(t)	V16(t+1)	…	V16(t+12)
1	25.25	…	23.47	1.32	…	1.57	27.42	…	29.42
2	17.67	…	17.24	1.87	…	2.11	18.68	…	19.41
...	…	…	…	…	…	…	…	…	…
120	45.36	…	44.14	4.89	…	4.59	35.47	…	46.31
121	38.78	…	37.21	3.21	…	3.87	41.96	…	48.21

4.5.2 Classification

Using a classification model on time series data is challenging and requires innovative approaches to solve (e.g., some hand crafting features from the time series data based on fixed-sized windows). The only solution is using LSTMs, which provide state-of-the art results. Another challenge is the unbalanced data variation due to lack of data for hazardous conditions over the last two years. Figure 4-35 shows over 350 safe events versus only around 50 hazardous events. Notably, some models do not work properly with imbalanced data, and results will be characterized by a poor f-score, primarily because the model predicts all events as safe and ignores hazardous conditions.

For the work reported in this thesis, the number of safe conditions was reduced, and all hazardous events were combined into one “dangerous” class.

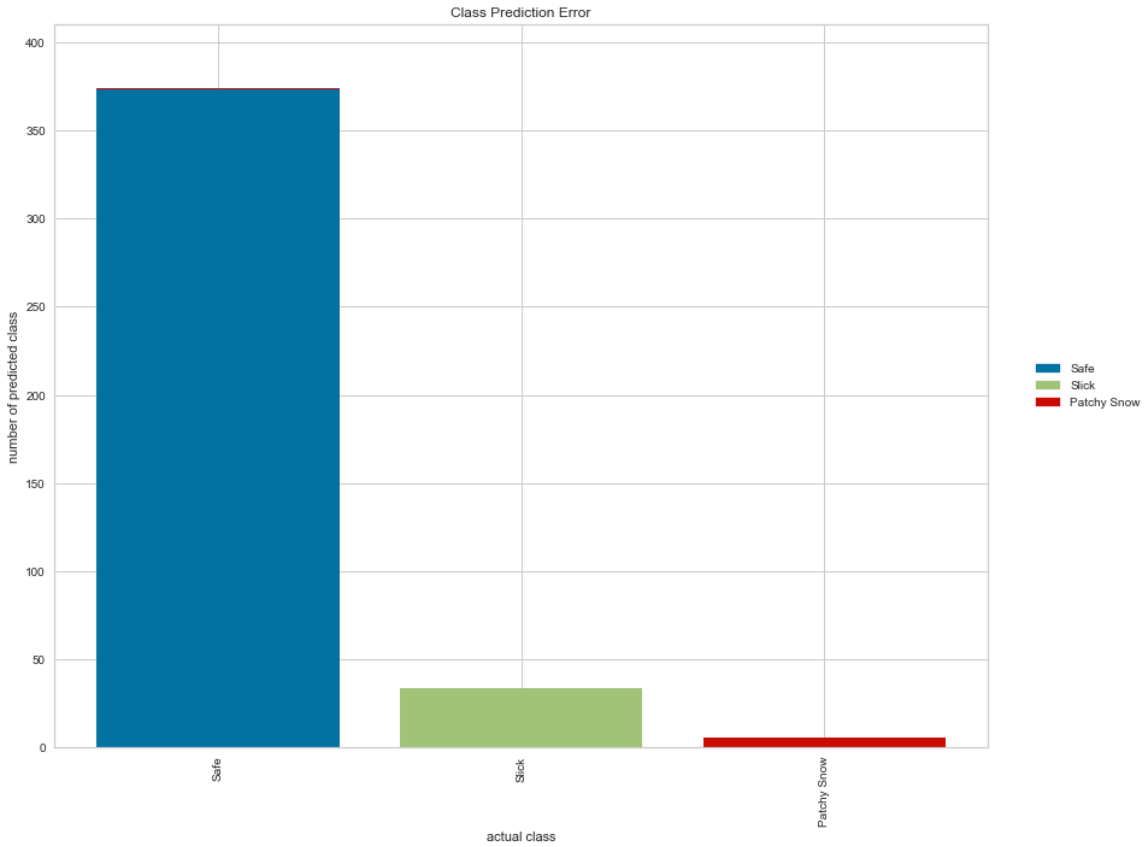


Figure 4-35. Target class count per event.

A parallel coordinates plot (i.e., a multi-dimensional feature where the vertical axis is duplicated horizontally for each feature) was used to allow the visualization of many dimensions in one image. Due to the tremendous range between the numerical value of each variable, Figure 4-36 does not clearly show the diversity between features. For example, brightness ranges between 36,000 and 10,000 while precipitation ranges between 5 and 0.1. Hence, data were normalized and scaled to obtain a clearer picture. Figure 4-37 shows each feature ranging from -5 to 25, where feature mean is set to zero and each feature has a unit variance between 1 and -1 applied. The graph provides a

better sense of feature distribution and informs if any attribute has a high value with respect to another class.

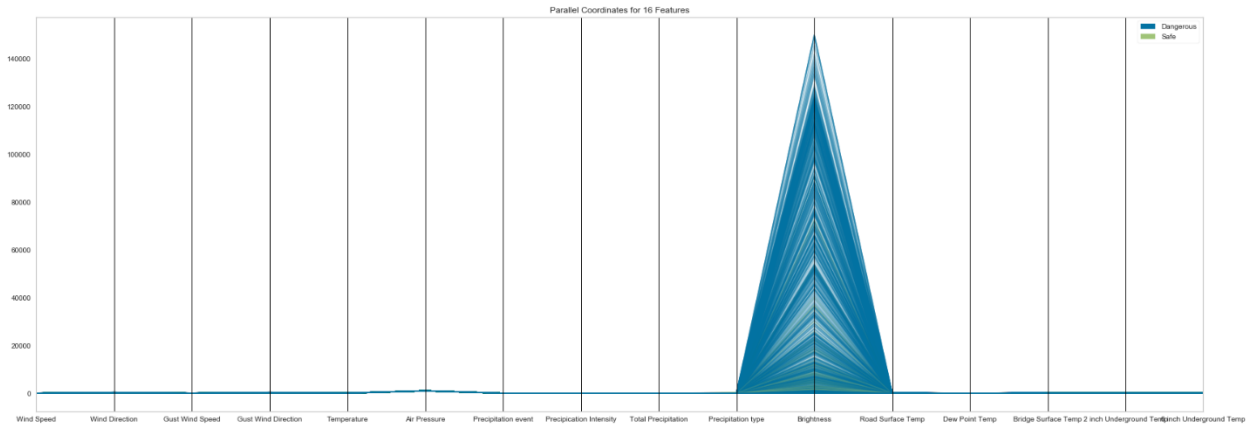


Figure 4-36. Parallel coordinates for RWIS features before rescaling.

Figure 4-37. Parallel coordinates for RWIS features after rescaling.

As previously mentioned, all attributes were rescaled and transformed for classification. Next, a percentage change between a current and prior element was applied to indicate the significant changes in each row and correlate each to conditions recorded at the same time. Figure 4-38 shows the changes in wind direction, precipitation value, and temperature. PCT change boosted the model's accuracy; this will be discussed in Section 5.

Figure 4-38. Scaled, transformed, and PCT attributes.

After preprocessing the dataset for use in the classification models, a data analysis was performed to compare attributes and classes. First, a correlation study was conducted using Pearson correlation similar to the one discussed in Section 4.4. Figure 4-39 shows precipitation attributes, brightness, and temperature have high correlation readings.

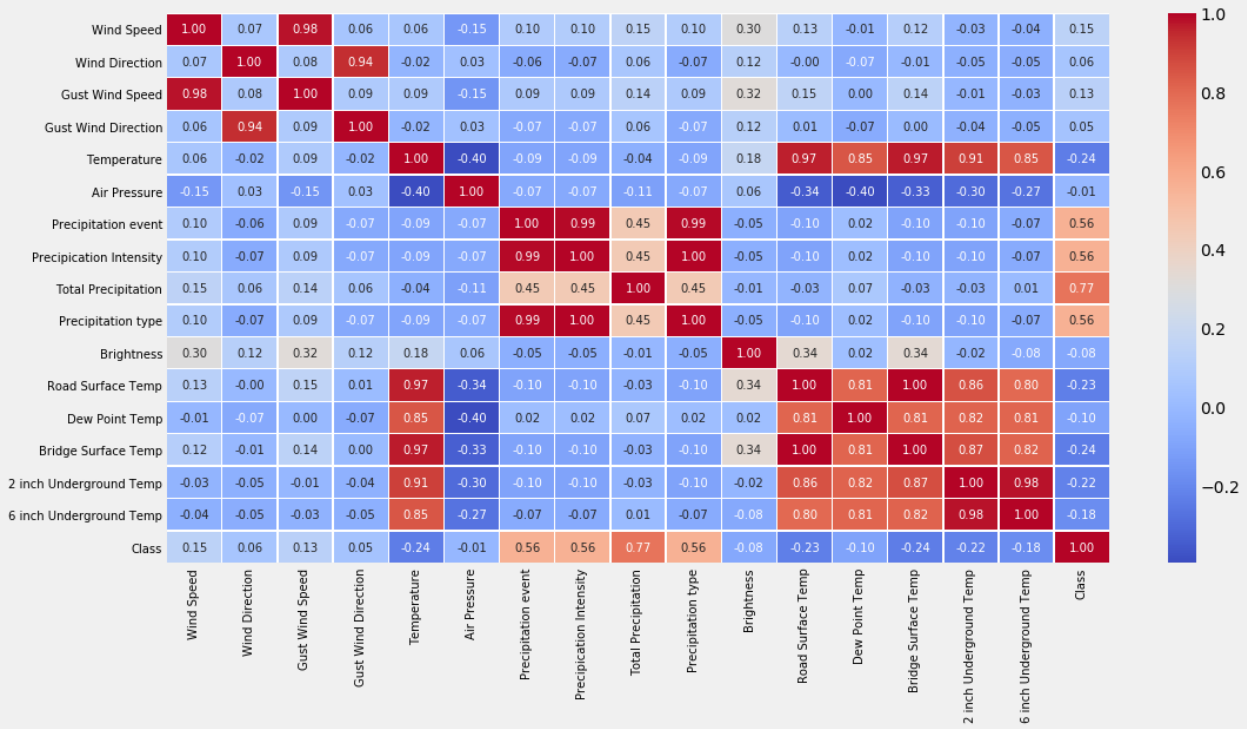


Figure 4-39. Correlation map of data features versus road condition class.

Next, a covariance ranking algorithm was applied, which attempts to compute the mean value of the product of deviations from their means. Covariance detects colinear relation between attributes. When comparing Figure 4-40 and 4-41, one can see how covariance shows more inverse correlation between temperature values and precipitation events. Also, air pressure and change in temperature values show high correlation, proving the relation between pressure, precipitation, and pressure changes when a cold front approaches the highway. Such atmospheric disturbance causes severe road conditions.

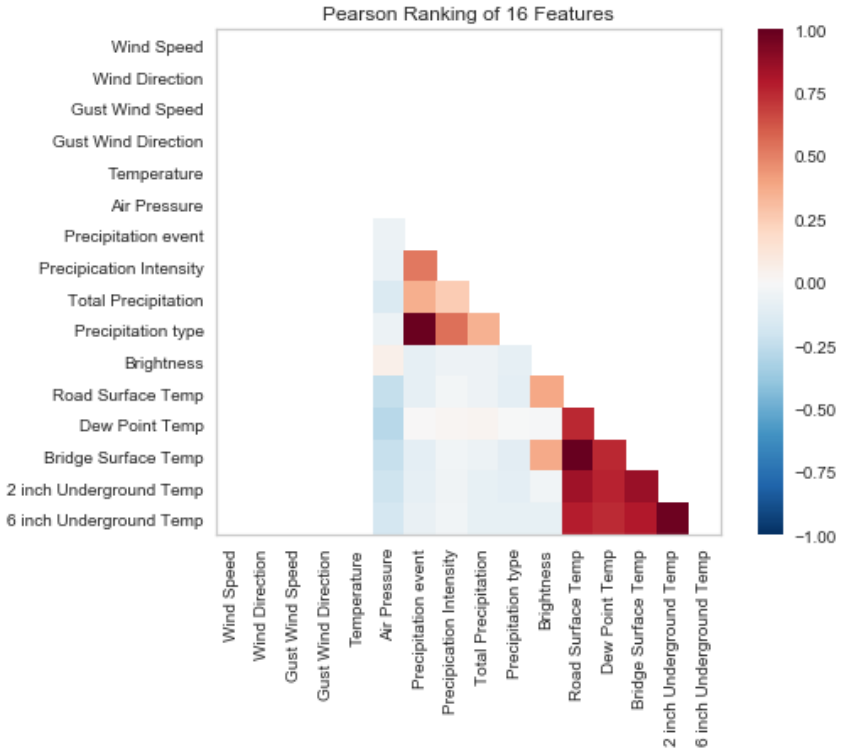


Figure 4-40 Two-dimensional Pearson ranking.

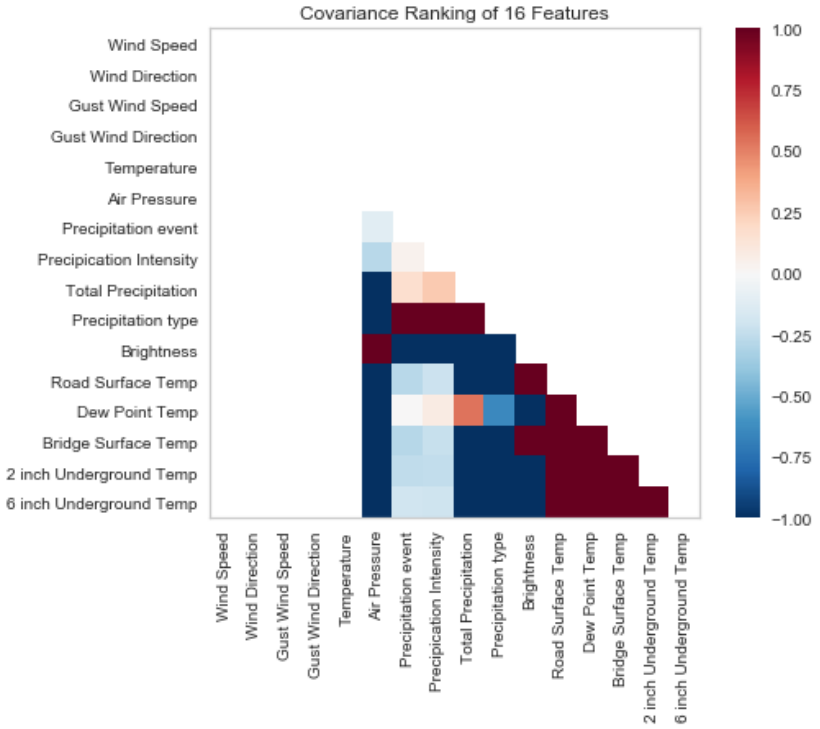


Figure 4-41. Two-dimensional covariance ranking.

Another ranking method (i.e., one-dimensional ranking for data features versus road condition class) was added to the data analysis. The Shapiro-wilk algorithm was used to obtain the normality of the distribution for each feature, providing an overview of which feature strongly affects model predictions. Figure 4-42 shows that all temperature values have strong rank followed by precipitation values.

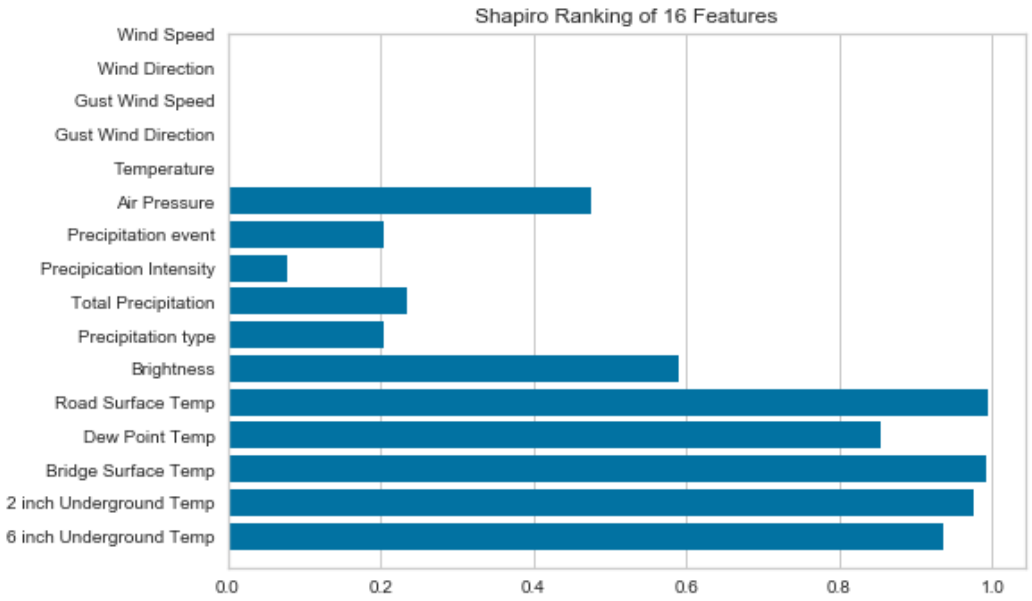


Figure 4-42. Shapiro data ranking features.

Next, a Radviz Visualizer plot [30] is provided. The multivariate data visualization algorithm plots each feature dimension on the circle and shows separability between classes, given an opportunity to detect a clustering method. Otherwise, data is randomly distributed and difficult to classify based on the distribution. Figure 4-43 clearly shows that the safe class tends to be more distributed toward temperature data while the dangerous class is distributed near the precipitation data. Both classes are randomly distributed and difficult to separate when near wind data.

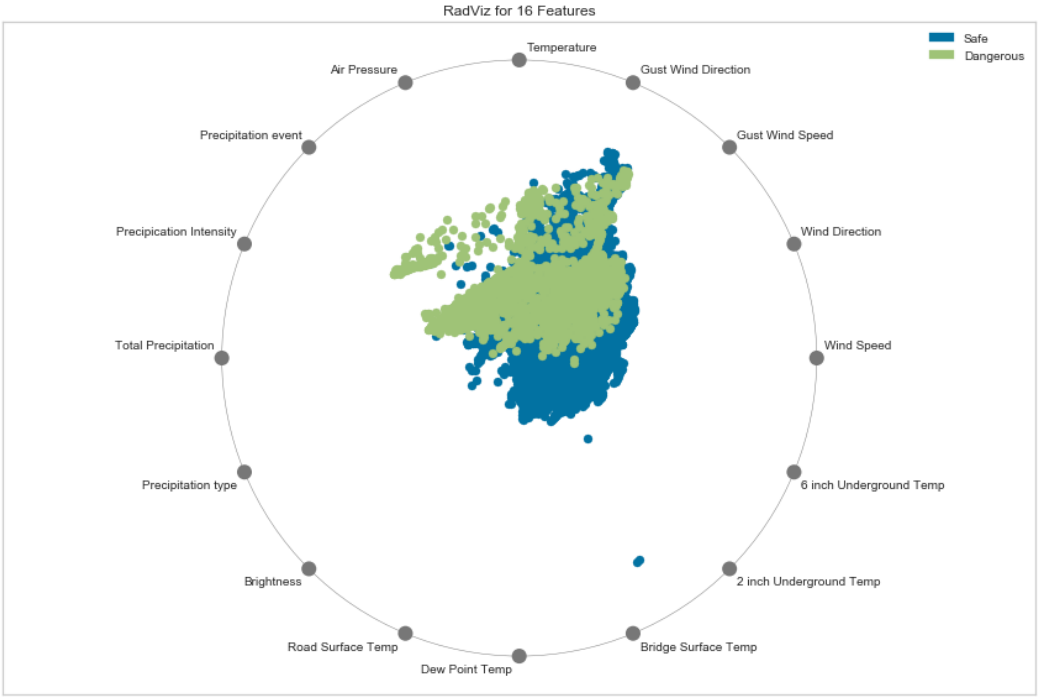


Figure 4-43. RadViz visualizer for features versus class.

For a better overview of data features, a PCA decomposition was used to decompose 16 features into three dimensions so it can be plotted, visualized, and tested; if it can be interpreted to determine if spherical distance metrics can be used in the modeling scheme. Figure 4-44 shows a 3D PCA projection of the classes, the graph shows that the 3D dimensional setup did not show any related variance that we can use in any further analysis.

Next, I did a Manifold Visualization of the dataset. The method output the same result as PCA, but it uses other functions, as it uses nearest neighbors approaches to embedding, while allowing them to capture non-linear structures. The result can show clear separability and make it easier for the model to create decision space. Figure 4-45 shows a promising result using only 3 features to predict the model classes, this Manifold feature is used in Section 5 later in classification method that can be implemented in real time models to have fast accurate predictions.

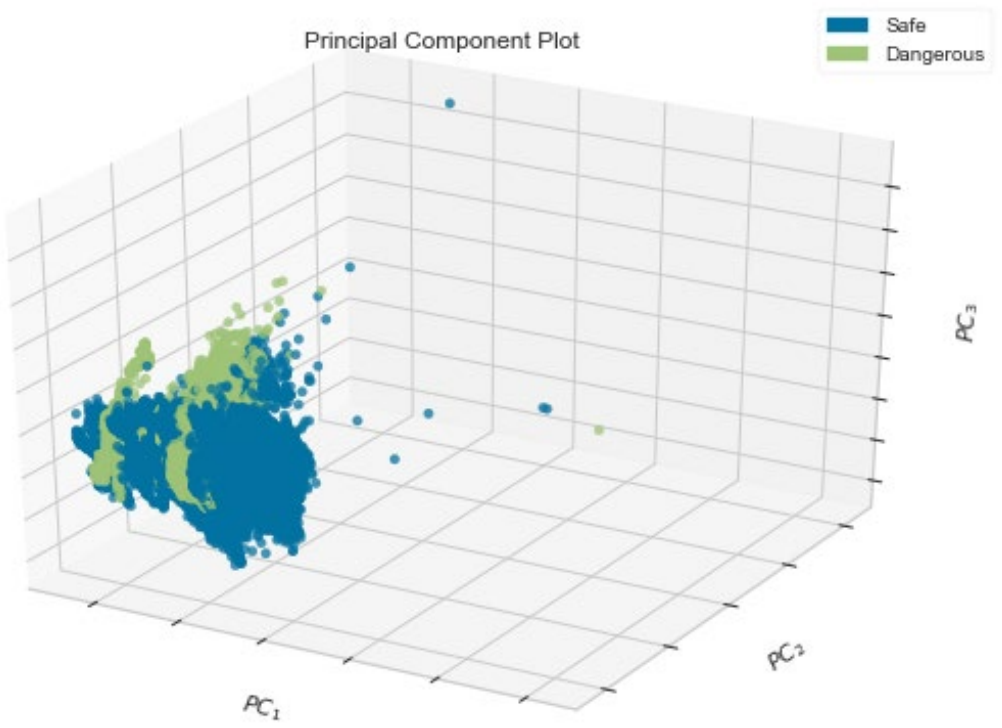


Figure 4-44. 3D PCA visualization.

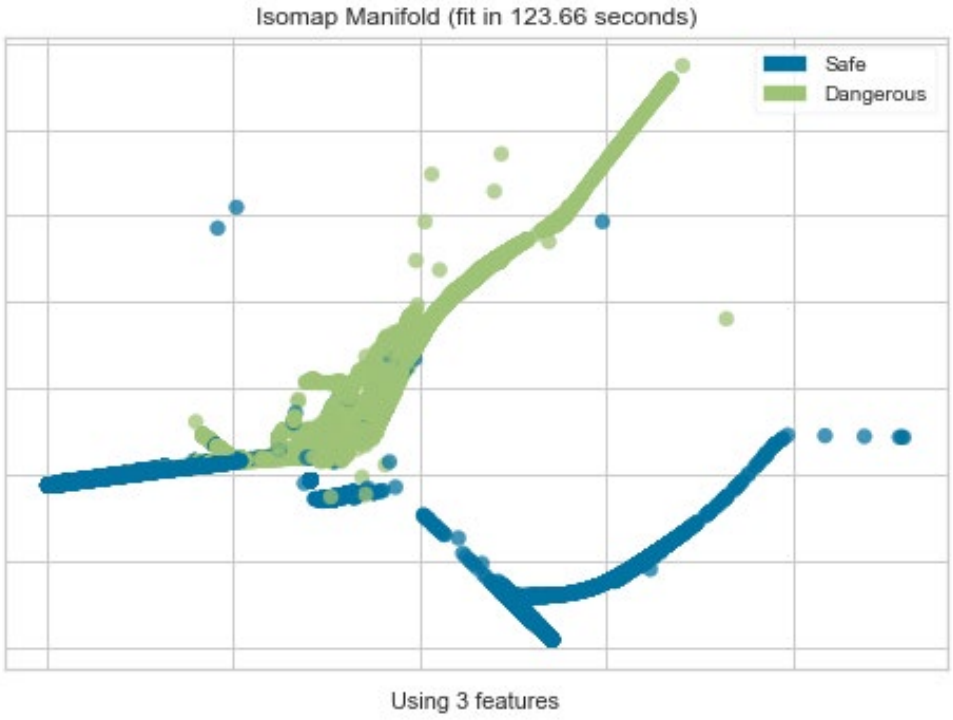


Figure 4-45. Three features manifold visualization.

5 RWIS Model Forecasting

This section discusses all models studied for regression and classification. Regression models predicted road surface temperature in a sequence between one and 12 hours. Model performance was measured in RMSE, as shown in Equation 4. Classification model predicted road conditions in two classes—safe and dangerous. Model performance was measured in accuracy according to Equation 5. Additional score metrics are detailed in Section 5.2.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}} \quad (4)$$

$$Accident\ Detection\ Rate\ (\%) = \frac{True\ Positives}{True\ Positives + False\ Negatives}\ x\ 100 \quad (5)$$

5.1 Supervised Regression Models

After converting the dataset from a timeseries format to a supervised format, supervised linear and nonlinear regression models were applied to the dataset. Converted datasets assist models in detecting the seasonal shape of the array in a dataset row. Neural network models were applied as LSTM, which provided the highest performance as a result of the short-term memory application (See Section 5.1.2).

5.1.1 Classic Machine Learning Regression Models

Scikit learn package offers a number of supervised learning linear and nonlinear models. SL is a task wherein the model learns a function that maps an input to an output based on a variety of input to output pairs in the testing portion of the dataset. A simple linear model was first used to verify data quality and indicate performance. Figure 5-1 shows that model data output was overfit due to data leakage in the conversion metric portion from time series to supervised learning. The output target value as the road surface temperature appears in data input and output. Figure 5-2 shows that this value that moves from t-n to t and will appear later in dataset output as t+1. The model

learned this sequence and overfit the data, resulting in an inaccurate perception of low RMSE. The target was removed from the input data, and data leakage did not occur.

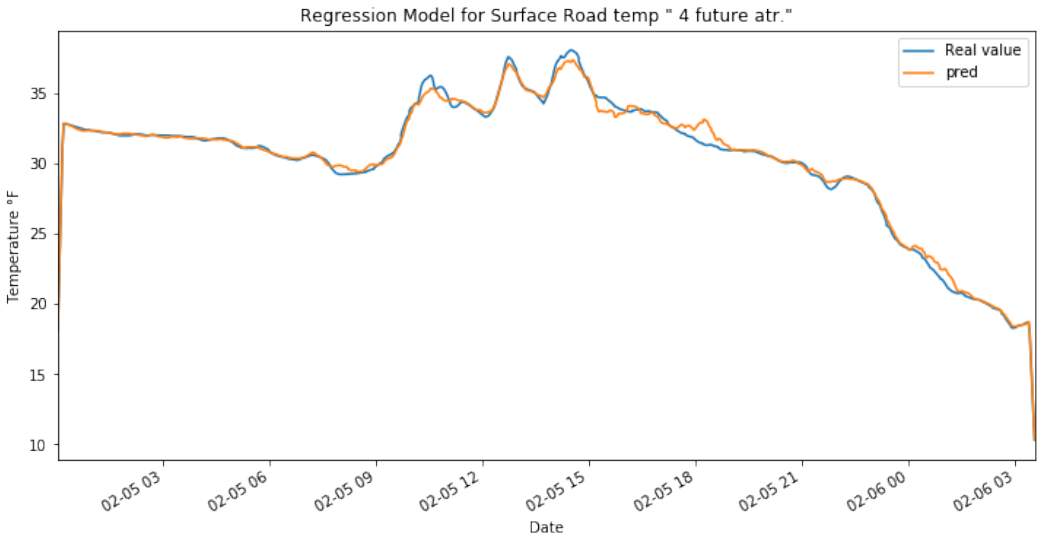


Figure 5-1. Overfitting linear regression model.

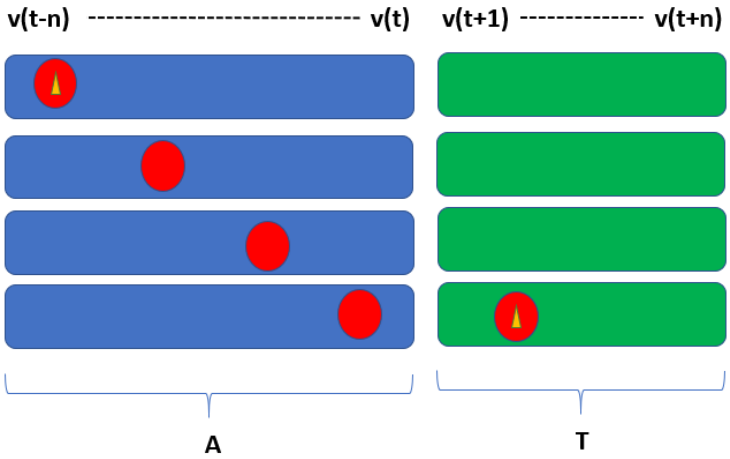


Figure 5-2. Target output data leakage.

Next, five regression models were each trained 12 times with a sequence from one to 12 hours. RMSE value was recorded to show performance at each sequence. Each target had a row of results. For example, if a dataset of six hours was sequenced, a row was generated in 5-minute frequencies, resulting in 72 values, and expressed as [x, x+1, …., x+72]. To achieve an acceptable indicator for measuring RMSE of each row, each point in the row was compared by calculating mean value of

each row, and then calculating total mean value of the dataset. A histogram of RMSE of each row was also plotted to obtain a good indication of model quality.

The Ridge regression model was first trained, as it addressed problems of ordinary least squares by imposing a penalty on coefficient size [31]. As value of α increased, shrinkage increased, and collinearity became more robust. Model alpha was 0.5.

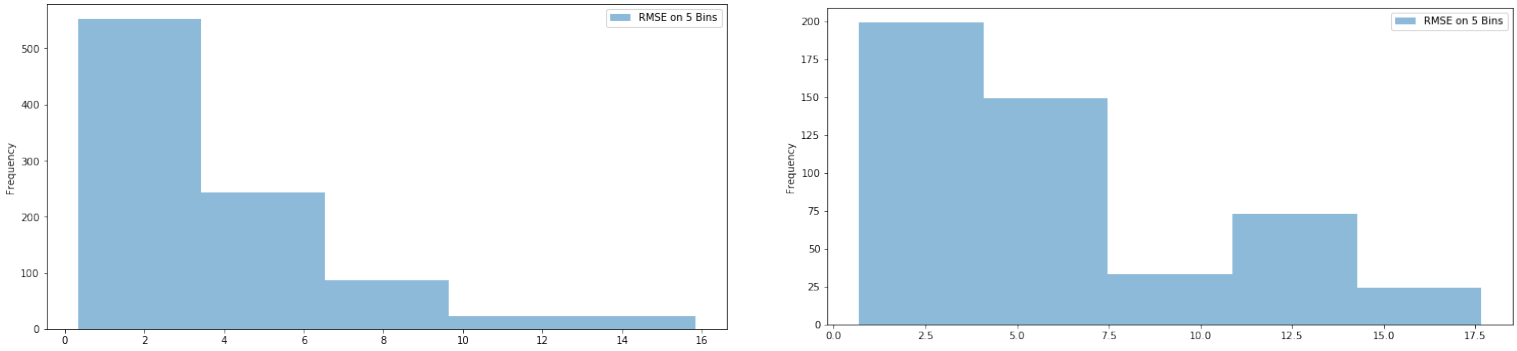
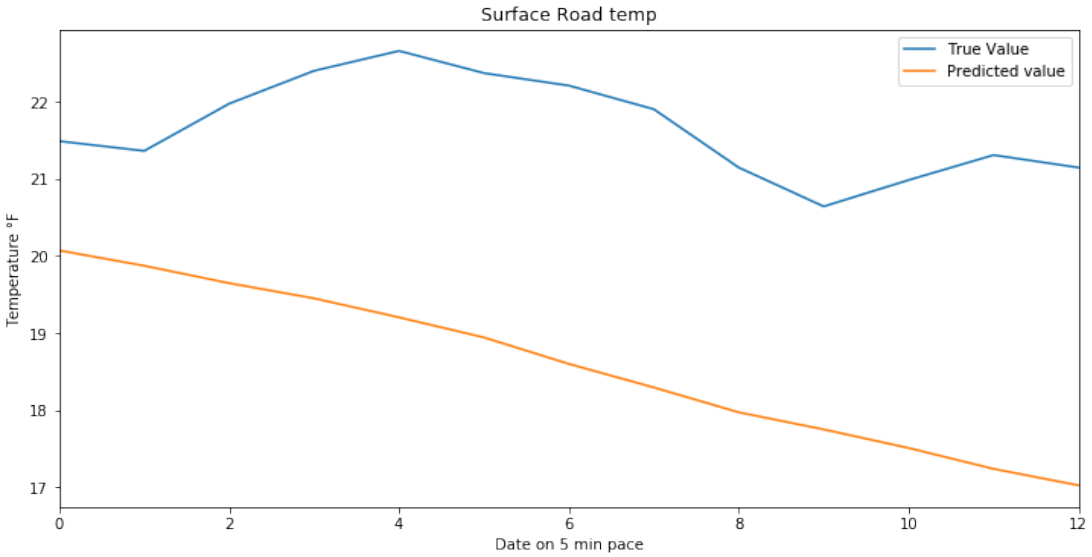
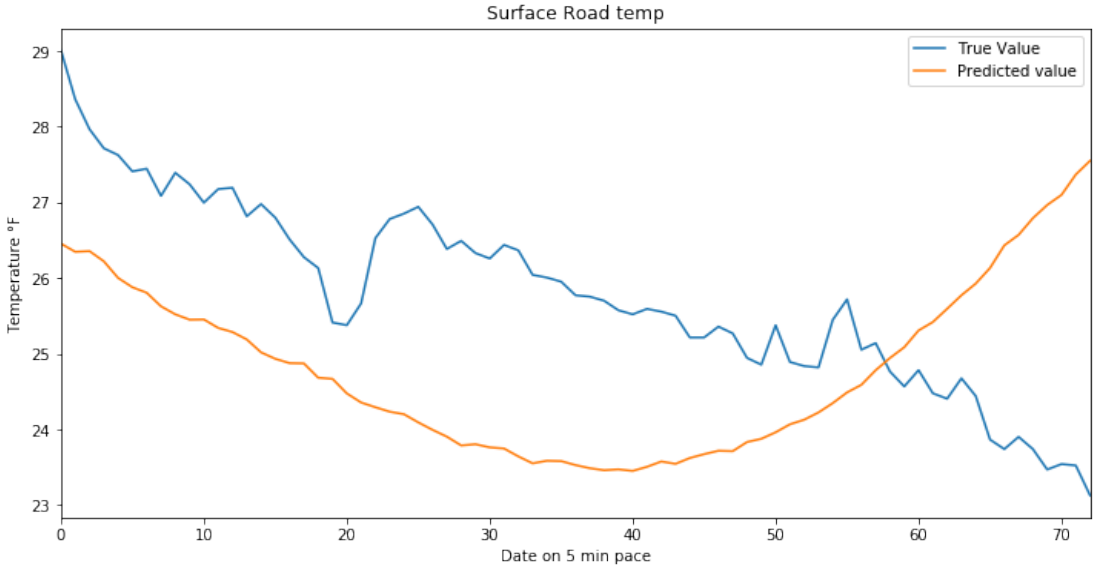


Figure 5-3. Ridge regression RMSE histogram at six and 12 hours.

The model was characterized by a mean RMSE of 3.4 at 6 hours. Figure 5.4 reveals poor performance after the 60-pace mark at five hours forecasting. Also, the model did not recognize knife edges (i.e., the points at which temperature falls as a result of a cold front). These results prove that this model is not acceptable for road condition forecasting.



(a)



(b)

Figure 5-4. (a) three-hour forecast; (b) 12-hours forecast using RIDGE regression

Next model is Lasso regression, using least-angle regression (LARS) algorithm for high dimentional data. The model showed good performance for 6 hours predicition, also it predicted the sharp drops. Figure 5-5 shows how the model followed the tempeature drop in a 12 hours prediction, however it didn't had a good RMSE performance, as there was a gap between the true and predicted value. Figure 5-6 shows the RMSE performance versus forcasted hours.

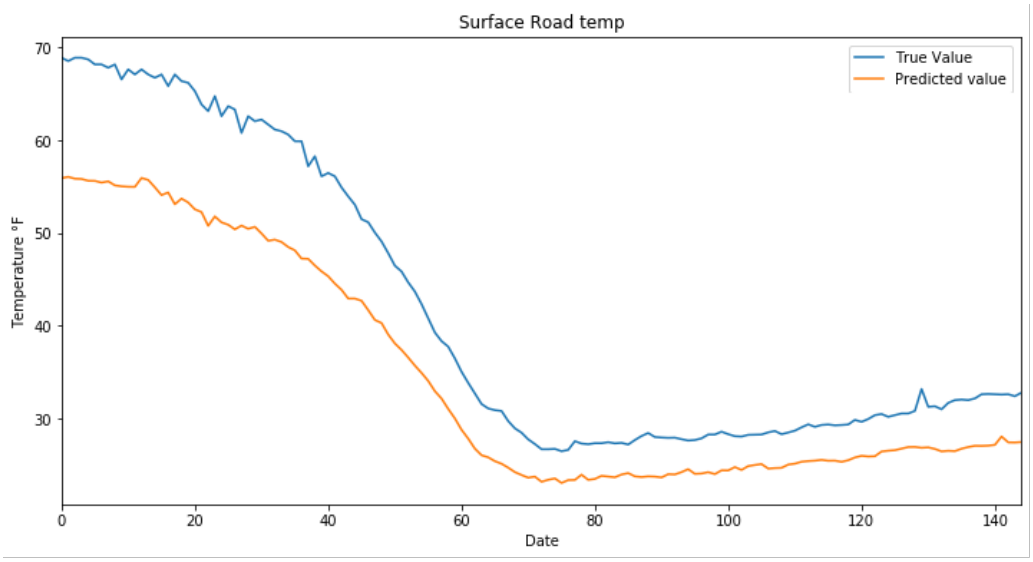


Figure 5-5. 12-hour forecast LARS lasso model.

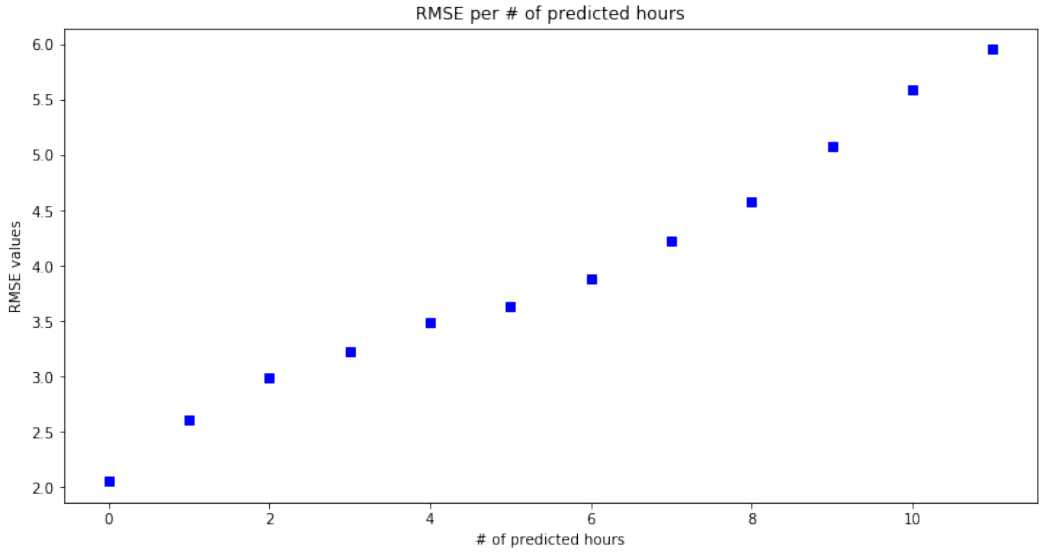


Figure 5-6. Mean RMSE value versus forecasted hours.

The Stochastic Gradient Descent (SGD) model is useful for large samples. Its functionality to fit models for regression using loss functions suggested a logistic model function for training the dataset. While the model showed good performance for the first three hours of forecasting, it did not perform well for the following nine hours forecast (See Figure 5-7).

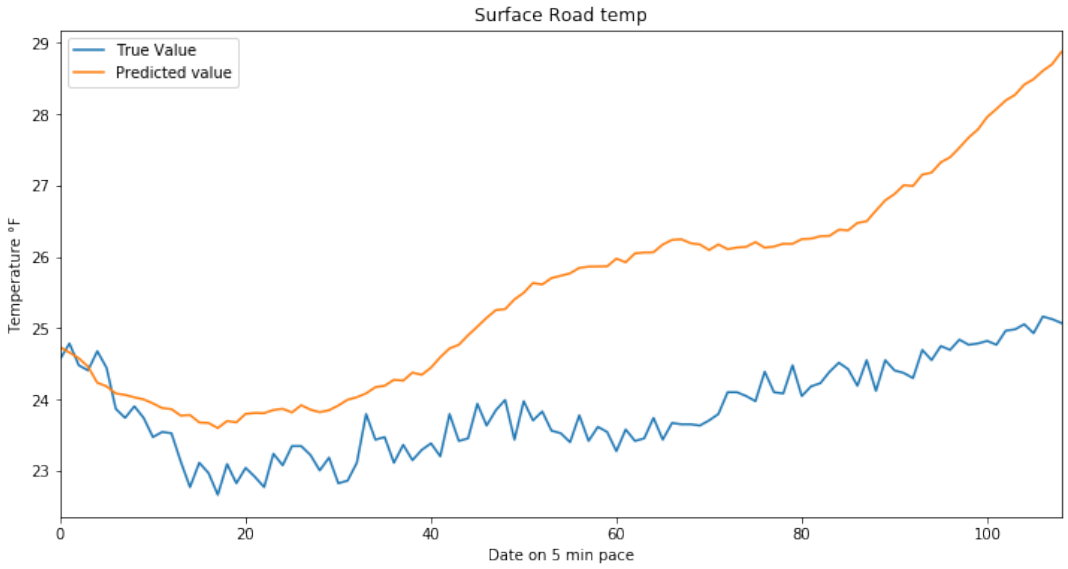


Figure 5-7. Nine-our forecast Stochastic gradient model.

Next, a powerful ensamble regression model —based on a randomized decision tree regression model— was trained. The model combines multiple outputs from a weak classifier and determines a final regressor based on majority vote from multiple decision tree regression models. The model uses 34 estimators. The estimators' value was selected based on the model accuracy, as it will stop getting better at a certain number of estimators.

The random forest showed good results for mean RMSE for all forecasting hours invtervals. A histogram (See Figure 5-8) reveals that some specific trends are not detected by the model. The histogram has several RMSE trend bins above 5.0. Figure 5-9 illustrates that a quick tempearture rise in the afternoon was not detected by the random forest model, indicating a lack of promising performance overall.

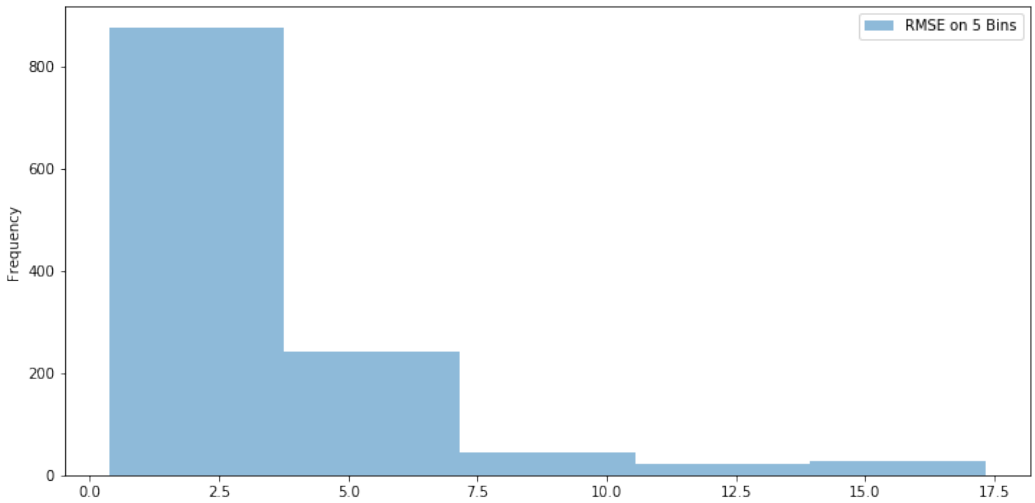


Figure 5-8. RMSE histogram at 12-hour random forest.

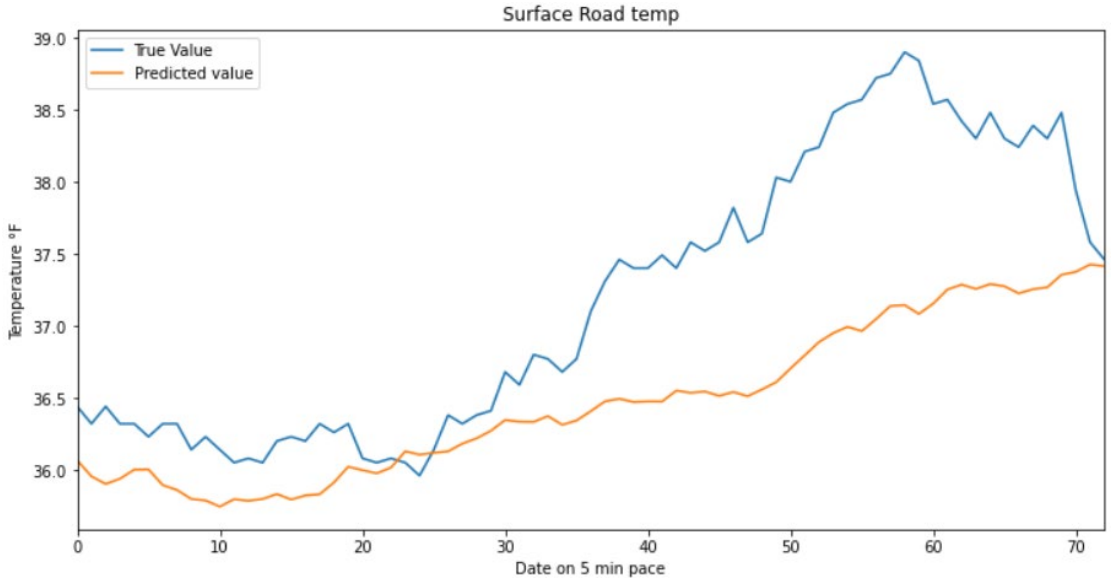


Figure 5-9. Six-hour forecast random forecast.

The kernel ridge model was evaluated last and outperformed all other regression models. This model combines ridge regression with kernel trick and corresponds to a non-linear function in the original space. Kernel ridge is similar to SVR, although it is characterized by different loss functions. The model uses squared error loss combined with l2 regularization. Figures 5-10 and 5-11 show the forecast for six and 12 hours. It is important to note that the model selects the low

temperature as it rises and crosses the freezing level. The graph demonstrated good performance for forecasting six- and 12-hour projections.

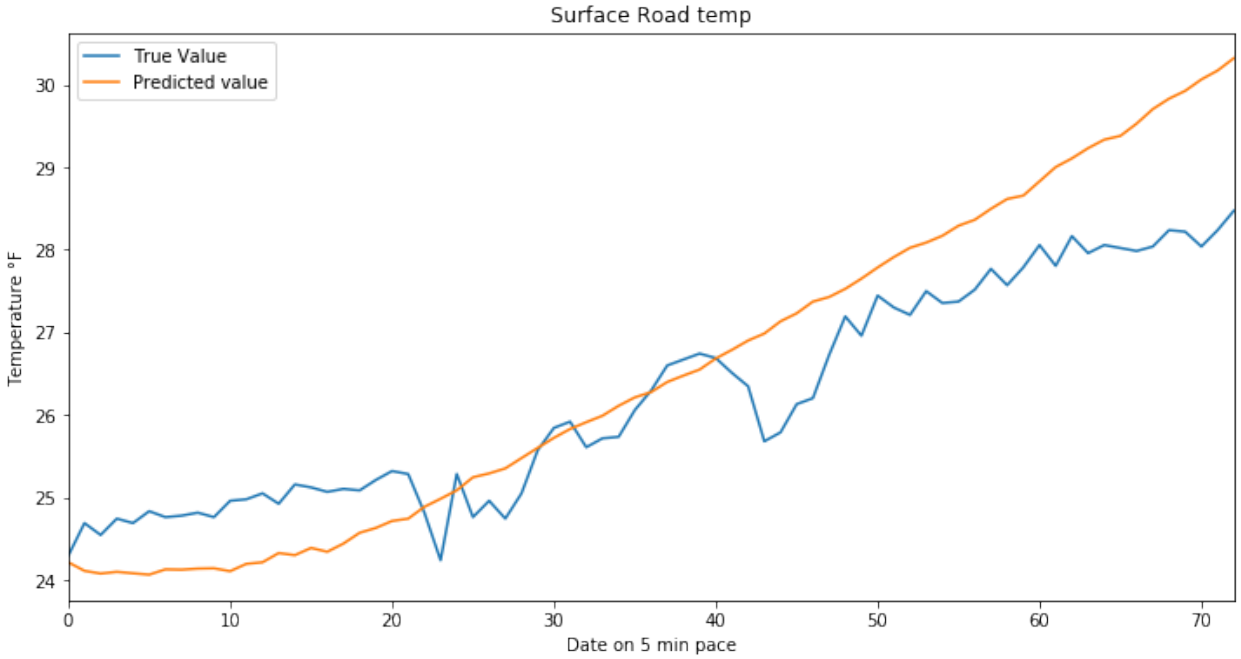


Figure 5-10. Six-hour kernel ridge forecast.

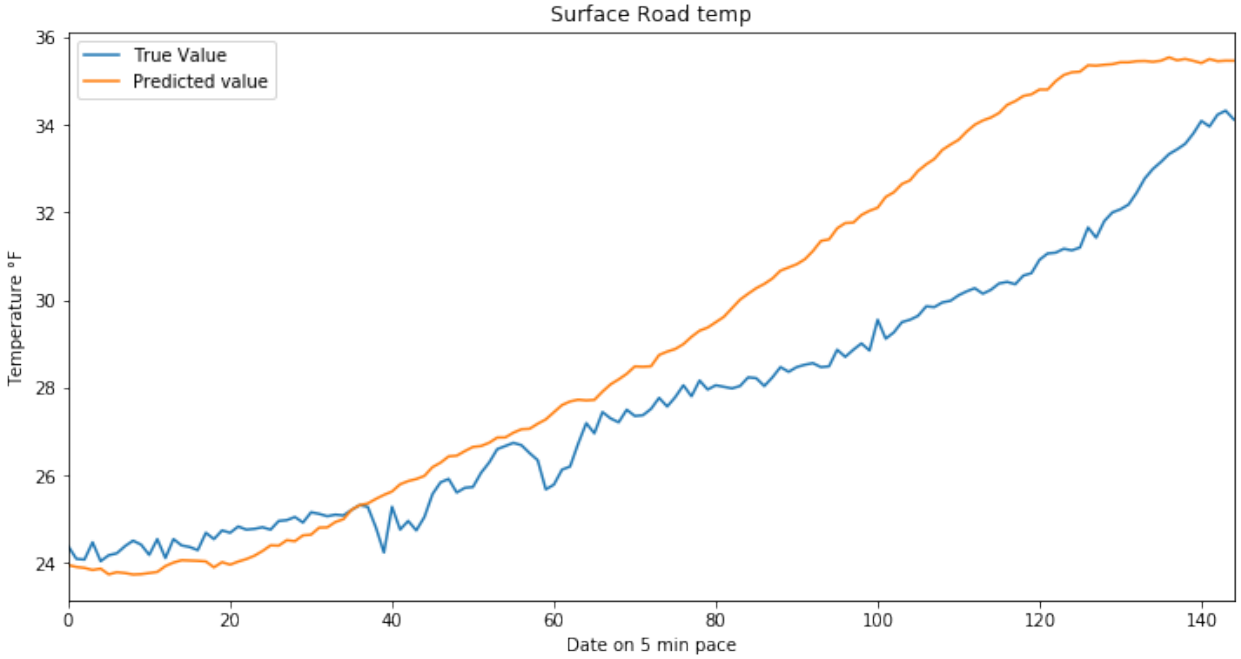


Figure 5-11. 12-hour kernel ridge forecast.

Figure 5-12 displays the overall performance of each model described above. The y-axis represents mean RMSE value of the model; the x value represents the forecasted hour for each model. The kernel ridge (See red line) proved to be the best model.

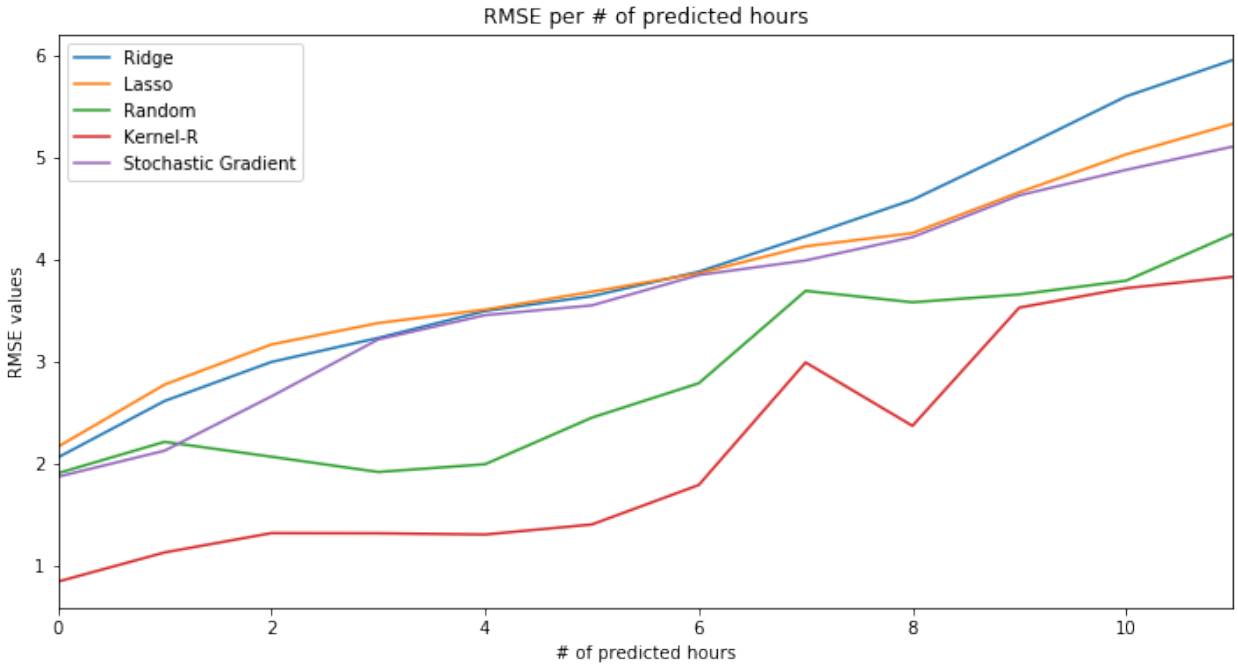


Figure 5-12. RMSE value of all models versus forecasted hour.

5.1.2 LSTM Recurrent Neural Network Model

Recurrent neural network (RNN) is considered one of the best models for time series dataset prediction. RNN is an optimal neural network method for regression. The model mechanism ensures that current output is dependent on previous information [32]. Given that RNN depends on backpropagation through time, gradient gradually becomes an issue. Architecture depends on long short-term memory (LSTM) [33] to manage the gradient problem via a constant error flow. This process will also manage the complex long time-lag problem. The difference between RNN and LSTM is the internal gate of LSTM wherein three gates determine the weight and importance of each time-step value for information. RNN information flows through a tanh function, thus cannot retain prior information and fails to follow data array trend. Figure 5-13 indicates that

LSTM has three additional neural network layers interacting in a unique way. The sigmoid layer output numbers between zero and one describe the amount of each component that should pass through. LSTM uses three components to protect and control cell state.

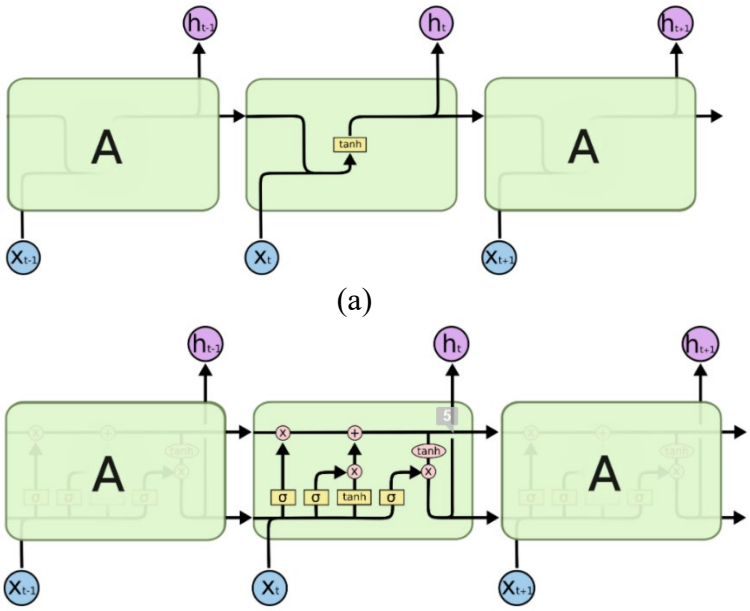


Figure 5-13. (a) RNN; (b) LSTM internal structure.

LSTM architecture was designed through trial and error and by comparing scoring metrics as RMSE and histogram distributions. Table 5-1 details the model architecture. The model begins with an input of 128. Data then passes through three layers of neurons at a size of 256 and a dropout layer of 0.1 to prevent overfitting. The training set has 500 epochs with an early monitoring function to stop training at a threshold of 1×10^{-4} . Loss function was mean absolute error. The final output layer has a size of 121.

Table 5-1
LSTM Model Architecture

Layer Type	Output Shape	Param	Activation Function
LSTM	(None, 128)	671	Relu
LSTM 1	(None, 265)	3300	Relu
LSTM 2	(None, 265)	45692	Relu
Dropout	(0.1)	0	-
Dense 1	(None, 121)	257	adam

The loss curve (See Figure 5-14) provides a snapshot of the training process and the direction wherein the network learns. The figure shows a good learning rate without over- or under-fitting.

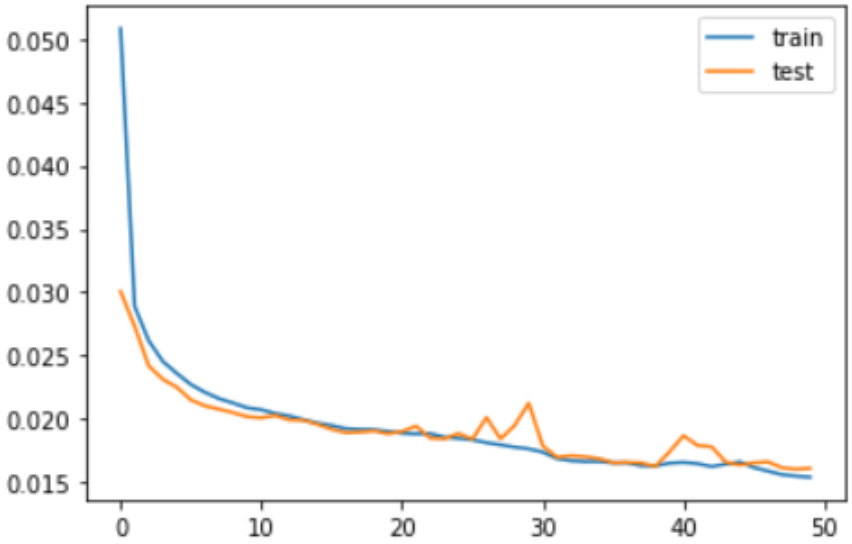


Figure 5-14. LSTM model loss curve,

The model outperformed overall for forecasting results. Figure 5-15 indicates a very low RMSE distribution in the histogram for the three- and six-hour forecast. The model also outperformed for the nine- and 12-hour forecast. Table 5-2 shows the mean RMSE value for the four forecast hours.

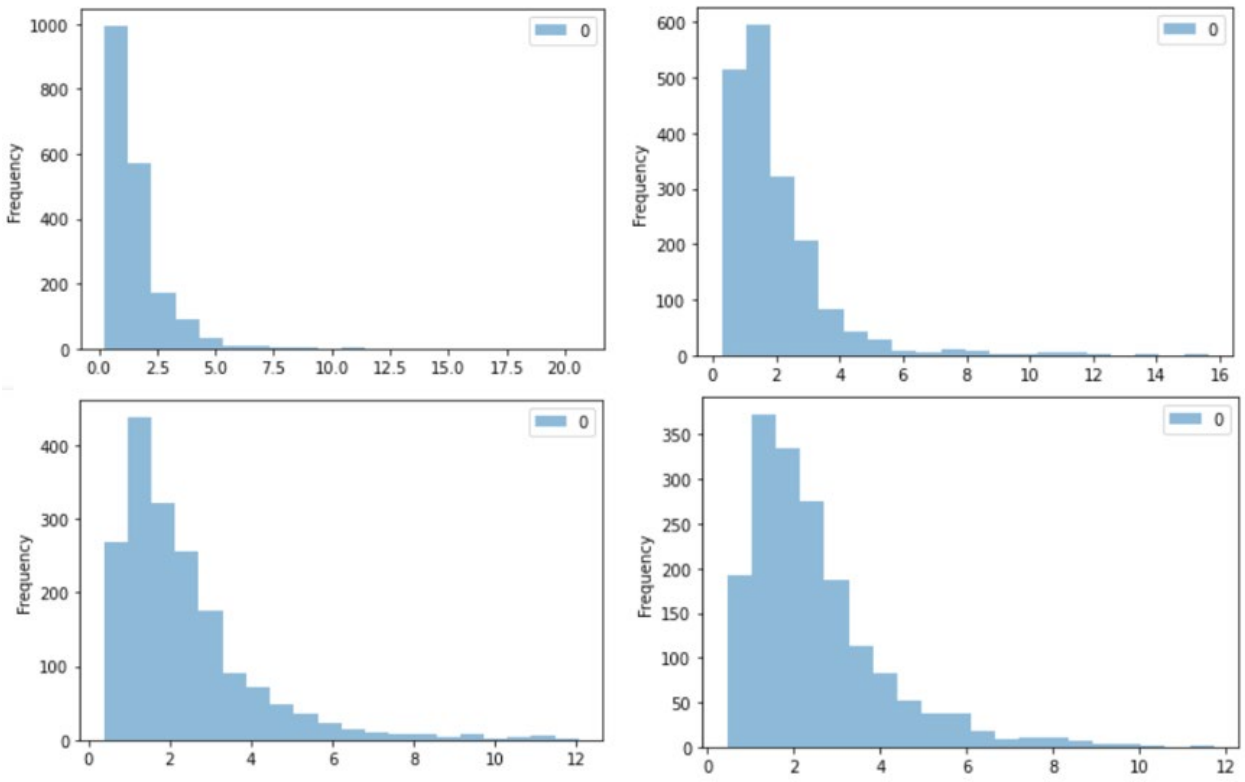


Figure 5-15. Histogram distribution of three-, six-, nine-, and 12-hour forecast from LSTM model.

Table 5-2
Mean RMSE Value of LSTM Forecast Hours

Forecasted Hours	RMSE value
3-Hour	1.569
6-Hour	1.957
9-Hour	2.328
12-Hour	2.505

The model also captured falling temperatures in mid-day when cold fronts dropped temperatures to minimum values. Figure 5-16 illustrates a 20°F decline on February 6, 2020, prior to the blizzard.

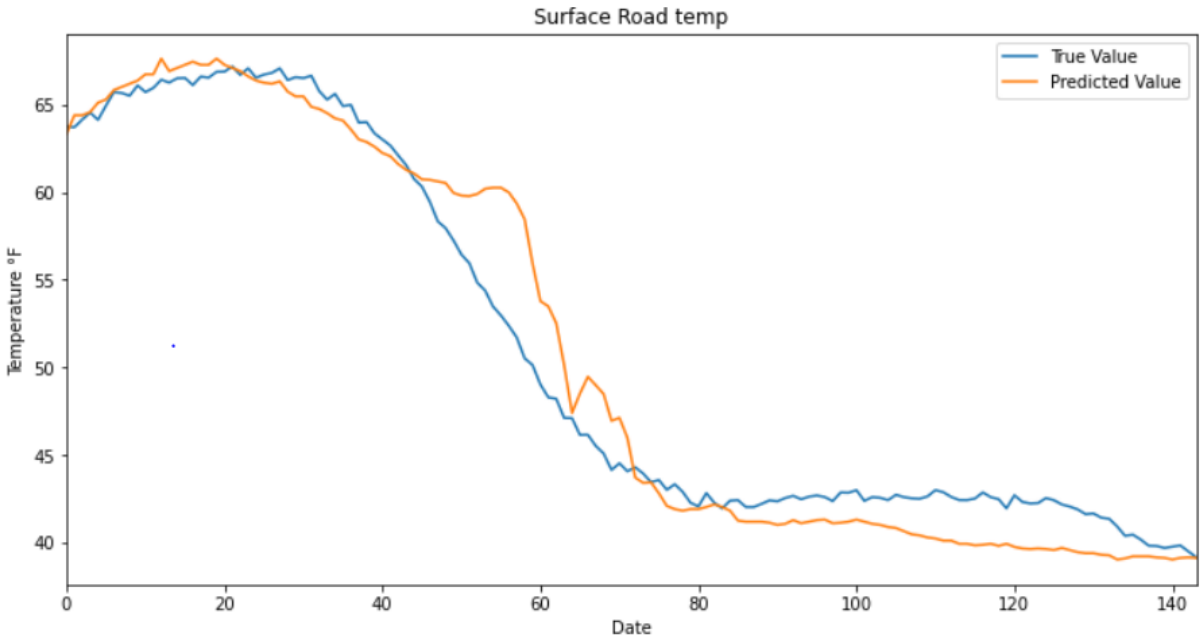


Figure 5-16. LSTM model 12-hour forecast.

5.1.3 Ensemble LSTM

Section 4.1 describes the way in which this research uses data from six stations to train the models. This section details how models trained from all six stations were combined to output a single model. The resulting model was tested individually in hopes of generating a reliable international model that can fit any station. The resulting technology should function with any variation of data sans limiting factors. To build an ensemble stacked model, six models were merged into one (e.g., ST1, ST2, ST3, ST4, ST5, and ST6). Each model will have its own hidden layers, yet and output results in a combined layer. Leveraging the merging function in the Keras package output the ensemble model depicted in Figure 5-17.

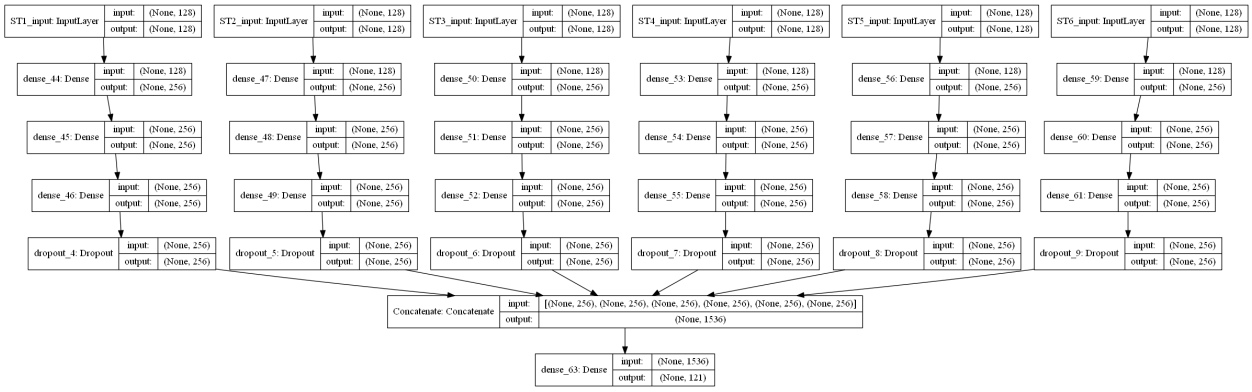


Figure 5-17. LSTM ensemble layers architecture.

The six-station ensemble model significantly enhanced forecasting RMSE over a more typical LSTM model using all six stations. Table 5-3 shows RMSE difference for February 11, 2020, forecasting temperature declines during nighttime hours. Figure 5-18 and 5-19 show information for two different stations (e.g., 213 and 1) on the same forecasting day. Note that the model clearly captured both trends without using one specific model for each station.

Table 5-3
RMSE Difference Between LSTM and Ensemble

Station ID	RMSE Normal LSTM	RMSE Ensemble LSTM
ST213	1.57	1.71
ST199	3.45	1.67
ST235	3.11	1.54
ST51	2.89	1.42
ST32	3.79	1.87
ST1	3.49	1.77

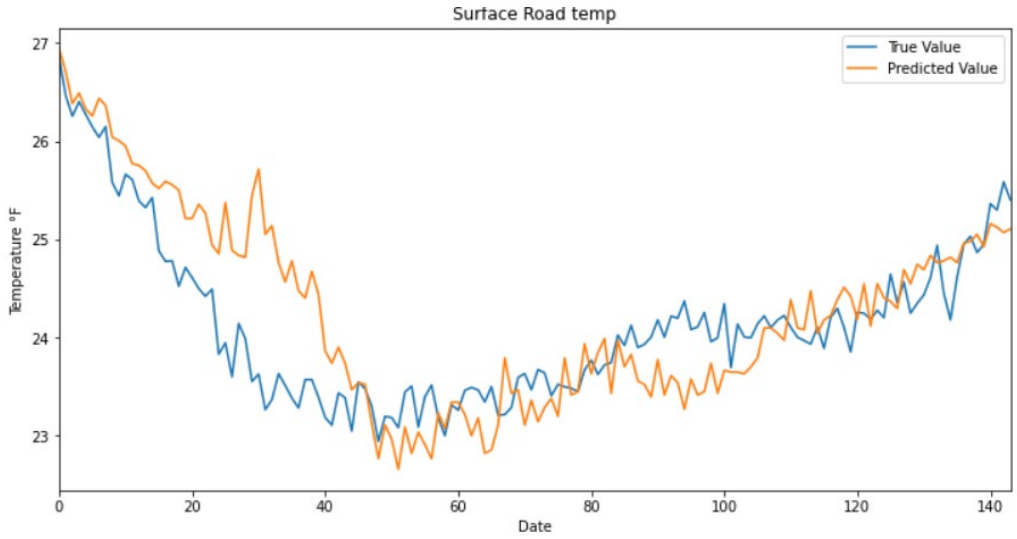


Figure 5-18. ST235 ensemble forecast.

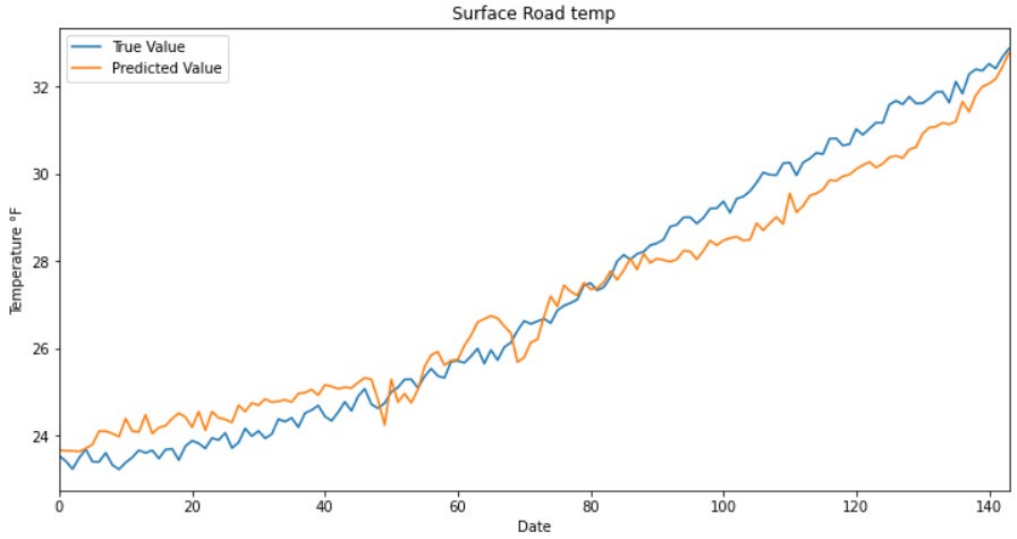


Figure 5-19. ST001 ensemble forecast.

A subsequent case study was performed on the ensemble's and its ability to predict additional hours based on the amount of historical training data (e.g., up to 100 years of progress archived by NWS/NOAA for forecasting models [36]). Figure 5-20 shows in red the correlation between actual and GFS forecast data. As the amount of historical data increases, the model accuracy increases, as well, with the exception of modeling improvement and NWP contribution. To simulate graph similar to a smaller scale model generated by NWS, the model was trained using a 5-week period of data for seven months to generate RMSE for every trained sample. Figure 5-21 summarizes

RMSE performance. Notably the score declined after an October winter event was added to the model. This result suggests that the more winter events are added to the model, the more performance will increase. An ARIMA model was also added to the graph to project future enhanced model performance (i.e., RMSE could decline to nearly 0 for the 12-hours forecast).

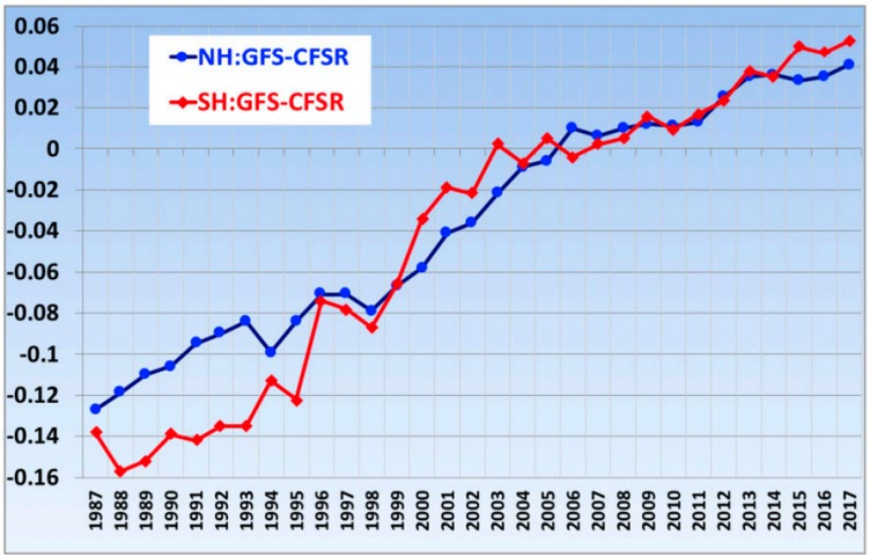


Figure 5-20. 500-hPa anomaly correlation in the Northern Hemisphere for 120-h forecasts by GFS model.

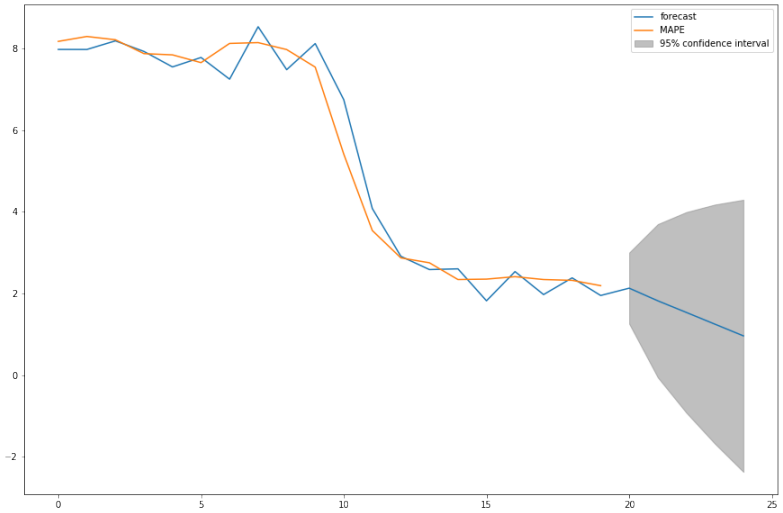


Figure 5-21. ARIMA model forecast of ensemble model overtime.

5.1.4 Regression with GFS Forecasted Model

This section uses the method detailed in Section 4.3 (e.g., parsing future forecasted data from GFS model). Future data from GFS with RWIS current real time data was used to predict the next 24 hours. For model evaluation, RMSE and accuracy with a threshold of ±1.5°F was used to test performance. Linear, polynomial, support vector, decision tree, and random forest were studied. Overall, random forest exhibited the best performance among all models, as shown in Figures 5-22 and 5-23. Random forest achieved 0.9 average RMSE and 95.41% accuracy.

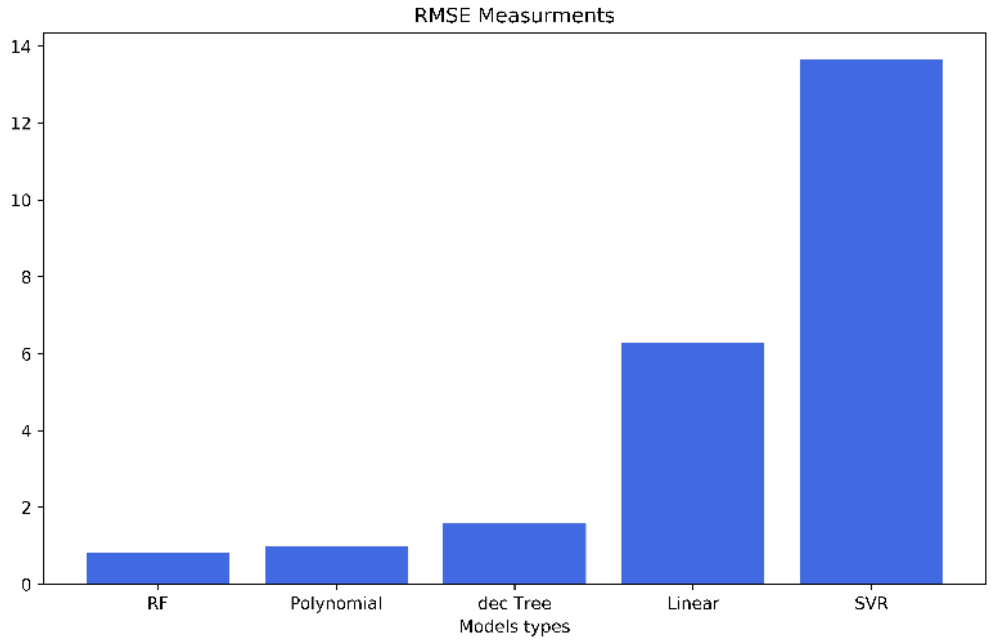


Figure 5-22. RMSE measurements for regression models.

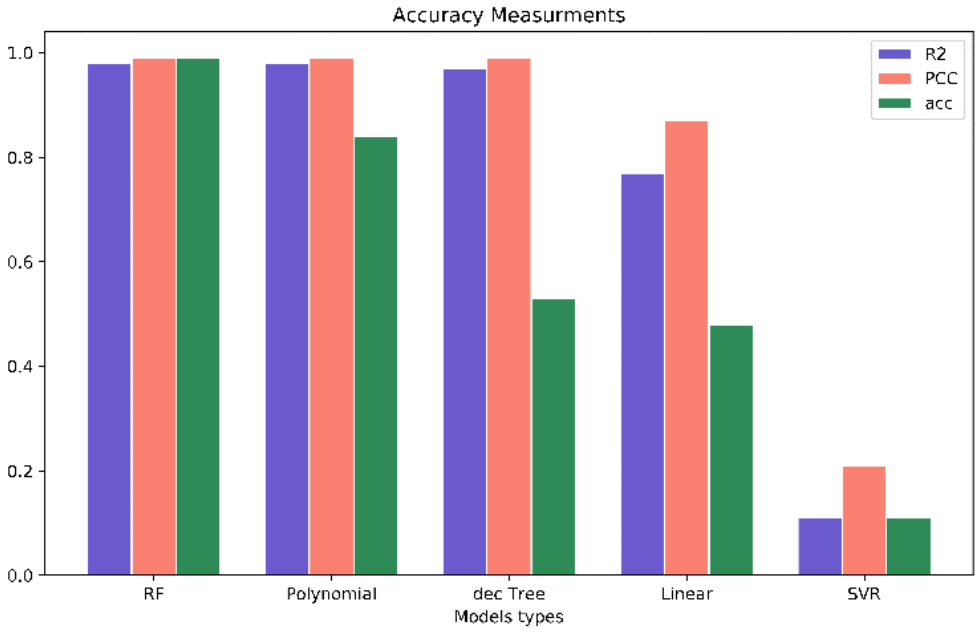


Figure 5-23. Accuracy measurements for regression models.

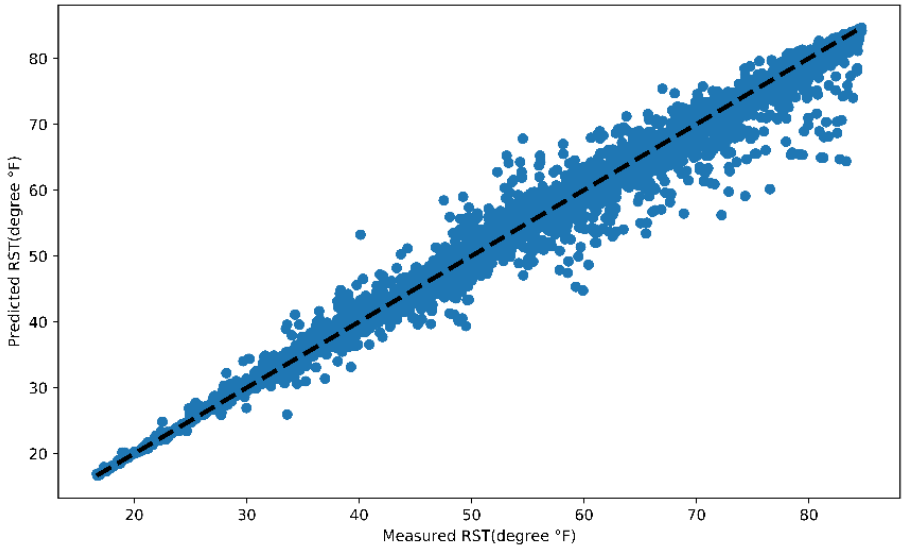
Further analysis of random forest model performance is shown in Table 5-4. RMSE was 1.36 for RST and 1.30 for BST; accuracy was 91% for RST and 85% for BST. Two additional formulas for R-Squared and Pearson correlation coefficient (PCC) were used for further analyses (See Equations 6 and 7). A scatter plot shows RST and BST predicted values versus observed values. Figure 5-24 indicates a strong correlation as points become proximate to the dashed horizontal trend line.

$$R^2 = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2} \quad (6)$$

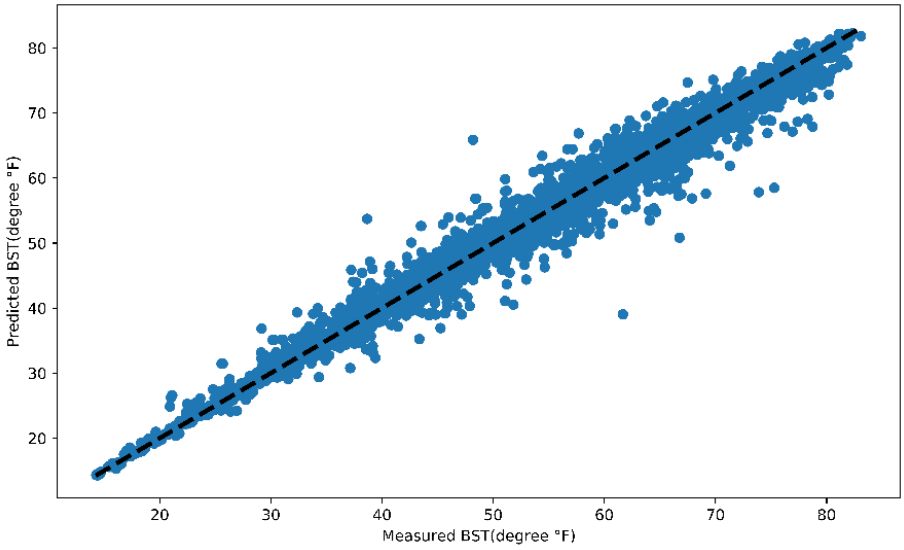
$$pcc = \frac{\sum(x_i-\bar{x})(y-\bar{y})}{(N-1)S_X S_Y} \quad (7)$$

Table 5-4
Random Forest Regression Model Results

Accuracy	RST	BST
RMSE	1.36	1.30
MAE	0.80	0.79
R-Squared	0.99	0.99
PCC	0.995	0.995
Accuracy	91%	86%



(a)



(b)

Figure 5-24. (a) RST; (b) BST RF scatterplots of predicted versus observed values.

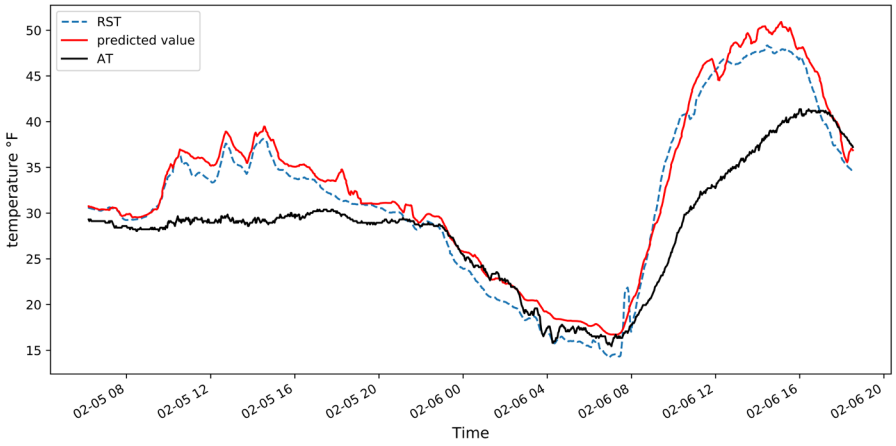
Forecast aims to predict sharp edge declines and temperatures rising above the freezing level. No correlation exists between surface temperature and ambient temperature. Random forest performance was tested on two wintry events on February 6 and February 21, 2020. Tables 5-5 and 5-6 provide model evaluation for the two wintry events. Figure 5-25 shows a rise and fall in BST and RST predictions, as well as model prediction accuracy.

Table 5-5
6 February 2020 RF Results

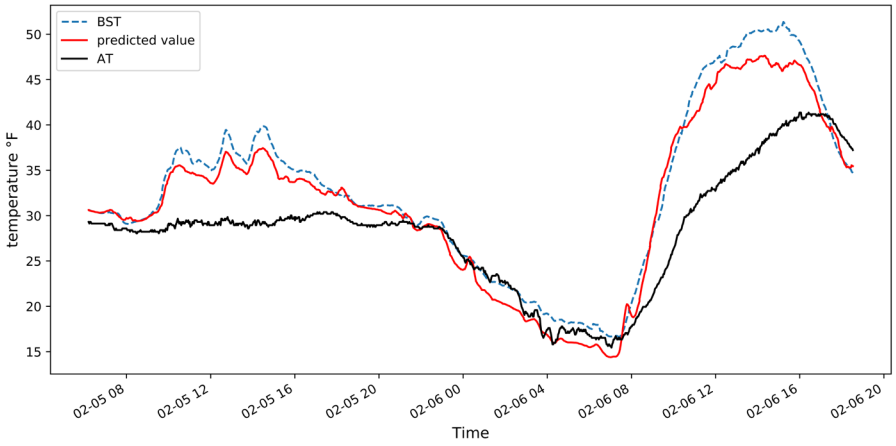
Accuracy	RST	BST
RMSE	2.10	2.21
MAE	1.29	1.30
R-Squared	0.95	0.95
PCC	0.995	0.995
Accuracy	91%	88%

Table 5-6
21 February 2020 RF Results

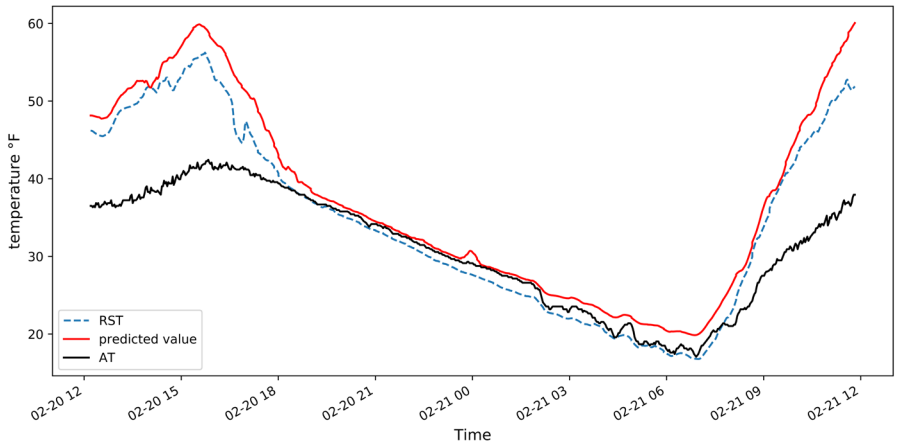
Accuracy	RST	BST
RMSE	3.75	3.34
MAE	1.69	1.67
R-Squared	0.90	0.93
PCC	0.994	0.994
Accuracy	94%	84%



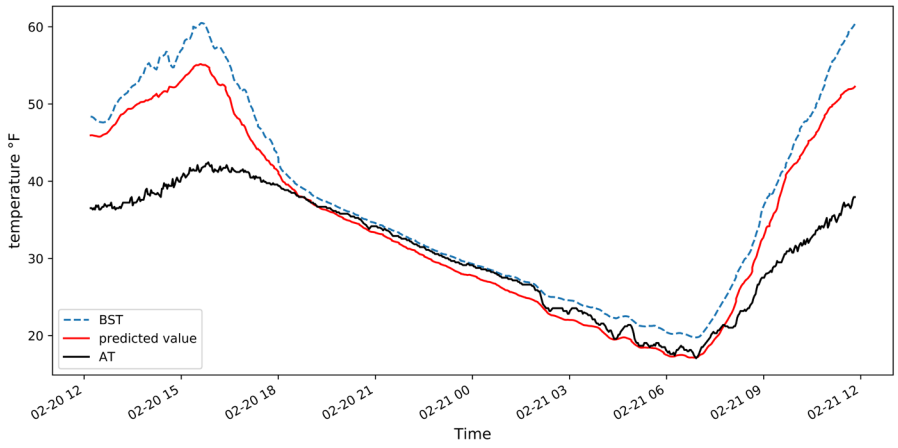
(a)



(b)



(c)



(d)

Figure 5-25. Predicted, actual, and ambient temperature for (a) February 6 RST; (b) February 6 BST; (c) February 21 RST; (d) February 21 BST.

5.2 Classification Models

This section describes data analysis described in Section 4.5.2 for road surface condition forecasting classification models. As previously mentioned, the model will only predict two road classes—safe and dangerous. The dataset was preprocessed (e.g., rescaling, transformation, and percentage change difference) as detailed in Section 4.5.2. Notably the classification model must have a historical data trend to predict one class for each attribute window. This means that the dataset must be reshaped in separate window sizes with a class target for each window. Selected

window size is 72, providing six hours of data. Data shape input into the model will take on a shape of [X, 72, Y], where X represents attributes and Y represents target class.

5.2.1 Supervised Classification Machine Learning Models

Classification model is a type of supervised learning wherein data fed to the network is labeled with the features previously separated into different categories. Hence, the network knows which input parts are important and if a specific target of the model can be verified. Results for each trained model is available below, and optimal performance is summarized using score metrics defined in terms of true / false positives and true / false negatives. Metrics employed are precision, which is defined as ratio of true positives to the sum of true and false positives and recall, which measure classifier completeness (i.e., the ability of the classifier to correctly find all positive instances). The F1 score is the harmonic center between precision and recall. The harmonic center optimizes balance between precision and recall. Model testing reported in this section predicts only the next 3 hours due to lack of memory learning, as well as limited window size and one-dimension target class value. Since machine learning classification models take only two-dimensional input, data in one row was repeated in a sequence of t-n to t. The 'n' value selected for all trained models was 5, and class value was shifted by 41. These provided three and half hours of data projection. Figure 5-26 shows the dataset shifting and reshaping procedure.

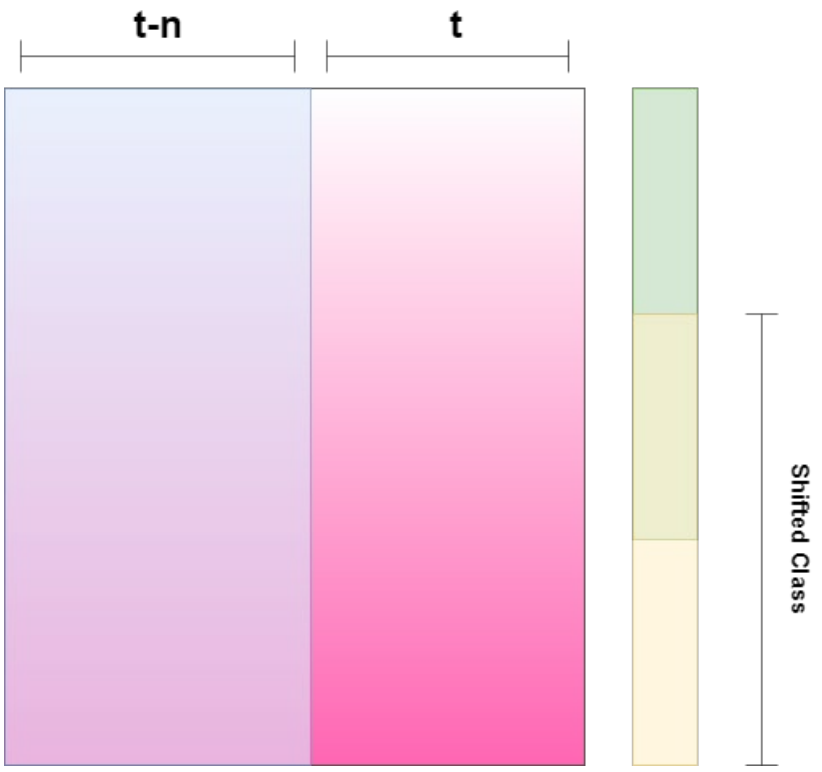


Figure 5-26. Dataset reshape and shifting.

Four models showed acceptable results. The logistic regression model was superior, with a pression of 0.84 for dangerous class and 0.93 for safe class (See Figure 5-28). Dangerous class had an acceptable F1 score of 0.54 and 0.96 for safe class. The threshold plot in Figure 5-27 visualizes all metric scores with respect to binary classifier discrimination threshold (i.e., probability at which positive class is chosen over negative class attribute). Figure 5-29 details the confusion matrix of the model.

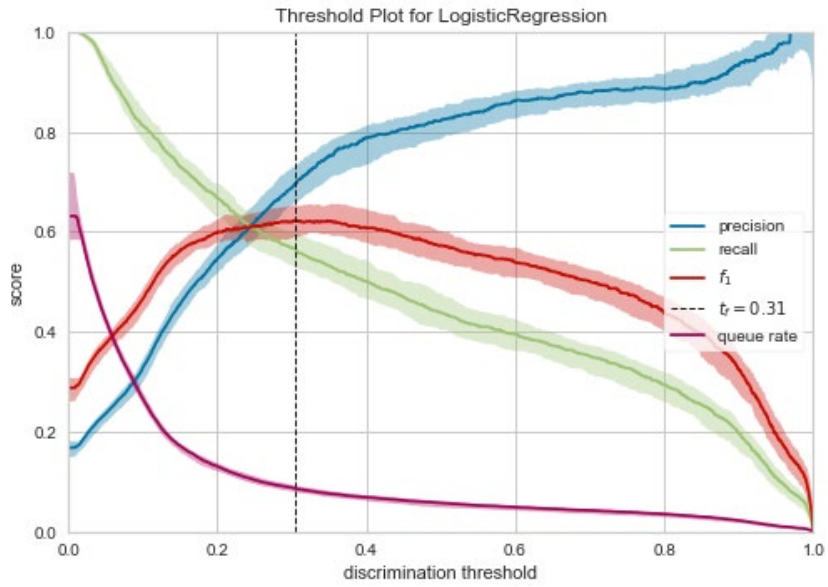


Figure 5-27. Logistic regression model discrimination threshold.

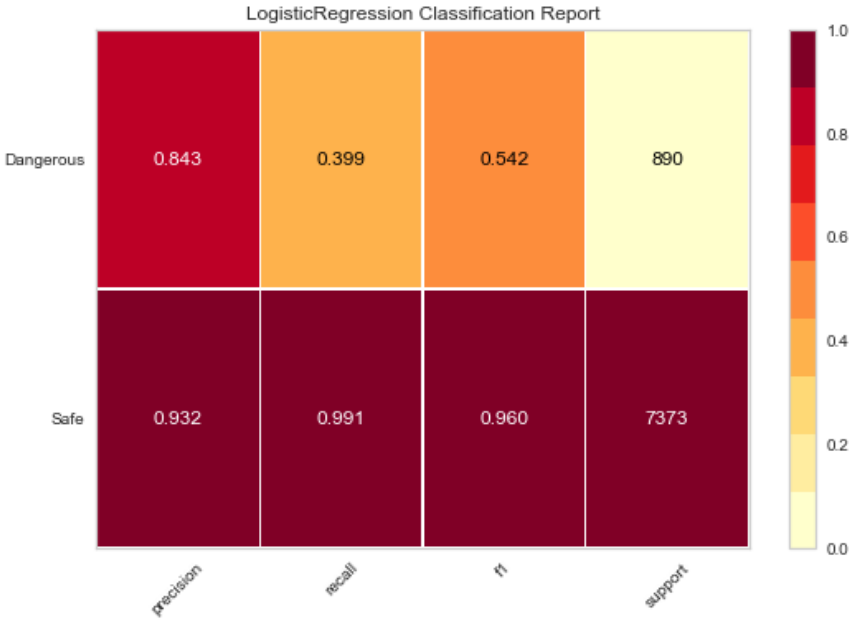


Figure 5-28. Logistic regression model metric scores.

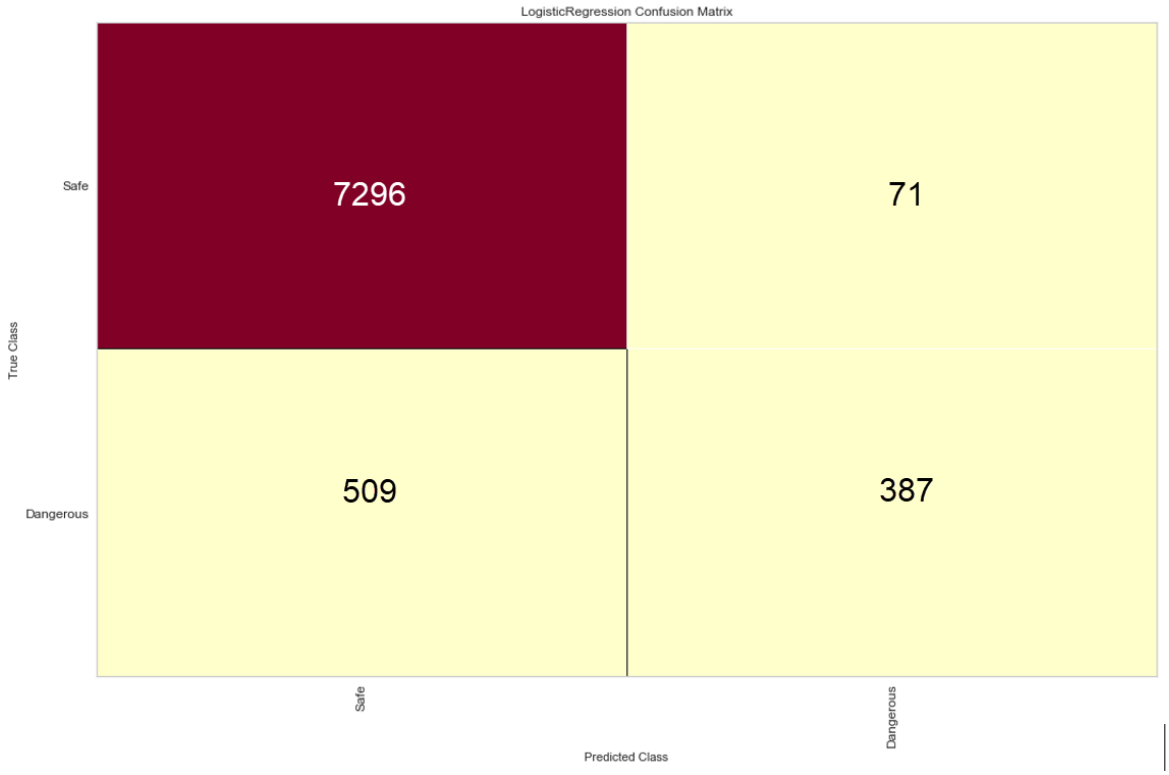


Figure 5-29. Logistic model confusion matrix.

KNN classifier at two neighboring classifiers delivered the next best results. Precision metric had high score for safe class and an acceptable score dangerous class. Figure 5-30 reveals that the precession score increased at 0.7 threshold. The F1 and recall score was also acceptable, measuring above 0.5. Results for the confusion matrix are shown in Figure 5-32 and demonstrate low false positive and false negative.

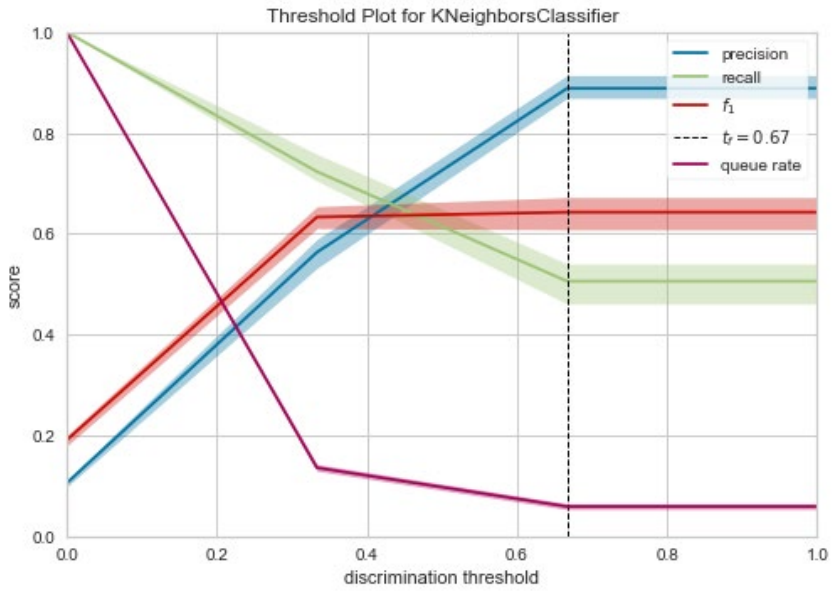


Figure 5-30. KNN model discrimination threshold.

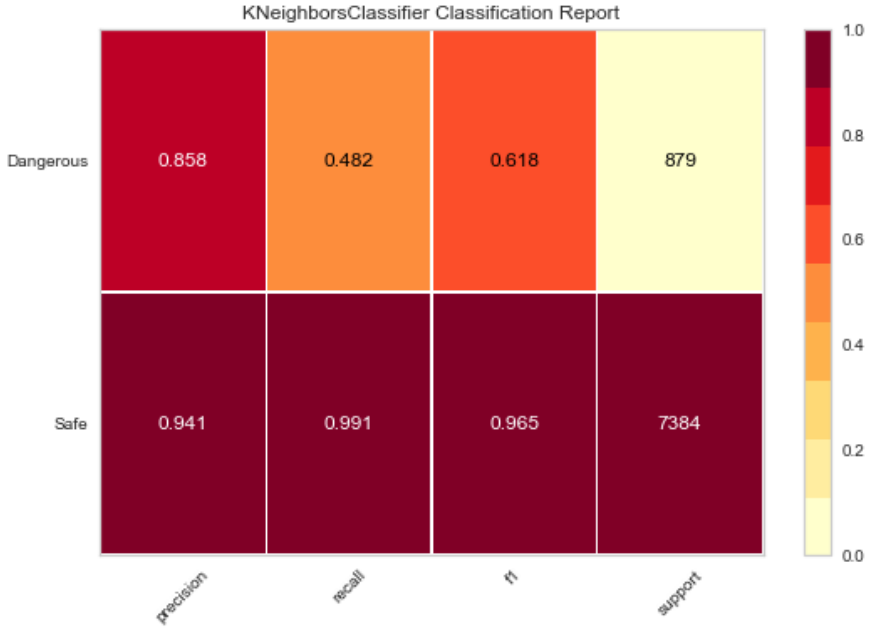


Figure 5-31. KNN Model metric scores.

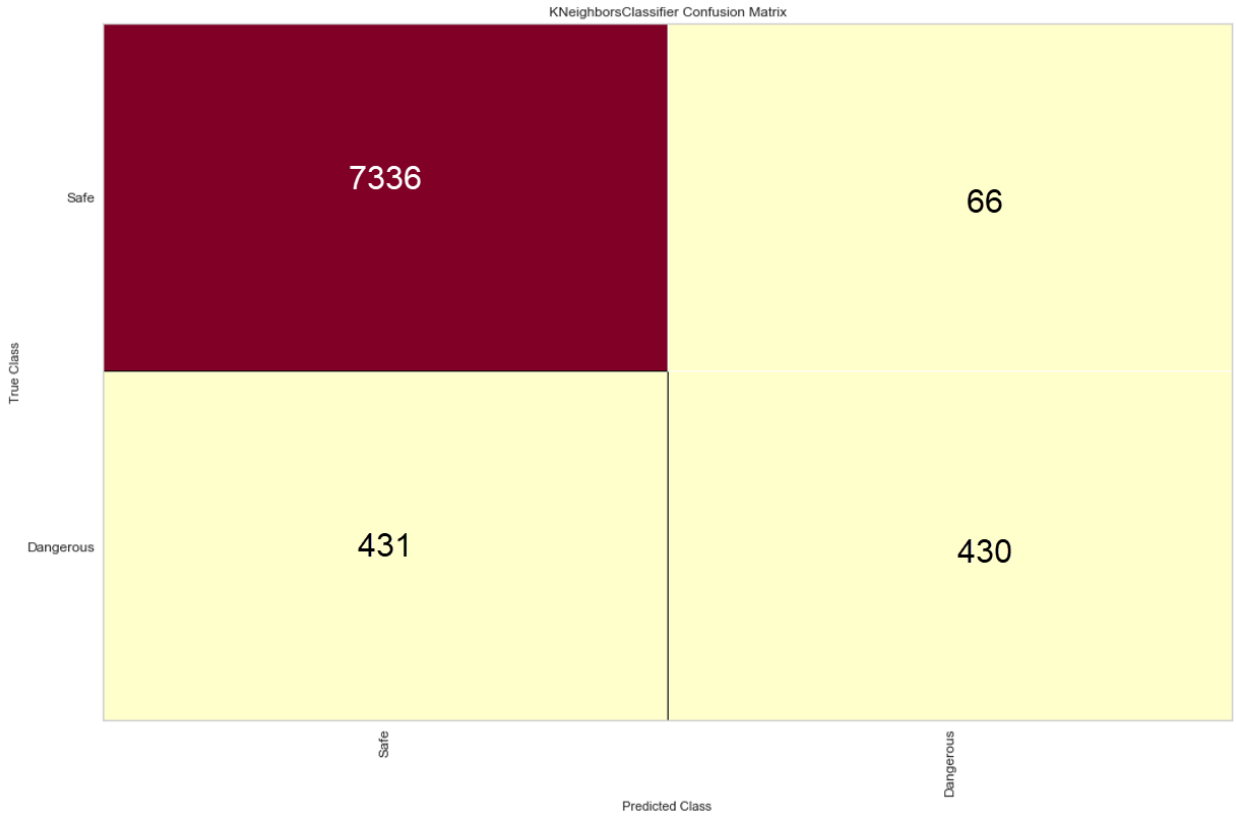


Figure 5-32 KNN Model confusion matrix.

Gaussian naïve bayes (NB) ranked next and did not exhibit promising metric scores (See Figure 5-33). Precision score for the dangerous class was 0.423, and the confusion matrix exhibited a false positive of 555 (See Figure 5-34).

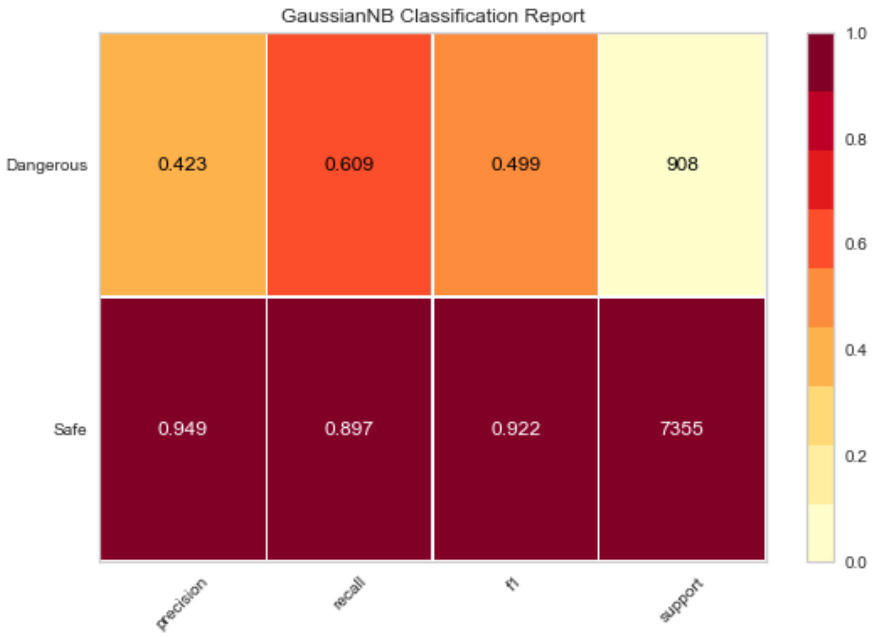


Figure 5-33. Gaussian NB model metric scores.

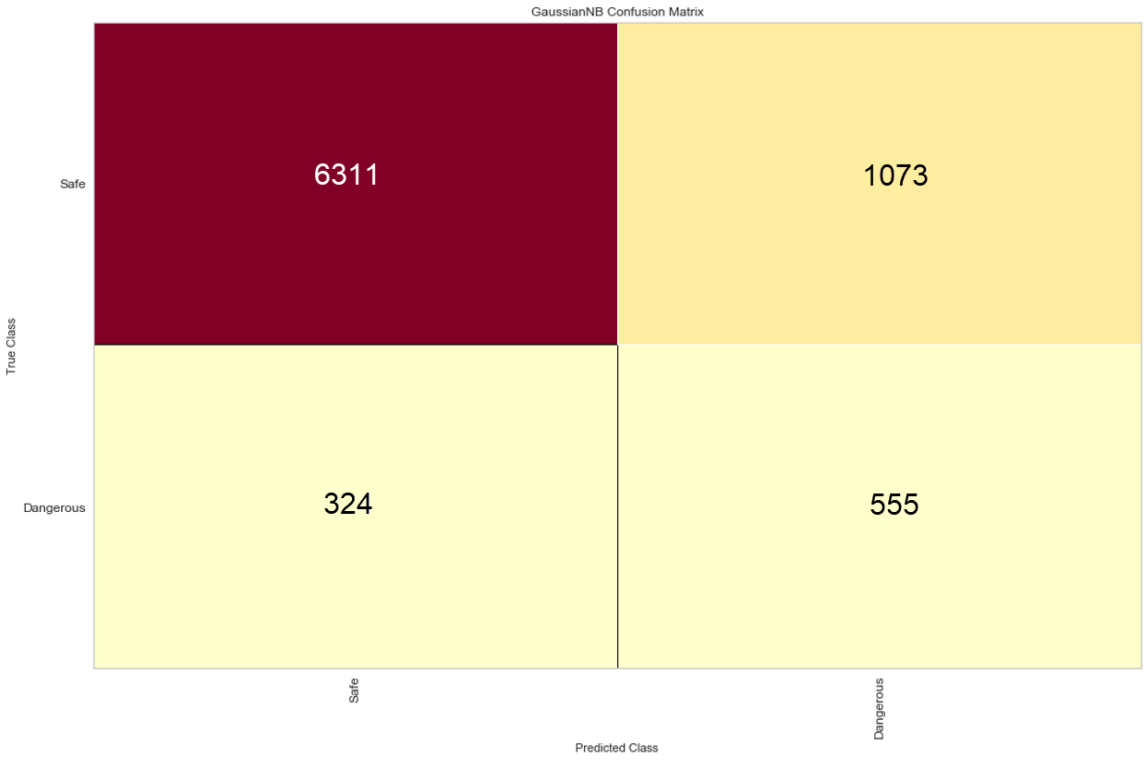


Figure 5-34. Gaussian NB model confusion matrix.

The final model under review was the gradient boosting classifier. Not only did it score the best performance for positive class indicating hazardous road conditions, but it also demonstrated a promising score for F1 and recall. Figure 5-35 shows an acceptable F1 score above 0.5.

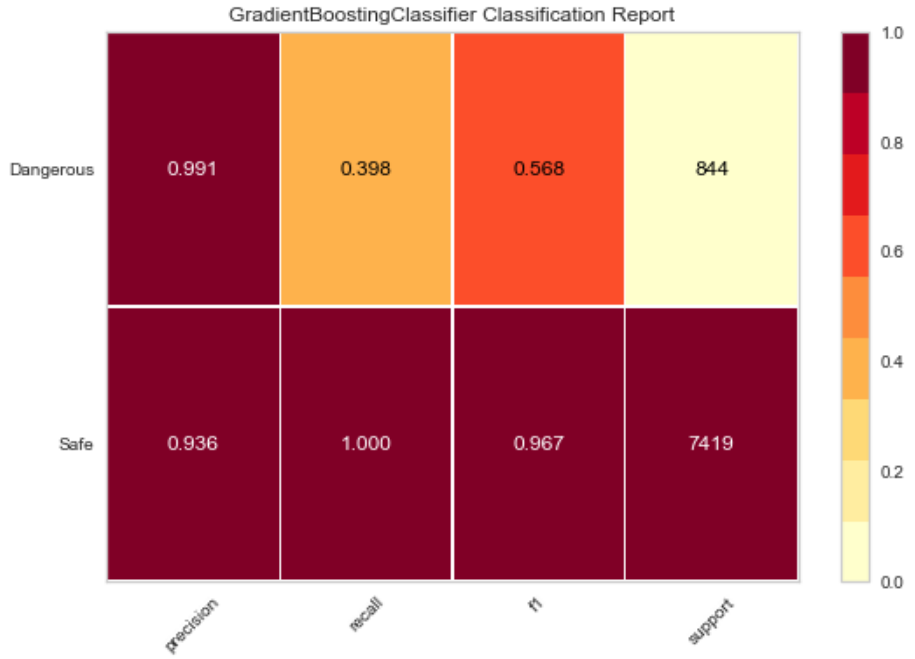


Figure 5-35. Gradient boosting classifier metric scores.

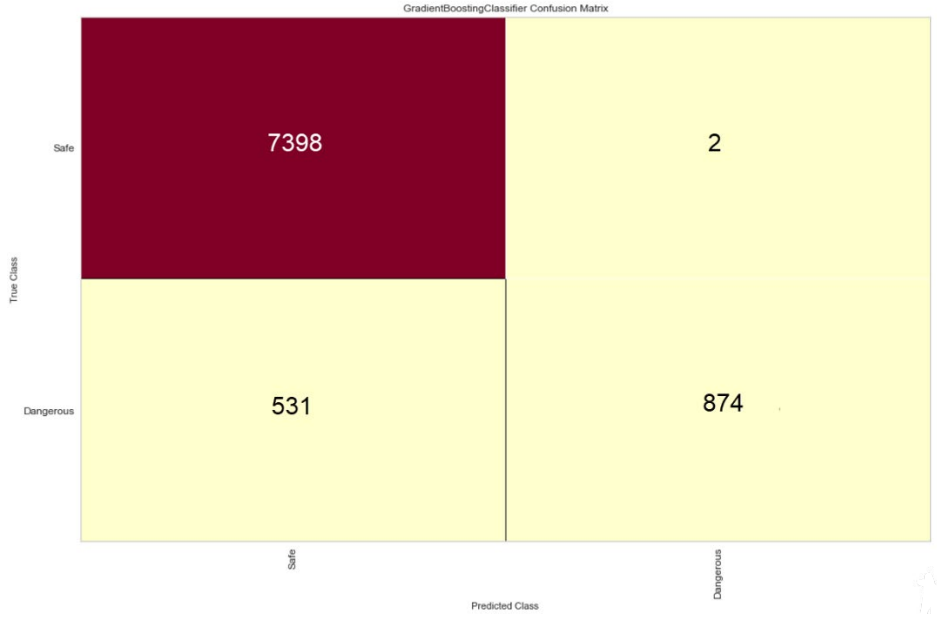


Figure 5-36. Gradient boosting classifier confusion matrix.

5.2.2 LSTM Classification Model

Section 5.1.2 highlighted information about the LSTM model mechanism and its ability to learn input data sequences. Because this research trained on a time series dataset, LSTM proves to be the best model for this process. Section 5.2 also described the way in which a sliding window of T was used on the dataset for reshaping into three dimensions—X, T, Y. In this way, the model can predict a bunch of classes every sequence. Discussion in the previous section emphasized that reshaping works only in one dimension and can predict one class each 3.5 hours. The model architecture is similar to that previously presented in the regression section, although three batch normalization layers were added to the model to account for the high dimensionality of the input. The model must be stabilized in the learning process to reduce the number of training epochs in the process.

Table 5-7
LSTM Model Architecture

Layer Type	Output Shape	Param	Activation Function
LSTM	(None,120, 8)	800	Hard Sigmoid
Batch Normalization	(None, 120,8)	32	-
LSTM 1	(None, 120,8)	544	Hard Sigmoid
Batch Normalization	(None, 120,8)	32	-
LSTM 2	(None, 120, 8)	544	Hard Sigmoid
Batch Normalization	(None, 120,8)	32	-
Dropout	(0.1)	0	-
Dense 3	(None, 1)	9	Sigmoid

Figure 5-37 shows LSTM model loss curve learning and demonstrates that the model reached accuracy within the first 15 epochs. Accuracy was 87.98% and was calculated based on a number of test sets, which were calculated each time the window rolled through the test set. The confusion matrix in Figure 5-38 shows one of the samples. Accuracy tends to be high in many test sets. Additional classes were tested to verify model accuracy for detecting slick and snowy conditions. Results proved good accuracy at 78% (See Figure 5-39).

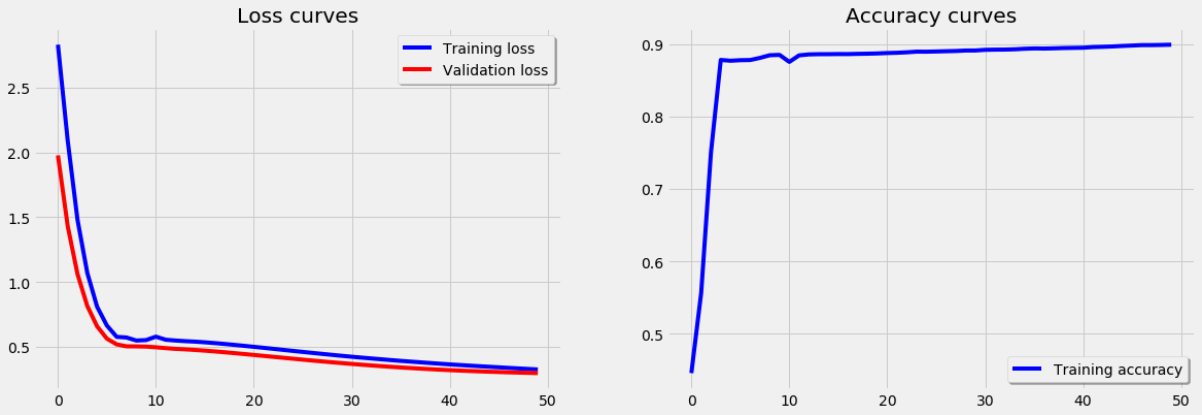


Figure 5-37. LSTM classification model loss and accuracy curve.

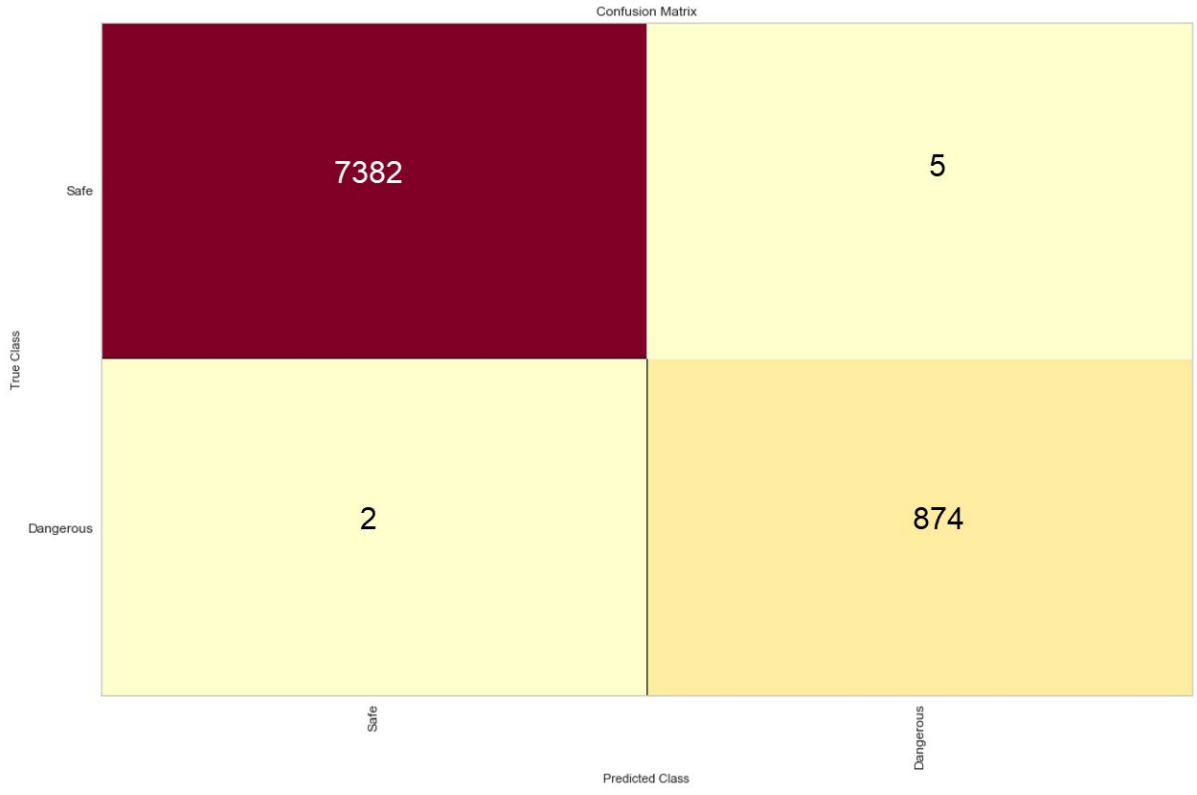


Figure 5-38. LSTM model confusion matrix.

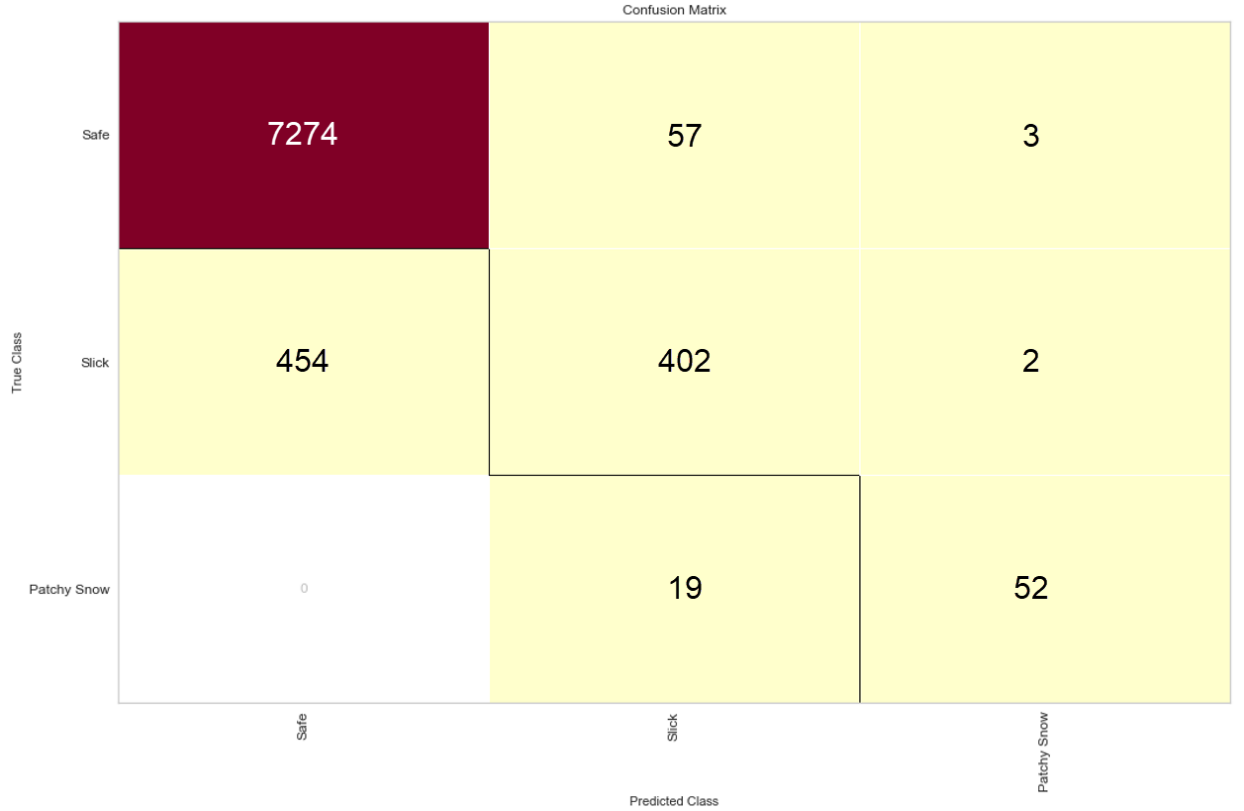


Figure 5-39. LSTM model three class confusion matrix.

5.2.3 Regression and Classification Model using GFS forecasted Model

A bridge model is described in this section. Road surface temperature data is forecasted using the ensemble LSTM model discussed in Section 5.1.3. This model merges forecast precipitation from the GFS model discussed in Section 4.3 and will trigger weather advisories based on freezing level threshold and precipitation type event. Based on the findings discussed in section 4.2, The model will trigger a dangerous condition in case there is a wintry precipitation as Sleet, Freezing rain and Ice Pellets. Another dangerous event will be triggered if the temperature crossed the freezing level twice with a wintry precipitation of snow, otherwise it will trigger a Safe event.

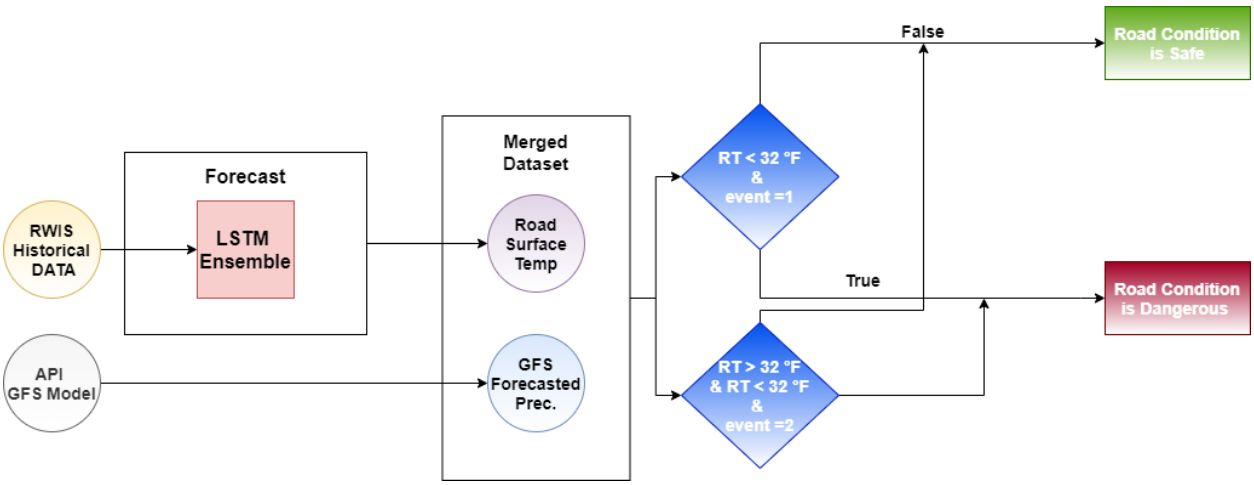


Figure 5-40. LSTM and GFS classification model.

Although the model was not tested in real time based on forecasted GFS data, it was tested on recorded historical data and past events that satisfied the conditions. Two tested events were selected from among October and December 2020 winter events characterized by a mix of wintery precipitation with temperatures fluctuating above and below the freezing level. Figure 5-41 and 5-42 of the two testing sets proved 100% accurate at detecting road condition class. Notably, the GFS model provides 100% accuracy at predicting time, type, and amount of precipitation in the upcoming 12-hour period. The ensemble LSTM provided accurate detection of temperature fluctuation (i.e., 1.67 RMSE value), which aided the model in accurately detecting road class..

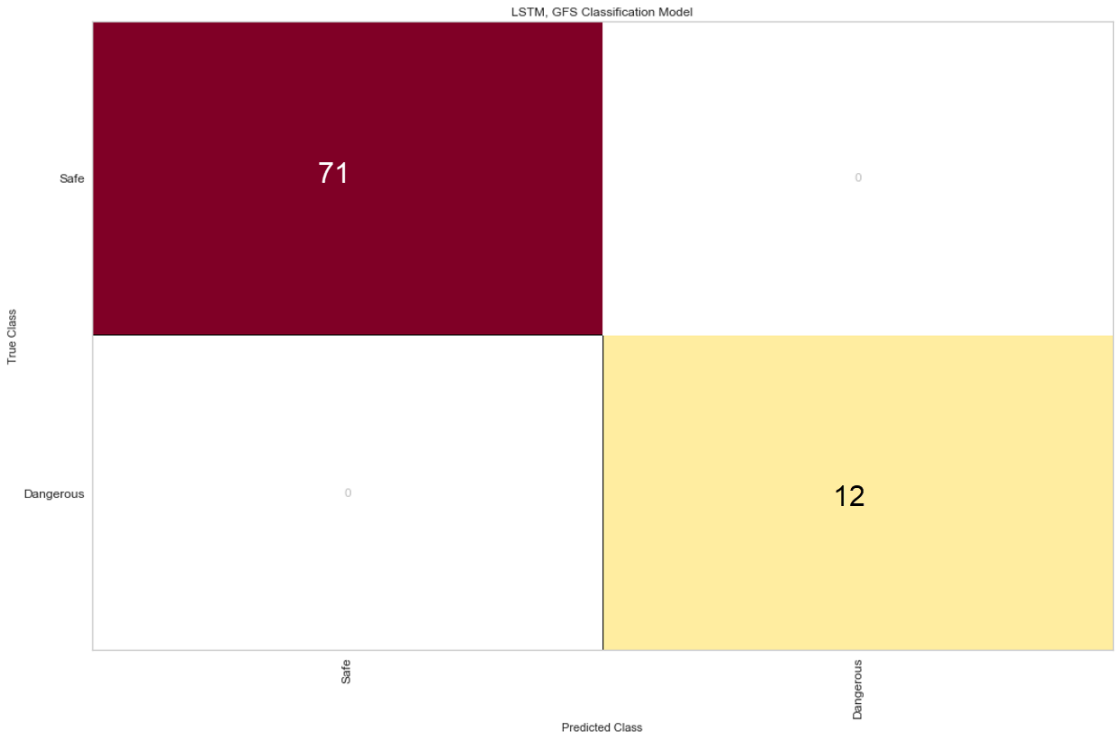


Figure 5-41. October 27, 2020, confusion matrix using LSTM and GFS models.

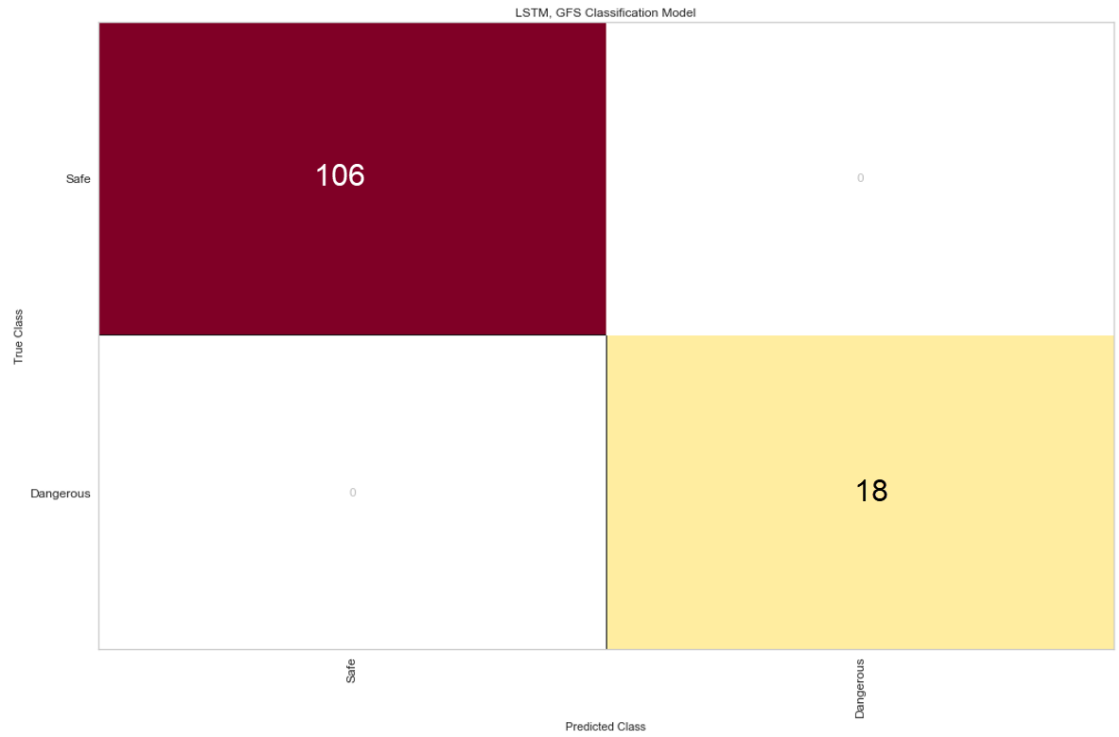


Figure 5-42. December 1, 2020, confusion matrix using LSTM and GFS models.

6 Conclusion and Future Work

The work presented in this thesis details a variety of analyses aimed at predicting road surface temperatures and conditions using model testing, training, and validation. The optimal approach for acquiring data from an RWIS server utilized restful API with time series model forecasting and LSTM neural network models. Research proved that the ensemble LSTM model was the best performing regression model, delivering an average RMSE of ±1.67 and 87.98% accuracy for detecting hazardous road conditions. Most referenced papers and meteorology agencies have previously reported near or lower scores than those reported in this thesis. Results demonstrated that the RWIS system offered reliable data and the possibility of implementing a real-time model for road conditions analyses and prediction.

The RWIS system showed significant potential for analyzing road conditions and traffic when leveraging traffic flow data and weather conditions in real time and providing analytical data and modeling methods implemented in the server.

RWIS historical data is able to record more complicated events with hazardous road conditions and contribute to the modeling scheme for enhancing metric scores utilized in this thesis. The bigger the training data size, the more accurate and reliable the model, as evidenced in Section 5.1.3. ARIMA model projects demonstrated that the RWIS model can accurately forecast temperatures for an upcoming 12 hours at nearly 0 RMSE.

ODOT and OU-Tulsa's WECAD are working together to deploy more roadway systems for analyzing traffic and road conditions using smart technologies. Models developed by researchers can enhance the learning processes and help guide future research teams toward solving future obstacles (e.g., accidents and road congestion). This thesis showed the importance of an LSTM

and time series model approach for capturing future weather events., These models can be utilized for solving many other traffic issues.

References

[1] "How Do Weather Events Impact Roads?" 2020. *How Do Weather Events Impact Roads? - FHWA Road Weather Management*. USDOT. https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm

[2] Michaels, M. (n.d.). February deep FREEZE Brings FIRST 2021 billion Dollar disaster. Retrieved April 19, 2021, from https://www.weathernationtv.com/news/february-deep-freeze-brings-first-2021-billion-dollar-disaster/?fbclid=IwAR1PRT6328LEX_pjAwFp6tIEuzHthqXJRz7UnGI32PeGiOMTDlXARPVNdTc

[3] "Why Do Bridges Ice before the Road?" 2011. The Weather Guys. Cooperative Institute for Meteorological Satellite Studies. https://wxguys.ssec.wisc.edu/2011/02/28/why-do-bridges-ice-before-the-road/.

[4] Mass, C. (n.d.). Roadway icing and Weather: A tutorial. Retrieved April 30, 2021, from https://www.atmos.washington.edu/~cliff/Roadway3.html

[5] Jacobs, Wilfried, and Wolfgang E Raatz. 2007. "Forecasting Road-Surface Temperatures for Different Site Characteristics." Meteorological Applications 3 (3): 243–256. doi:10.1002/met.5060030306.

[6] "Independent Gritting." 2020. Met Office. UK gov. Accessed May 20. https://www.metoffice.gov.uk/services/transport/road/independent-gritting.

[7] Donald M. Moore, NOAA/NWS, Billings, MT; and R. J. Miller. 2007. "Road pavement temperatures and their impact on travel during snow storms". American Meterological Society.

[8] Kangas, Markku, Martti Heikinheimo, and Marjo Hippi. 2015. "RoadSurf: a Modelling System for Predicting Road Weather and Road Surface Conditions." *Meteorological Applications* 22 (3): 544–553. doi:10.1002/met.1486.

[9] Yang, Choong Heon, Duk-Geun Yun, and Jung Gon Sung. 2012. "Validation of a Road Surface Temperature Prediction Model Using Real-Time Weather Forecasts." *KSCE Journal of Civil Engineering* 16 (7): 1289–1294. doi:10.1007/s12205-012-1649-7.

[10] Belles, Jonathan. 2018. "21-Degree Drop in 4 Minutes: The Biggest Temperature Swings from Winter Storm Oliver." *The Weather Channel*. The Weather Channel. https://weather.com/news/weather/news/2018-02-21-big-temperature-swings-winter-storm-oliver-cold-front.

[11] Miscellaneous devices. (n.d.). Retrieved May 06, 2021, from https://www.thiesclima.com/en/Products/Miscellaneous-Devices/

[12] Surface sentinel temperature sensor - model 5439-00. (2020, December 23). Retrieved May 06, 2021, from https://hsierra.com/product/surface-temperature-sensor-model-5439-00/

[13] 109 - temperature probe. (n.d.). Retrieved May 06, 2021, from https://www.campbellsci.com/109#:~:text=The%20109%20is%20a%20rugged,or%20submerge%20it%20in%20water.

[14] REECE capabilities. (n.d.). Retrieved May 06, 2021, from http://itrafficsystems.com/index.html

[15] AirLink RV50X: Industrial LTE-Advanced Gateway: Sierra Wireless. (n.d.). Retrieved May 06, 2021, from https://www.sierrawireless.com/products-and-solutions/routers-gateways/rv50/

[16] Dashboard. (n.d.). Retrieved May 06, 2021, from https://rwis.tulsa.ou.edu/rwis/

[17] Common errors in infrared temperature measurement. (2019, September 19). Retrieved April 28, 2021, from https://www.bestech.com.au/blogs/temperature/common-errors-in-infrared-temperature-measurement/

[18] Convolution kernels. (n.d.). Retrieved April 28, 2021, from https://het.as.utexas.edu/HET/Software/Astropy-1.0/convolution/kernels.html

[19] Miller, B. (n.d.). Types of winter precipitation. Retrieved April 30, 2021, from https://weatherworksinc.com/news/types-of-precipitation-11-19-19#:~:text=During%20the%20winter%2C%20there%20are,sleet%2C%20and%20of%20course%20snow.

[20] Wassenhoven, L. (2004, June 02). THE ATMOSPHERE. Retrieved April 30, 2021, from https://theozonehole.com/atmosphere.htm

[21] Bechtold, P. (2015, May). Atmospheric Thermodynamics. Retrieved from https://www.ecmwf.int/sites/default/files/elibrary/2015/16954-atmospheric-thermodynamics.pdf

[22] Weather fronts. (n.d.). Retrieved April 30, 2021, from https://okfirst.mesonet.org/train/meteorology/Fronts.html

[23] Lederman, J. I. (2017). *Dynamics in the Anvil outflow of Tropical Convection* (Master's thesis, 2017). Toronto, ON: York University.

[24] Skew-T parameters. (n.d.). Retrieved April 30, 2021, from https://www.weather.gov/source/zhu/ZHU_Training_Page/convective_parameters/skewt/skewtinfo.html

[25] US Department of Commerce, N. (2016, September 17). Winter precipitation types. Retrieved April 30, 2021, from https://www.weather.gov/ama/preciptypes

[26] O. (n.d.). Ok.maps.arcgis.com. Retrieved May 03, 2021, from https://ok.maps.arcgis.com/apps/Viewer/index.html?appid=023e821ebf7b4acd999ccfd58d92c3da

[27] Weather in Washington, DC, District of Columbia, United States. (n.d.). Retrieved May 04, 2021, from https://www.tomorrow.io/weather/

[28] Badr, W. (2019, January 22). Why feature correlation matters .... a lot! Retrieved May 10, 2021, from https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4

[29] Zeeble, B. (2021, April 23). NTSB report details Deadly 133-car pileup On I-35 In Fort Worth. Retrieved May 10, 2021, from https://www.keranews.org/news/2021-04-22/ntsb-releases-report-on-deadly-i-35-pile-up-in-fort-worth

[30] *RadViz Visualizer*¶. RadViz Visualizer - Yellowbrick v1.3.post1 documentation. (n.d.). https://www.scikit-yb.org/en/latest/api/features/radviz.html.

[31] *non-Linear Models, Kernel*. scikit. (n.d.). https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification.

[32] Smagulova, Kamilya, Kazybek Adam, Olga Krestinskaya, and Alex Pappachen James. "Design of CMOS-memristor Circuits for LSTM Architecture." (2018): IEEE

[33] International Conferences on Electron Devices and Solid-State Circuits, 2018. Web.

[34] Hochreiter, Sepp, and Schmidhuber, Jürgen. "Long Short-Term Memory." Neural Computation 9.8 (1997): 1735-780. Web.

[35] Olah, C. (n.d.). *Understanding LSTM Networks*. Understanding LSTM Networks -- colah's blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[36] Benjamin, S. G., Brown, J. M., Brunet, G., Lynch, P., Saito, K., & Schlatter, T. W. (2019). 100 Years of Progress in Forecasting and NWP Applications. *Meteorological Monographs*, *59*. https://doi.org/10.1175/amsmonographs-d-18-0020.1

Appendix A: Algorithms and Functions

1. Parse RWIS data using API

```
def Get_RWIS_Data_15min(var2, var3,var4):
  import requests
  import numpy as np
  import pandas as pd
  import json
  from astropy.convolution import convolve, Box1DKernel
  var1 = "https://rwis.tulsa.ou.edu/rwis/api/weatherAll/data?"
  vr = "station="
  var = "&"
  varr = "date_from="
  varrr = "date_to="
  varT = "".join([var1, vr,var2,var,varr,var3,var,varrr,var4])
  Json_data = requests.get(varT).json()
  n= len(Json_data)
  station = []
  date_time = []
  wind_speed = []
  wind_direction = []
  gust_wind_speed = []
  temperature = []
  air_pressure = []
  precipitation_event = []
  precipitation_intensity = []
  total_precipitation = []
  precipitation_type = []
  brightness = []
  temp1_surface_temperature = []
  temp1_dew_point = []
  temp2_surface_temperature = []
  temp_probe_1 = []
  temp_probe_2 = []
  gust_wind_direction = []
  for x in range(n):
    station.append(Json_data[x]['station'])
    date_time.append(Json_data[x]['date_time'])
    wind_speed.append(Json_data[x]['wind_speed'])
    wind_direction.append(Json_data[x]['wind_direction'])
    gust_wind_speed.append(Json_data[x]['gust_wind_speed'])
    gust_wind_direction.append(Json_data[x]['gust_wind_direction'])
    temperature.append(Json_data[x]['temperature'])
```

```
    air_pressure.append(Json_data[x]['air_pressure'])
    precipitation_event.append(Json_data[x]['precipitation_event'])
    precipitation_intensity.append(Json_data[x]['precipitation_intensity'])
    total_precipitation.append(Json_data[x]['total_precipitation'])
    precipitation_type.append(Json_data[x]['precipitation_type'])
    brightness.append(Json_data[x]['brightness'])
    temp1_surface_temperature.append(Json_data[x]['temp1_surface_temperature'])
    temp1_dew_point.append(Json_data[x]['temp1_dew_point'])
    temp2_surface_temperature.append(Json_data[x]['temp2_surface_temperature'])
    temp_probe_1.append(Json_data[x]['temp_probe_1'])
    temp_probe_2.append(Json_data[x]['temp_probe_2'])
  RWIS_Dataframe = pd.DataFrame({'Station':station,'Date':date_time,'Wind
Speed':wind_speed,'Wind Direction':wind_direction,'Gust Wind Speed':gust_wind_speed,
                           'Gust Wind
Direction':gust_wind_direction,'Temperature':temperature,'Air Pressure':air_pressure,
                           'Precipitation event':precipitation_event,'Precipication
Intensity':precipitation_intensity,
                           'Total Precipitation':total_precipitation,'Precipitation
type':precipitation_type,'Brghtness':brightness,
                           'Road Surface Temp':temp1_surface_temperature,'Due Point
Temp':temp1_dew_point,'Bridge Surface Temp':temp2_surface_temperature,
                           '2 inch Underground Temp':temp_probe_1,'6 inch Underground
Temp':temp_probe_2})
  RWIS_Dataframe['Date'] = pd.to_datetime(RWIS_Dataframe['Date'])
  RWIS_Dataframe.set_index('Date', inplace=True)
  df = RWIS_Dataframe.resample('15Min').mean()
  df = df.iloc[::-1]
  df = df.query('`Road Surface Temp` < 85')
  df = df.query('`Road Surface Temp` > -10')
  df = df.query('`Bridge Surface Temp` < 85')
  df = df.query('`Bridge Surface Temp` > -10')
  df = df[df['Road Surface Temp'] != 0]
  df = df[df['Bridge Surface Temp'] != 0]
  smoothed_signal_Train = convolve(df['Road Surface Temp'], Box1DKernel(10))
  smoothed_signal_Train = pd.DataFrame(smoothed_signal_Train)
  smoothed_signal_Train = smoothed_signal_Train.set_index(df.index)
  df['Road Surface Temp'] = smoothed_signal_Train
  smoothed_signal_Train1 = convolve(df['Bridge Surface Temp'], Box1DKernel(10))
  smoothed_signal_Train1 = pd.DataFrame(smoothed_signal_Train1)
  smoothed_signal_Train1 = smoothed_signal_Train1.set_index(df.index)
  df['Bridge Surface Temp'] = smoothed_signal_Train1
  return df
```

2. Parse RWIS data in real time for the last hour

```
import requests
import numpy as np
import pandas as pd
import json
import statistics
from datetime import date, datetime
from pytz import timezone
import os
import geopandas as gpd
import pandas as pd
from bokeh.io import show
from bokeh.models import (CDSView, ColorBar, ColumnDataSource,
                          CustomJS, CustomJSFilter,
                          GeoJSONDataSource, HoverTool,
                          LinearColorMapper, Slider)
from bokeh.layouts import column, row, widgetbox
from bokeh.palettes import brewer, Category20, Turbo256
from bokeh.plotting import figure
from bokeh.embed import file_html
from bokeh.resources import CDN
from bokeh.models import FixedTicker
from bokeh.models import Legend, LegendItem
tz = timezone('US/Central')
tz1 = timezone('US/Mountain')
time = datetime.now(tz)
time1 = datetime.now(tz1)
t = time.strftime("%Y-%m-%d")
t1 = time.strftime("%H:%M:%S")
t2 = time1.strftime("%Y-%m-%d")
t3 = time1.strftime("%H:%M:%S")
comma = ' '
S1 = 'https://rwis.tulsa.ou.edu/rwis/api/weatherAll/data?station='
P1= '&datetime_from='
P2 = '&datetime_to='
st =
['35ST213','35ST199','35ST187','35ST165','35ST154','35ST141','35ST235','35ST107','35S
T092','35ST074','35ST058','35ST051','35ST032',
      '35ST015','35ST001']
st_n = ['213 - Road','213 - Bridge','199 - Road','199 - Bridge','187 - Road','187 -
Bridge','165 - Road','165 - Bridge','154 - Road','154 - Bridge'
        ,'141 - Road','141 - Bridge','235 - Road','235 - Bridge','107 - Road','107 -
Bridge','92 - Road','92 - Bridge',
```

```
        '74 - Road','74 - Bridge','58 - Road','58 - Bridge','51 - Road','51 -
Bridge','32 - Road','32 - Bridge',
        '15 - Road','15 - Bridge','1 - Road','1 - Bridge']
temp1 = []
for x in range(len(st)):
    #print(x)
    temp1_surface_temperature = []
    time_now = []
    temp2_surface_temperature = []
    total_msg = ("".join([S1,st[x],P1,t2,comma,t3,P2,t,comma,t1]))
    data = requests.get(total_msg).json()
    if not data:
        temp = temp1[-1]
        tempp = temp1[-2]
        temp1.append(tempp)
        temp1.append(temp)
    else:
        time_now.append(data[0]['date_time'])
        n= len(data)
        for i in range(n):
            temp1_surface_temperature.append(data[i]['total_precipitation'])
            Not_none_values = filter(None.__ne__, temp1_surface_temperature)
            temp1_surface_temperature = list(Not_none_values)
        if not temp1_surface_temperature:
            temp = temp1[-1]
            temp1.append(temp)
        else:
            temp1.append(statistics.mean(temp1_surface_temperature))

        for i in range(n):
            temp2_surface_temperature.append(data[i]['total_precipitation'])
            Not_none_values = filter(None.__ne__, temp2_surface_temperature)
            temp2_surface_temperature = list(Not_none_values)
        if not temp2_surface_temperature:
            temp = temp1[-1]
            temp1.append(temp)
        else:
            temp1.append(statistics.mean(temp2_surface_temperature))
        del data
        del temp1_surface_temperature
        del temp2_surface_temperature
        time_st = time_now*30
df = pd.DataFrame(
    {'Time': time_st,
     'codes': st_n,
```

```
     'speed': temp1
    })
os.chdir(r"C:\Users\Afify\Documents\ArcGIS")
fp = 'stationsI35_Merge2.shp'
map_df = gpd.read_file(fp)
merged = map_df.set_index('Name').join(df.set_index('codes'))
merged.reset_index(level=0, inplace=True)
merged1 = merged.loc[~merged['NAME_0'].isin(['United States'])]
geosource = GeoJSONDataSource(geojson = merged.to_json())
geosource1 = GeoJSONDataSource(geojson = merged1.to_json())
```

3. Parse specific parameters in real-time

```
def RWIS_data_Now(argument):
    switcher = {
        'W'  : Wind(),
        'T'  : Temperature(),
        'P'  : Pressure(),
        'Pr' : Precipitation(),
        'U_2': Underground_2(),
        'U_6': Underground_6(),
        'G'  : Gust(),
        'S'  : Surface()
    }
    return switcher.get(argument, "Invalid input, please choose a correct one from
the list")
if __name__ == "__main__":
    argument=0
    print (numbers_to_strings(argument))
```

4. Parse Underground 6-inch temperature Function

```
def Underground_6():
    import requests
    import numpy as np
    import pandas as pd
    import json
    import statistics
    from datetime import date, datetime
    from pytz import timezone
    tz = timezone('US/Central')
    tz1 = timezone('US/Mountain')
```

```
    time = datetime.now(tz)
    time1 = datetime.now(tz1)
    t = time.strftime("%Y-%m-%d")
    t1 = time.strftime("%H:%M:%S")
    t2 = time1.strftime("%Y-%m-%d")
    t3 = time1.strftime("%H:%M:%S")
    comma = ' '
    S1 = 'https://rwis.tulsa.ou.edu/rwis/api/weatherAll/data?station='
    P1= '&datetime_from='
    P2 = '&datetime_to='
    st =
['35ST213','35ST199','35ST187','35ST165','35ST154','35ST141','35ST235','35ST107','35S
T092','35ST074','35ST058','35ST051','35ST032',
      '35ST015','35ST001']
    temp1 = []
    for x in range(len(st)):
        temp_probe_2 = []
        time_now = []
        total_msg = ("".join([S1,st[x],P1,t2,comma,t3,P2,t,comma,t1]))
        data = requests.get(total_msg).json()
        if not data:
            temp = temp1[-1]
            temp1.append(temp)
        else:
            time_now.append(data[0]['date_time'])
            n= len(data)
            for i in range(n):
                temp_probe_2.append(data[i]['temp_probe_2'])
                Not_none_values = filter(None.__ne__, temp_probe_2)
                temp_probe_2 = list(Not_none_values)
            if not gust_wind_speed:
                temp = temp1[-1]
                temp1.append(temp)
            else:
                temp1.append(statistics.mean(temp_probe_2))
            del data
            del temp_probe_2
            time_st = time_now*15
    gust_df = pd.DataFrame(
        {'Time': time_st,
         'ST': st,
         'Probe 2': temp1
        })
    return UG2_df
```

5. Parse GFS Dataset for 24 hours

```
headers = {'accept': 'application/json','apikey': 'TfvwMAIRVIF7NkNuB7RAEJIEJA437sdM'}
    response1 =
requests.get('https://api.climacell.co/v3/weather/forecast/hourly?lat=35&lon=-
97&unit_system=us&fields=precipitation&start_time=now', headers=headers)
    RR1 = response1.content
    RR1 = json.loads(RR1)
    m= len(RR1)
    date = []
    Temp = []
    for x in range(m):
        date.append(RR1[x]['observation_time']['value'])
        Temp.append(RR1[x]['precipitation']['value'])
```

6. Convert Time series data to Supervised learning dataset

```
def shift_data(dataa, n_in=1, n_out=1, dropnan=True):
    from pandas import DataFrame
    from pandas import concat
    n_vars = 1 if type(dataa) is list else dataa.shape[1]
    df = DataFrame(dataa)
    cols, names = list(), list()
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
    agg = concat(cols, axis=1)
    agg.columns = names
    if dropnan:
        agg.dropna(inplace=True)
    return agg
```