

Complex Meteorology over Complex Terrains: Assessment of  
Topography, Land Use, Grid Resolution, and PBL Scheme  
Modifications in the Weather Research and Forecasting (WRF)  
Model

by  
Md. Rafsan Nahian

A Thesis  
presented to  
The University of Guelph

In partial fulfilment of requirements  
for the degree of  
Master of Applied Science  
in  
Mechanical Engineering

Guelph, Ontario, Canada N1G 2W1  
© Md. Rafsan Nahian, August, 2019

## ABSTRACT

Complex Meteorology over Complex Terrains: Assessment of Topography, Land Use, Grid Resolution, and PBL Scheme Modifications in the Weather Research and Forecasting (WRF) Model.

Md. Rafsan Nahian

University of Guelph, 2019

Advisors:

Dr. Amir A. Aliabadi

Dr. Shohel Mahmud

The performance of the Weather Research and Forecasting (WRF) model is evaluated in predicting the meteorological conditions over complex terrains such as an urban area in southwest Ontario and a mining facility in northern Canada. WRF is studied in a series of sensitivity tests by varying topography, land use, horizontal and vertical grid resolutions, and Planetary Boundary-Layer (PBL) parameterization to arrive at optimum configurations for reducing modelling errors in comparison to field observations. Moreover, the seasonal variations of different atmospheric properties as well as forecast versus reanalysis initialization and boundary datasets are also analyzed over the mining facility to understand the meteorological conditions more comprehensively. Overall, the model has presented a better performance in predicting heat-related properties such as temperature and relative humidity compared to the momentum related properties such as wind speed and direction both in the urban region and mining facility during the Summer season although some large variations of temperature and humidity were observed during the Winter season in the mining facility. Accounting for realistic topography, land use, and implementation of a lake model in the mining facility significantly improved the model predictions against field observations. For

instance, model biases were reduced by 0.02 to 0.94 m s<sup>-1</sup> and 3.84 to 47.4 W m<sup>-2</sup> ( $\alpha = 0.1$ ) for surface level wind speed and turbulent sensible heat flux, respectively. Increasing model horizontal and vertical resolutions also reduced model biases by 1.52 to 3.40 m s<sup>-1</sup> and -8.89 to 23.3 W m<sup>-2</sup> ( $\alpha = 0.1$ ) for surface level wind speed and sensible vertical turbulent heat flux, respectively. Overall, the resolution having a horizontal grid size of 0.12 km or 0.09 km in the smallest domain and implementation of 90 vertical levels should be sufficient to provide accurate predictions in comparison to experimental measurements of the mining facility. Yonsei University (YSU) and Large-Eddy Simulation (LES) schemes provided similar performance, although LES diverged numerically for some simulations. However, LES predicted marginally more accurate results for sensible vertical turbulent heat flux than the YSU in most of the WRF simulations by reducing the model bias from -11.4 to 21.1 W m<sup>-2</sup> ( $\alpha = 0.1$ ). Global Forecasting System (GFS) Forecast datasets exhibited better accuracy than the GFS Reanalysis in predicting the meteorological properties both at surface level and higher elevations. Based on the findings of this study, WRF showed better agreement with observations in the Summer season compared to the Winter season. Future investigations should consider anthropogenic heat release at the surface and plume dispersion modelling using WRF.

# Dedication

To my beloved parents, brothers, and sisters,

To all my teachers,

To all my friends.



# Acknowledgements

This work would not have been possible without the reconciliation of Allah the Almighty, the most gracious, and the most merciful. I would like to express my humble and sincere thanks to my advisors Dr. Amir Abbas Aliabadi and Dr. Shohel Mahmud for their endless effort, guidance and support to complete my thesis successfully. Their kind concern and encouragement of thinking new ideas and to implement them practically always helped me to overcome difficulties and keep my progress on schedule. I am thankful to my colleagues, Dr. Manoj K. Kizhakkeniyil, Dr. Mojtaba Ahmadi-Baloutaki, Mohsen Moradi, Seyedahmad Kia, Amir Nazem, and Ryan Byerlay who helped me time to time by giving useful suggestions and guidance to complete this thesis work. I also extend my gratitude to the faculty and staffs at the University of Guelph for their technical and administrative support.

Completing this work would not have been impossible without the technical support of Rowan Williams Davies and Irwin Inc. (RWDI) and financial support from the Discovery Grant program (401231) from the Natural Sciences and Engineering Research Council (NSERC) of Canada, Government of Ontario through the Ontario Centres of Excellence (OCE) under the Alberta-Ontario Innovation Program (AOIP) (053450), and Emission Reduction Alberta (ERA) (053498).

Special gratitude to my parents who have supported me throughout my years of education both emotionally and financially. Finally, I am extremely thankful to my family members and friends for their unconditional love, support, and blessings throughout my life.

# Table of Contents

<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Abbreviations</b>	<b>xvi</b>
<b>List of Mathematical Symbols</b>	<b>xix</b>
<b>1 Introduction and Literature Review</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Literature Review . . . . .	3
1.2.1 Topography and Land Use Classification . . . . .	3
1.2.2 Grid Resolution . . . . .	5
1.2.3 Planetary Boundary-Layer Parameterization . . . . .	6
1.3 Research Gaps . . . . .	7
1.4 Objectives . . . . .	8
1.5 Mining Project, Area Fugitive Emission Measurements and Collaboration De- tails . . . . .	9
1.6 Thesis Structure . . . . .	11
<b>2 Weather Research and Forecasting (WRF) Model</b>	<b>13</b>
2.1 WRF System Components . . . . .	13
2.1.1 WRF Preprocessing System . . . . .	14
2.1.2 WRF-Var System . . . . .	14
2.1.3 Physics and Dynamics Options . . . . .	15
2.1.4 ARW Solver . . . . .	15
2.1.5 WRF-Chem . . . . .	15
2.2 Governing Equations . . . . .	16
2.2.1 Vertical Coordinates . . . . .	16
2.2.2 Euler Equations in ARW Solver . . . . .	17
2.2.3 Projections in ARW Solver . . . . .	18

<b>3</b>	<b>WRF Simulations of Urban Climate</b>	<b>20</b>
3.1	Experimental Observation . . . . .	20
3.1.1	SOmic Detections And Ranging (SODAR) . . . . .	20
3.1.2	Weather Station . . . . .	23
3.2	WRF Model . . . . .	23
3.2.1	Domain Configurations . . . . .	23
3.2.2	Initialization . . . . .	24
3.2.3	Planetary Boundary Layer Parameterization . . . . .	26
3.2.4	Model Evaluation . . . . .	26
3.3	Results and Discussion . . . . .	27
3.3.1	Horizontal Wind Velocity . . . . .	27
3.3.2	Vertical Wind Velocity . . . . .	29
3.3.3	Wind Direction . . . . .	33
3.3.4	Potential Temperature . . . . .	34
3.3.5	Relative Humidity . . . . .	37
3.3.6	Sensible Vertical Turbulent Heat Flux . . . . .	38
3.4	Error Analysis . . . . .	38
3.5	Conclusions . . . . .	43
<b>4</b>	<b>WRF Simulations over a Complex Mining Facility</b>	<b>45</b>
4.1	Experimental Observation . . . . .	45
4.1.1	SOmic Detections And Ranging (SODAR) . . . . .	46
4.1.2	Tethered And Navigated Air Blimp (TANAB) . . . . .	48
4.1.3	3D Ultrasonic Anemometer . . . . .	49
4.1.4	2D Ultrasonic Anemometers . . . . .	49
4.1.5	Weather Stations . . . . .	49
4.1.6	LIDAR Topographical Measurements . . . . .	50
4.2	WRF Model . . . . .	50
4.2.1	Domain Configurations . . . . .	50
4.2.2	Initialization . . . . .	52
4.2.3	Evaluation . . . . .	56
4.3	Results and Discussion . . . . .	58
4.3.1	Effect of Topography and Land Use . . . . .	58
4.3.2	Effects of Grid Resolution and Planetary Boundary Layer Parameterization . . . . .	78
4.3.3	GFS Forecast versus Reanalysis Initial and Boundary Conditions . . . . .	89
4.3.4	Meteorological Properties over the Mining Facility in Winter . . . . .	92
4.4	Conclusions . . . . .	101
<b>5</b>	<b>Conclusions and Further Work</b>	<b>104</b>
5.1	WRF and Topography . . . . .	104
5.2	WRF and Land Use Classification . . . . .	105

5.3	WRF and Grid Resolution . . . . .	106
5.4	WRF and Planetary Boundary-layer Parameterization . . . . .	106
5.5	WRF and Global Forecasting System Initial and Boundary Datasets . . . . .	106
5.6	WRF and Seasonal Variations of Meteorological Properties . . . . .	107
5.7	Future Work . . . . .	107
<b>References</b>		<b>108</b>
<b>A How to Operate the PA-5 SODAR</b>		<b>121</b>
A.1	Turning on the SODAR . . . . .	121
A.2	Setting the Date and Time . . . . .	121
A.3	Changing the Averaging Time . . . . .	121
A.4	Saving the SODAR Data on a USB Flash Memory . . . . .	122
A.5	Turning off the SODAR . . . . .	122
<b>B Spatial Distributions of Various Atmospheric Properties over a Mining Facility in Summer 2018</b>		<b>123</b>
B.1	Summer: Horizontal Wind Velocity at 10 m . . . . .	124
B.2	Summer: Surface Skin Temperature . . . . .	128
B.3	Summer: Potential Temperature at 2 m . . . . .	132
B.4	Summer: Relative Humidity at 2 m . . . . .	136
B.5	Summer: Sensible Vertical Turbulent Heat Flux at 10 m . . . . .	140
<b>C Spatial Distributions of Various Atmospheric Properties over a Mining Facility in Winter 2019</b>		<b>144</b>
C.1	Winter: Horizontal Wind Velocity at 10 m . . . . .	145
C.2	Winter: Wind Direction . . . . .	146
C.3	Winter: Surface Skin Temperature . . . . .	147
C.4	Winter: Potential Temperature at 2 m . . . . .	148
C.5	Winter: Relative Humidity at 2 m . . . . .	149
C.6	Winter: Sensible Vertical Turbulent Heat Flux at 10 m . . . . .	150
<b>D Source Code</b>		<b>151</b>
D.1	Code to Extract Columns of Meteorological Solution Variables from WRF . . . . .	151
D.1.1	Surface Level Extractor . . . . .	151
D.1.2	High Level Extractor for SODAR and TANAB . . . . .	157
D.1.3	Surface Level Extractor for Contour Plots . . . . .	162
D.1.4	High Level Extractor for Profile Plots . . . . .	164
D.2	Code to Perform Statistical Analysis Between Observations and WRF . . . . .	167
D.2.1	WRF Compare to SODAR for Percentile Calculations . . . . .	167
D.2.2	WRF Compare to SODAR for Error Calculations . . . . .	172
D.2.3	WRF Compare to TANAB for Percentile Calculations . . . . .	178
D.2.4	WRF Compare to TANAB for Error Calculations . . . . .	183

D.2.5	WRF Compare to Surface Level for Percentile Calculations . . . . .	187
D.2.6	WRF Compare to Surface Level for Error Calculations . . . . .	192
D.2.7	Two Samples Statistical Estimation . . . . .	197
D.3	Code to Plot and Show the Comparison between Observations and WRF . .	198
D.3.1	WRF Plotter to Observe the Hourly Variations of Meteorological Prop- erties . . . . .	198
D.3.2	WRF Contour Plotter to Analyze Spatial Distribution . . . . .	225
D.3.3	WRF Profile Plotter . . . . .	230
<b>E</b>	<b>Published Work</b>	<b>241</b>
E.1	Peer-Reviewed Journal Papers . . . . .	241
E.2	Refereed Conferences . . . . .	242
E.3	Poster Presentations . . . . .	242

# List of Tables

3.1	Domain configurations and associated parameters. . . . .	25
3.2	Effect of topography showing the FB and NMSE of different atmospheric properties. . . . .	43
4.1	Measured parameters by PA-5 SODAR. . . . .	47
4.2	Domain configurations and associated parameters. . . . .	51
4.3	Effects of topography and land use change on atmospheric dynamics; the table shows the bias and RMSE of different atmospheric properties between the WRF model and field observations. . . . .	59
4.4	Effects of topography and land use changes on atmospheric dynamics showing the two-sample estimation test ( $\alpha=0.1$ ) among different cases; a range for difference in bias is reported for pairs of model configurations. . . . .	59
4.5	Effects of grid resolution and planetary boundary layer parameterization on bias and RMSE of different atmospheric properties for model-observation comparison. . . . .	82
4.6	Effects of GFS datasets on WRF simulations; table shows the bias and RMSE of different atmospheric properties between the WRF model and field observations. . . . .	91
4.7	Seasonal variations of various atmospheric properties in the mining facility; table shows the bias and RMSE between the WRF model and field observations.	93

# List of Figures

2.1	WRF system components [106]. . . . .	14
2.2	Vertical coordinate system of Advanced Research WRF (ARW) [106]. . . . .	16
2.3	Horizontal and vertical grid spacing in ARW solver [106]. . . . .	18
3.1	Working principle of three beam SODAR [10]. . . . .	22
3.2	WRF simulation domains with latitude and longitude centred at Guelph, Ontario, Canada. . . . .	24
3.3	Surface height from sea level: GTOPO 30s has low resolution; SRTM 1s has high resolutions. . . . .	25
3.4	Land use classification of GTOPO 30s and SRTM 1s. . . . .	26
3.5	Effect of topography on horizontal wind velocity at 10 m, 70 m, and 170 m; The error bars represent the Normalized Mean Square Error (NMSE). . . . .	28
3.6	Effect of topography on 10-m horizontal wind velocity and direction at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	30
3.7	Effect of topography on 10-m horizontal wind velocity magnitude and direction at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	31
3.8	Horizontal wind velocity profile based on pressure height on August 4, 2018. . . . .	32
3.9	Effect of topography on vertical wind velocity at 70 m and 170 m; The error bars represent the Normalized Mean Square Error (NMSE). . . . .	32
3.10	Effect of topography on wind direction at 70 m and 170 m showing the median values of wind direction bias for each hour. . . . .	33
3.11	Fractional bias of potential temperature at 2 m; The error bars represent the Normalized Mean Square Error (NMSE). . . . .	34
3.12	Effect of topography on 2-m potential temperature at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	35
3.13	Effect of topography on 2-m potential temperature at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	36
3.14	Potential temperature profile based on pressure height on August 4, 2018. . . . .	37
3.15	Fractional bias of relative humidity at 2 m. The error bars represent the Normalized Mean Square Error (NMSE). . . . .	38
3.16	Effect of topography on 2-m relative humidity at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	39

3.17	Effect of topography on 2-m relative humidity at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	40
3.18	Effect of topography on 10-m vertical turbulent heat flux at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	41
3.19	Effect of topography on 10-m vertical turbulent heat flux at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT). . . . .	42
4.1	Locations of the meteorological instruments deployed at the mining facility for model comparison in the Summer campaign; Figure color coded with surface height above sea level. . . . .	46
4.2	Horizontal wind velocity (a) and temperature (b) distribution for the entire month of May in 2018 as measured by the Wood Buffalo Environmental Association (WBEA) weather station indicated in Figure 4.1. . . . .	47
4.3	WRF simulation domains centred at the mining facility colour coded with surface height above sea level. . . . .	51
4.4	Surface height from sea level: GTOPO 30s is the non-modified land; SRTM 1s with LIDAR contains land modifications showing the mine and the tailings pond. . . . .	53
4.5	Topography and land use classifications for WRF simulations. . . . .	54
4.6	Wind direction bias calculation for different cases to report the bias within $\pm 180^\circ$ . . . . .	57
4.7	Effects of topography and land use changes on horizontal wind velocity near surface level showing the median values of model and observation for each hour; here the instruments were set up at 10 m elevations from the surface except for the TANAB. . . . .	60
4.8	Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 0200 MST on May 18, 2018; times in Mountain Standard Time (MST). . . . .	62
4.9	Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 1400 MST on May 18, 2018; times in Mountain Standard Time (MST). . . . .	63
4.10	Horizontal wind velocity profiles based on pressure height at different locations on May 18, 2018; times in Mountain Standard Time (MST). . . . .	64
4.11	Effects of topography and land use changes on bias of wind direction at 10 m elevations from the surface showing the median values of wind direction bias for each hour. . . . .	66
4.12	Wind direction profiles based on pressure height at different locations on May 18, 2018; times in Mountain Standard Time (MST). . . . .	67
4.13	Effect of topography and land use changes on temperature at 2 m showing the median values of model and observation for each hour. . . . .	68
4.14	Effect of topography and land use changes on surface skin temperature at 0200 MST on May 18, 2018. . . . .	69



4.15	Effect of topography and land use changes on surface skin temperature at 1400 MST on May 18, 2018. . . . .	70
4.16	Effect of topography and land use changes on 2-m potential temperature at 0200 MST on May 18, 2018. . . . .	71
4.17	Effect of topography and land use changes on 2-m potential temperature at 1400 MST on May 18, 2018. . . . .	72
4.18	Potential temperature profiles based on pressure height at different locations on May 18, 2018; times in Mountain Standard Time (MST). . . . .	73
4.19	Effect of topography and land use changes on relative humidity at 2 m showing the median values of model and observation for each hour. . . . .	75
4.20	Effect of topography and land use changes on 2-m relative Humidity at 0200 MST on May 18, 2018. . . . .	76
4.21	Effect of topography and land use changes on 2-m relative Humidity at 1400 MST on May 18, 2018. . . . .	77
4.22	Effect of topography and land use changes on sensible vertical turbulent heat flux at 10 m. . . . .	78
4.23	Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 0200 MST on May 18, 2018. . . . .	79
4.24	Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 1400 MST on May 18, 2018. . . . .	80
4.25	Effect of grid resolution on horizontal wind velocity near surface level showing the median values of model and observation for each hour. . . . .	83
4.26	Effect of PBL parameterization on horizontal wind velocity near surface level showing the median values of model and observation for each hour. . . . .	84
4.27	Effect of grid resolution on wind direction at 10 m showing the median values of wind direction bias for each hour. . . . .	85
4.28	Effect of PBL parameterization on wind direction at 10 m showing the median values of wind direction bias for each hour. . . . .	86
4.29	Effect of grid resolution on temperature at 2 m showing the median values of model and observation for each hour. . . . .	87
4.30	Effect of PBL parameterization on temperature at 2 m showing the median values of model and observation for each hour. . . . .	88
4.31	Effect of grid resolution on relative humidity at 2 m showing the median values of model and observation for each hour. . . . .	88
4.32	Effect of PBL parameterization on relative humidity at 2 m showing the median values of model and observation for each hour. . . . .	89
4.33	Effect of grid resolution on sensible vertical turbulent heat flux at 10 m showing the median values of model and observation for each hour. . . . .	90
4.34	Effect of PBL parameterization on sensible vertical turbulent heat flux at 10 m showing the median values of model and observation for each hour. . . . .	90
4.35	Horizontal wind velocity at 10 m elevation in Winter 2019 showing the median values of model and observation for each hour. . . . .	94

4.36	Horizontal wind velocity at 10 m in Winter on March 16, 2019; times in Mountain Standard Time (MST). . . . .	94
4.37	Horizontal wind velocity profiles based on pressure height at different locations on March 16, 2019; times in Mountain Standard Times (MST). . . . .	95
4.38	Bias of wind direction at 10 m elevation in Winter 2019 showing the median values of wind direction bias for each hour. . . . .	96
4.39	Temperature and relative humidity at 2 m in Winter 2019 showing the median values of model and observation for each hour. . . . .	98
4.40	Surface skin temperature on March 16, 2019. . . . .	98
4.41	Potential temperature at 2 m on March 16, 2019. . . . .	99
4.42	Relative Humidity at 2 m on March 16, 2019. . . . .	99
4.43	Potential temperature profiles based on pressure height at different locations on March 16, 2019; times in Mountain Standard Time (MST). . . . .	100
4.44	Sensible vertical turbulent heat flux at 10 m in Winter 2019 showing the median values of model and observation for each hour. . . . .	101
4.45	Sensible vertical turbulent heat flux at 10 m on March 16, 2019. . . . .	102
B.1	Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 0600 MST in domain 5. . . . .	124
B.2	Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 1000 MST in domain 5. . . . .	125
B.3	Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 1800 MST in domain 5. . . . .	126
B.4	Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 2200 MST in domain 5. . . . .	127
B.5	Effect of topography and land use changes on surface skin temperature at 0600 MST in domain 5 on May 18, 2018. . . . .	128
B.6	Effect of topography and land use changes on surface skin temperature at 1000 MST in domain 5 on May 18, 2018. . . . .	129
B.7	Effect of topography and land use changes on surface skin temperature at 1800 MST in domain 5 on May 18, 2018. . . . .	130
B.8	Effect of topography and land use changes on surface skin temperature at 2200 MST in domain 5 on May 18, 2018. . . . .	131
B.9	Effect of topography and land use changes on 2-m potential temperature at 0600 MST in domain 5 on May 18, 2018. . . . .	132
B.10	Effect of topography and land use changes on 2-m potential temperature at 1000 MST in domain 5 on May 18, 2018. . . . .	133
B.11	Effect of topography and land use changes on 2-m potential temperature at 1800 MST in domain 5 on May 18, 2018. . . . .	134
B.12	Effect of topography and land use changes on 2-m potential temperature at 2200 MST in domain 5 on May 18, 2018. . . . .	135

B.13	Effect of topography and land use changes on 2-m relative Humidity at 0600 MST in domain 5 on May 18, 2018. . . . .	136
B.14	Effect of topography and land use changes on 2-m relative Humidity at 1000 MST in domain 5 on May 18, 2018. . . . .	137
B.15	Effect of topography and land use changes on 2-m relative Humidity at 1800 MST in domain 5 on May 18, 2018. . . . .	138
B.16	Effect of topography and land use changes on 2-m relative Humidity at 2200 MST in domain 5 on May 18, 2018. . . . .	139
B.17	Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 0600 MST in domain 5 on May 18, 2018. . . . .	140
B.18	Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 1000 MST in domain 5 on May 18, 2018. . . . .	141
B.19	Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 1800 MST in domain 5 on May 18, 2018. . . . .	142
B.20	Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 2200 MST in domain 5 on May 18, 2018. . . . .	143
C.1	Horizontal wind velocity and wind direction at 10 m on March 16, 2019. . .	145
C.2	Wind direction profiles based on pressure height at different locations on March 16, 2019. . . . .	146
C.3	Surface skin temperature on March 16, 2019. . . . .	147
C.4	Potential temperature at 2 mon March 16, 2019. . . . .	148
C.5	Relative Humidity at 2 mon March 16, 2019. . . . .	149
C.6	10-m sensible vertical turbulent heat flux on March 16, 2019. . . . .	150

# List of Abbreviations

ACM	Asymmetric Convective Model
ARW	Advanced Research WRF
ASP	Acoustic Signal Processor
ABL	Atmospheric Boundary Layer
AVHRR	Advanced Very High Resolution Radiometer
CPUs	Central Processing Units
CFD	Computational Fluid Dynamics
CLC	CORINE Land Cover
CORINE	Coordination of Information on the Environment
DEM	Digital Elevation Model
ECMWF	European Centre for Medium Range Weather Forecasting
EM	Eulerian Mass
FB	Fractional Bias
FFT	Fast Fourier Transform
GCM	General Circulation Model
GFS	Global Forecasting System
GHG	Green House Gas
GLCC	Global Land Cover Characterization
GTOPO30	Global 30 Arc-Second
HTTP	Hyper Text Transfer Protocol
IPCC	Intergovernmental Panel on Climate Change
LDT	Local Daylight Time

LES	Large-Eddy Simulation
LIDAR	LIght Detection And Ranging
LSM	Land Surface Model
LU	Land Use
MM5	Fifth Generation Mesoscale Model
MODIS	MODerate Resolution Imaging Spectroradiometer
MST	Mountain Standard Time
MYJ	Mellor-Yamada-Janjić
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
NFS	Non-hydrostatic Forecast System
NMC	National Meteorological Center
NMM	Non-hydrostatic Mesoscale Model
NMSE	Normalized Mean Square Error
NWP	Numerical Weather Prediction
PBL	Planetary Boundary Layer
RMSE	Root Mean Square Error
SL	Surface Layer
SODAR	SONic Detections And Ranging
SRTM	Shuttle Radar Topography Mission
TANAB	Tethered And Navigated Air Blimp
UEMS	Unified Environmental Modelling System
UMD	University of Maryland
USGS	United States Geological Survey
WBEA	Wood Buffalo Environmental Association
WHO	World Health Organization
WMO	World Meteorological Organization
WPS	WRF Preprocessing System
WRF	Weather Research and Forecasting

WRFDA

WRF Data Assimilation System

YSU

Yonsei University

# List of Mathematical Symbols

## Latin Symbols

$c$	Velocity of Sound
$CT$	Echo Strength
$C_T$	Temperature Structure Function
$C_V$	Velocity Structure Function
$D$	Distance
$f$	Frequency
$F$	Force
$g$	Gravitational Acceleration
$hr$	Hour
$k$	Wave Number
$m_{x,y}$	Map Scale Factor
$M$	Model Results
$N_f$	Number of Points in the Spectrum
$O$	Observation Results
$p$	Pressure
$p_h$	Hydrostatic Pressure
$p_{hs}$	Hydrostatic Pressure at Surface Level
$p_{ht}$	Hydrostatic Pressure at Top of the Domain
$p_0$	Reference Pressure
$R_d$	Gas Constant of Dry Air
$SU$	Standard Deviation of Horizontal Wind Speed (Along Wind)

$SV$	Standard Deviation of Horizontal Wind Speed (Cross Wind)
$SW$	Standard Deviation of Vertical Wind Speed
$t$	Time
$T$	Temperature
$U$	Horizontal Velocity along $X$ Axis
$V$	Horizontal Velocity along $Y$ Axis
$\boldsymbol{v}$	Covariant Velocity
$W$	Vertical Velocity along $Z$ Axis
$z$	Height

## Greek Symbols

$\alpha$	Inverse of Density
$\Delta$	Implies difference
$\eta$	Vertical Mass Coordinate
$\gamma$	Heat Capacity Ratio
$\mu$	Pressure Difference between Surface Level and Top of the Domain
$\eta$	Vertical Mass Coordinate
$\dot{\eta}$	Contravariant Velocity
$\kappa$	Von Kármán Constant
$\Omega$	Flux Form Variable as a Function of Contravariant Velocity
$\phi$	Wind Direction / Non-conserved Variable
$\varphi$	Beam Width
$\rho$	Density
$\sigma$	Scattered Acoustic Power
$\tau$	Acoustic Pulse Duration
$\theta$	Potential Temperature
$\Theta$	Flux Form Variable as a Function of Potential Temperature



# Chapter 1

## Introduction and Literature Review

### 1.1 Background

The change of weather is one of the major global concerns which is affected by solar radiation, continental drift, volcano eruptions, changes in earth's orbit, oceanic circulations, and mostly by the human civilization [129]. Weather change cause global-scale warming, sea-ice melting, rise in global mean sea level, and surge the repetition of extreme weather conditions such as floods, droughts, wildfires, and cyclonic storms [65]. The Intergovernmental Panel on Climate Change (IPCC) stated that human impacts are more than the natural influences on weather change [43]. Moreover, the northern latitudes are becoming warmer in recent years due to global warming. According to the World Health Organization (WHO), many human diseases are caused due to the change of temperature in the atmosphere. Hence, weather change has a significant impact on human living conditions that affect the ecosystem [92]. The characteristics of global weather changes in historic, current, and future periods have been studied substantially due to their significant impacts. The global weather forecasting models are used for simulating global weather systems which identify the physical processes among the oceans, the atmosphere, and the land surface. However, these models are only capable of simulating global-scale weather change and variabilities. Parameterization is one source of uncertainties in global model simulations of weather change. Moreover, uncertainties in various mechanisms in models including water vapor, radiation, clouds, ocean circulation, ice, and snow also deviate the simulation results for the same initial boundary conditions [129]. Since the mesoscale processes are not expressly represented in global weather models, one solution is to choose the regional modelling approach with higher resolution. A regional weather model is to some extent similar to a global weather model having higher resolutions

than a global weather model. Therefore, regional weather models are used to resolve more details than global weather models [89]. Besides, the upscale influence of regional forcing on large scale weather change can also be described with more accuracy by using regional mesoscale models over the global weather models [74].

An alternative to weather models, a high-resolution weather model can describe regional topography, land use, and vegetation aspects more accurately and at much greater spatiotemporal resolutions than the global weather models [108]. Regional mesoscale weather forecasting models have been used to study short term weather change and air pollution processes in remote environments at high resolutions [100]. The regional weather models have already been employed to simulate the short term climate transition in distinct zones including North America, Asia, Europe, and Africa [3, 98]. Two widely used weather models are the Weather Research and Forecasting (WRF) and European Centre for Medium-Range Weather Forecasting (ECMWF) models. The analysis from ECMWF is used to observe the atmospheric properties and as boundary conditions for the Weather Research and Forecasting (WRF) model [77]. It is also observed that in terms of predicting temperature near surface level WRF and ECMWF forecasts are better than other regional mesoscale models [76]. Moreover, the temperature at 2 m is well represented by both WRF and ECMWF simulations with hourly experimental datasets. The advantage of WRF over ECMWF for atmospheric transport modelling is adequate to motivate further assessment to compare the outputs of model  $CO_2$  mixing ratios with experimental measurements in an urban area [69]. It has been demonstrated that WRF exhibits lower bias than ECMWF in meteorological measurements, compared to aircraft observations in northern latitudes [8]. It is also observed that forecasts executed by the WRF and ECMWF model are almost similar to in-situ observations although WRF predicts the rainfall more accurately compared to the other regional models [64]. WRF can capture the effect of gravity wave dynamics and momentum and energy propagation, while the vertical structures of simulated potential energy profiles also show good agreement with LIDAR measurements [33].

Such models should be further investigated over complex terrains with possibly thermally stable boundary layers, particularly at night time, due to the inherent difficulty to simulate the stable boundary layer characterized by weak and intermittent turbulent mixing [6, 7]. Also, the success of the WRF model with applications in complex topography and land use in the open-pit mining facility is yet to be investigated in more detail. WRF is a Numerical Weather Prediction (NWP) model which has been developed to combine new and existing atmospheric models within a common software structure. The software architecture of WRF

is developed with a high degree of pliability, which plays a crucial role to access accurate predictions with high-reliability [1]. In WRF, the dynamical cores and physics packages of distinct atmospheric models are associated for performance optimization as well as for intermodal correlations by conducting sensitivity studies [106]. WRF is applicable for an extensive span of operations including large-eddy to global simulations, real-time NWP, regional weather simulations, air quality modelling, atmosphere-ocean coupled simulations, and idealized simulations [81, 97, 106].

## 1.2 Literature Review

Land surface, surface layer, and Planetary Boundary-Layer (PBL) parameterizations are some important parameters in the atmospheric models to account for the transfer of heat, moisture, and momentum between the surface and the atmosphere [46]. WRF exhibits an excessive bias of surface-level wind velocity in the earlier versions [29]. The bias still exists in the latest versions, which imposes a limitation for accurate predictions of atmospheric properties [101]. Since the present mesoscale models exhibit extensive configurations and physical parameters, selecting the optimum configuration imposes numerous challenges [84]. The following subsections review some important configurations of the mesoscale models such as the topography and land use classification, grid resolution, and the PBL parameterization.

### 1.2.1 Topography and Land Use Classification

The inaccuracy of the real terrain representation is one of the main limitations of WRF as for accurate predictions, topographic and land use data are important inputs for weather forecasting models, and high-resolution terrain data plays a significant role to represent the complex characteristics of topography [58, 71]. Moreover, the topography also plays an important role in predicting the sea breeze and urban heat island circulations [75]. The complex terrains influence the meteorology by altering various atmospheric parameters such as wind speed and direction, albedo, and surface heat flux [25]. The coarser topographic resolutions and redundant land surface datasets cannot capture the fine-scale and realistic heterogeneities, which are very crucial to WRF simulation. In addition, the improper depiction of land surface parameters from the meteorological datasets induces uncertainty in the WRF simulations [125]. In recent years, with the availability and ease of access to satellite data, many studies have emphasized the importance of high-resolution land surface boundary

datasets in producing more precise weather forecasts with finer spatial resolutions [63]. The elevation has a significant impact on WRF model not only for representing the nature of real topography but also the landform representation derived from the Digital Elevation Model (DEM) that modifies the grid discretization [117]. The predictability of surface temperature also improves in the WRF model when updated fractional vegetation cover datasets from the Advanced Very High-Resolution Radiometer (AVHRR) are implemented [66]. When surface elevation, land use index, vegetation fraction, and soil type datasets are modified for WRF simulations, the root means square errors of temperature and precipitation are greatly reduced [51]. The multi-scale terrain data is more sensitive in WRF than the model resolutions for accurate forecasting [135]. The regional circulations are also affected by topographic properties, and thus play important roles in a mesoscale model by altering the precipitation, wind velocity, and heat-related properties [54, 127]. The high-resolution WRF simulations show more accuracy in predicting the spatial distribution of precipitation during thermally stable and unstable conditions [61]. When the topographic resolution is reduced by half of the real topography in a WRF model, the rainfall distribution significantly changes by underestimating the rainfall amount with varying magnitudes [85]. The wind direction of WRF is compared with in-situ observations and it is observed that the bias is more over the complex terrain compared to the flat regions. Moreover, when the speed is higher, the bias of wind direction is lower than low-velocity wind flow [59]. However, some limitations in WRF model can arise for higher topographical resolutions as smoothing is also necessary to minimize the error of the horizontal pressure gradient for substantial variations in elevation between the contiguous grid cells [13].

The terrain characteristics are generally implemented by land surface parameterization in the WRF model, primarily based on topographical properties associated with different land-use categories [103]. Hence, the WRF model includes land-use classifications derived from remote sensing instruments to assign the actual land surface properties [104]. A land-use map is developed for the Fifth-Generation National Center for Atmospheric Research (NCAR) Mesoscale Model (MM5) based on the Coordination of Information on the Environment (CORINE) Land Cover (CLC) 2000 [93]. The default 24-category Global Land Cover Characterization (GLCC) is implemented in the WRF model with a spatial resolution of 30 meters although significant bias of temperature is observed after evaluation [105]. The impact of three different land-cover modifications is estimated by executing numerous WRF simulations using the U.S. Geological Survey (USGS) and University of Maryland (UMD) datasets which are based on the Advanced Very High-Resolution Radiometer (AVHRR),

and from the MODerate Resolution Imaging Spectroradiometer (MODIS). It is observed that UMD and MODIS predict greater wind speed than GLCC near surface level in the southern part of Texas although the phenomena are opposite for southern Wyoming [113]. The influence of Shuttle Radar Topography Mission (SRTM) topography of 100-m spatial resolution CLC datasets on WRF simulations are also evaluated to predict wind speed and temperature near surface level. It was concluded that there was  $0.1$  to  $0.4 \text{ m s}^{-1}$  bias of wind velocity in WRF compared to the experimental datasets in the Lombardy region of Italy [38]. Alteration of the local and regional climate is also observed due to the land use classification by affecting the energy and water balances between the surface and atmosphere [114]. A WRF sensitivity test has been conducted in the northern part of Italy wherein high resolution topographic and land use data were implemented [38]. The simulation results showed that the wind speed and the temperature changed due to surface friction and high roughness length in that region. The WRF simulation was conducted in Hong Kong by using different resolutions of topographic data and the results showed that the temperature and relative humidity were close to the experimental values when high-resolution topographic data were used, demonstrating that high-resolution topographic data should be applied to execute numerical models in any region [131]. The WRF simulations with high topographical resolution and land use modification also improve the predictability of precipitation, potential temperature, and heat flux although it underestimates the wind speed and greenhouse gas concentration [37]. Moreover, the CLC dataset in WRF simulations has improved the prediction of wind speed in the urban region [103]. Different Land Surface Models (LSMs) usually vary the WRF simulations for resolving low-level jets and also influence the surface level energy budget [36]. Furthermore, the periodic updating for better representation of land use index also improves the performance of atmospheric air quality models [32].

### 1.2.2 Grid Resolution

The importance of conducting sensitivity tests of a numerical model has been emphasized by changing its physical parameters along with different initialization fields [52]. It is also identified that WRF is suitable for short term forecasting, reanalysis, assimilating in-situ LIDAR observations, and selecting the appropriate PBL scheme [23]. The reduction of bias in the simulation can be accomplished by testing and selecting an optimum physical configuration for the region of interest. Although the high grid resolutions may present a better representation of fine-scale meteorological processes, this may not be compelling

due to the contingency in the overall performance of the various physical parameters [24]. The high horizontal and vertical grid spacing in WRF is increasingly feasible due to the improvement of computational resources although the wind flow with refined spacing is unrealistic in the convective boundary layer [134]. Although the numerical solution of the largest eddies is sensitive to grid resolution during convective hours, such phenomenon is mostly not represented in thermally stable condition for the smaller size of eddies [68]. However, higher grid resolutions provide opportunities to obtain more information from WRF simulations, but they require further numerical testing [107]. Moreover, improvement in the vertical grid resolution is also important for simulating low-level jets as sharp gradients are difficult to determine with coarse vertical grid resolutions [78]. The sensitivity studies by some researchers showed that increasing the grid resolutions improved the model simulations in complex orographic zones [73, 96]. Due to high spatial grid resolutions in the WRF model, using modified land use classification can reduce errors in predicting surface properties. Furthermore, it also affects the mesoscale circulations over complex terrains [128]. A study over two wind farms in California and Texas shows that the higher grid resolutions in the WRF model reduce the error of low-level wind prediction due to improved representation of terrain characteristics. Moreover, the physical parameters also influenced the numerical results in California although it did not in the Texas region [30].

### 1.2.3 Planetary Boundary-Layer Parameterization

Several physical parameterizations are described such as microphysics, radiation, and clouds as well as the surface layer (SL), the PBL, and the LSM. Such schemes have nonlinear interactions with each other and hence, optimization of the model is difficult due to these complex relationships [106]. Although the WRF model can identify some important characteristics of low-level jet events over the west Texas and southern Kanas region, it is suggested to run some sensitivity tests by varying the PBL parameterizations as in stable hours when the WRF model has the poor capability to determine the actual properties of the atmospheric boundary layer [109]. In WRF, local and nonlocal closure schemes are the two options to parameterize the PBL processes [110]. The sub-grid scale eddies and large asymmetrical thermals play an important role to accomplish the mixing during convective hours. Turbulence kinetic energy theory in the local closure schemes is limited to simulate turbulent mixing within the adjacent layers uniformly. Hence numerous nonlocal closure models have been established to overcome the limitations of turbulence kinetic energy theory in the local

closure schemes [123]. The nonlocal upward transport can be introduced either as a parameterized adjustment term to represent large-scale motions for medium-range forecasting or by particularly evaluating the movement of conserved atmospheric scalars [95]. The vertical diffusion in the medium-range forecast PBL scheme is revised and then renamed as Yonsei University (YSU) PBL scheme to evaluate the entrainment mechanism [53]. WRF simulations are executed in the United States to evaluate the performances of five different PBL parameterizations and it is found that nonlocal PBL schemes give less error to predict the diurnal variation of wind near surface level due to direct exchanges of mass and momentum between the surface and unstable PBL [132]. Moreover, the nonlocal PBL schemes based on bulk Richardson number produce more well-mixed convective boundary layers than the local PBL schemes [20]. WRF simulations are executed during summer in the south-central United States with Mellor-Yamada-Janjić (MYJ), Yonsei University (YSU), and Asymmetric Convective Model, version 2 (ACM2) schemes. It was concluded that the entrainment process is weaker in MYJ PBL scheme at the top of PBL predicting lower temperature and higher relative humidity than the other two PBL schemes near surface level [55]. The dispersion process of pollutants is highly affected due to vertical turbulent mixing among the plume models. It is suggested to understand the performance of PBL parameterizations to evaluate mixing characteristics of turbulence properties near surface level and higher elevations, which is an important factor in simulating mesoscale transport and pollution dispersion [50]. The uncertainties from the model physical parameterization schemes such as cumulus convection, land use, and planetary boundary layer (PBL) are sources of error in WRF [121]. Such model errors are due to the different numerical solvers, domain and grid size, regional topography, boundary conditions, terrain, vegetation characteristics, and grid resolution both in horizontal and vertical directions [15].

### 1.3 Research Gaps

The influence of topographic effects on the surface radiation and energy budgets in complex terrain is studied both for the valley and sloped flows [62]. Although the representation of surface heat fluxes over complex terrain is improved by better representation of topography, the terrain-following vertical coordinate used in the WRF model becomes misrepresented over moderate slopes that result in numerical errors [130]. This limitation restricts the implementation of high-resolution topography during WRF simulations in a complex terrain where topographic effects on radiation are most important [14]. Moreover, two kinds of

problems occur during modifying the land properties for accurate prediction of wind in WRF. The limitation is due to grid resolution which is usually higher in the WRF model for wind estimation. Since the spatial variability of land use modification can be less than the assigned grid resolution in WRF, the use of 1-km land use classifications can result in significant errors during predicting the atmospheric properties near surface level by rendering to a poor representation of terrain characteristics, which affects the regional circulations [128]. Another problem is due to the time period of land use modification as it changes continuously with time. Hence, the swiftly outdated land use libraries can cause misrepresentation of land use classification for WRF simulations [105].

Although the effects of grid resolution variations on simulating atmospheric wind flow over complex terrains are studied [47, 88, 115], there are few studies where the mesoscale approach with higher horizontal and vertical grid resolutions are applied. This is because when WRF simulations are run with high grid resolution, it is assumed that the energy-containing eddies are much smaller than the grid elements which depends on the atmospheric stability [122]. The Large-Eddy Simulation (LES) approach is required for any size of turbulent eddies which is larger than the grid spacing. As most of the extensively available land use datasets do not have a high resolution, few WRF simulations have been executed to simulate the atmospheric properties for higher grid resolutions [60].

## 1.4 Objectives

The understanding of the spatial and temporal variability of various meteorological properties such as wind speed, wind direction, temperature, humidity, and heat flux is still lacking for open-pit mining environments. Moreover, there is a lack of observation both in surface level and higher elevations due to the adverse and remote environments where collecting experimental observations are highly challenging. Therefore, more work is required to better understand the regional microscale meteorological phenomena over complex open-pit mining terrains. The inadequacy of experimental data, coarse grid resolution, and simplified PBL parameterization make it difficult for models to represent the complex microscale and mesoscale processes simultaneously in a region. Furthermore, the high spatial variability of a complex terrain can result in significant variations of surface-level atmospheric properties. Hence, WRF simulations should be executed with proper terrain characteristics and land use classification to understand the real meteorological phenomena on regions of interest.

Although a number of studies have been conducted to improve the prediction perfor-



mance of the WRF model by using high-resolution data on topographic datasets, there has been a lack of research involving WRF in the open-pit mining facility. Specifically, the effects of high topographic resolution, modified land use, grid resolutions in the horizontal and vertical directions, and PBL parameterization, on the accuracy of the model results require more research. Moreover, most of the experimental observations are conducted in surface level by using various meteorological instruments on a meteorological tower. It is still a challenge for researchers to initialize high topographic resolution datasets with actual land use classification for WRF simulations as well as to observe the atmospheric properties in higher altitudes due to the complexity and unavailability of expensive weather sensing instruments. Hence, the main objectives of this study are to model (using WRF) and experimentally observe the meteorological properties over a complex open-pit mining facility in northern Canada both in surface level and higher elevations, and to perform a comparative analysis between the model and the observations. Further, the model is investigated in a series of sensitivity studies to assess the effects of terrain topographic resolution, land use change, grid resolution, and PBL parameterization, on the accuracy of its predictions. Before applying the WRF model to the open-pit mine, it is tested to predict the meteorological conditions over an urban area. This test provides a proof of concept before extending the model’s application for open-pit mine simulations.

## **1.5 Mining Project, Area Fugitive Emission Measurements and Collaboration Details**

This thesis provides the modelling support for a multi-institution research project that aims to quantify area fugitive emissions from an open-pit mining facility in northern Canada. The detailed information about the project, such as the name of mine, location, and client specification, cannot be disclosed due to a non-disclosure agreement with the industrial partner. However, the generic information is provided to help propel research in the topic of area fugitive emissions. The research project employs numerous graduate students and research staff members at the University of Guelph. In addition, various other industrial partners are also listed with specific roles of each entity detailed below.

The Tethered And Navigated Air Blimp (TANAB) is an experimental platform used in this project to probe atmospheric profiles of wind speed, wind direction, pressure, and relative humidity. TANAB was deployed in the open-pit mining facility in May 2018 by Amir

Nazem, Md. Rafsan Nahian, Ryan Byerlay, Manoj K. Nambiar, and Amir A. Aliabadi. The measurements were conducted to cover the full diurnal cycle as well as different locations in the mining facility including the mine and near a tailings pond. The collected data were post-processed by Amir Nazem with the assistance of Manoj K. Nambiar and Amir A. Aliabadi.

A SONic Detection And Ranging (SODAR) instrument measured profiles of atmospheric wind speed and direction within the planetary boundary layer. This instrument was contributed to the project by RWDI and was deployed in the open-pit mining facility in May 2018 by Amir Nazem, Md. Rafsan Nahian, Ryan Byerlay, Manoj K. Nambiar, and Amir A. Aliabadi. The collected data were post-processed by the author.

Numerous partners in the project collected surface level meteorological data including temperature and relative humidity at 2 m, wind speed and direction at 10 m, and sensible vertical turbulent heat flux at 10 m. The partners were RWDI, Southern Alberta Institute of Technology (SAIT), and the University of Alberta (UofA). They post-processed the data and provided them to the author of this work for model comparison.

In the modelling approach, Amir Nazem updated the Weather Research and Forecasting (WRF) software with the latest topography, land use classification, and lakes at the mining facility. The author, exclusively conducted a series of sensitivity simulations in WRF to quantitatively research the model's performance with respect to: 1) grid resolutions, 2) land use modifications, 3) Planetary Boundary Layer (PBL) parameterizations, 4) forecast and reanalysis initialization datasets, and 5) seasonal variations. The author also rigorously compared WRF output results with experimental measurements of meteorological variables in the field.

The author with the assistance of Amir A. Aliabadi developed numerous Python codes to post-process and analyze the WRF model's output results and compared them to field observations statistically. A Ph.D. student, Seyedahmad Kia, is employing a Computational Fluid Dynamics (CFD) program to model the emissions flux over the mining facility. He obtains the initial and boundary conditions from WRF predictions in this work and runs the model accordingly, using the Open Fields Operation And Manipulation (FOAM) software, at higher spatiotemporal resolutions at different atmospheric stability and wind conditions. Meanwhile, Manoj K. Nambiar is running WRF simulation with plume dispersion to quantifying the area fugitive emission flux.

## 1.6 Thesis Structure

The structure of this thesis is as follows.

**Chapter 1:** It provides a brief discussion of the background of mesoscale models, literature reviews, and research gaps regarding modification of topography, land use, grid resolution, and PBL parameterization in the Weather Research and Forecasting (WRF) Model. It offers the context for meteorological modelling activity over a complex open-pit mining facility and outlines the roles and responsibilities of various other entities involved in a large scale project.

**Chapter 2:** It provides the methodology, working principle, and governing equations of the Weather Research and Forecasting (WRF) model briefly.

**Chapter 3:** It provides the outcomes of changing the topography resolution in WRF simulations over an urban area. Section 3.1 gives details of the field meteorological campaign and the experimental observations, which were conducted in the Guelph Turfgrass area. Section 3.2 represents the details for the WRF model including domain configurations, initialization, topography, land use, grid resolution, PBL parameterization, and validation. Section 3.3 shows the results of different atmospheric properties as a function of topography in the urban area. This chapter was published in the proceedings of the Joint Canadian Society for Mechanical Engineering (CSME) and CFD Society of Canada (CFDSC) International Congress (2019).

**Chapter 4:** It provides the outcomes of implementing the modified topography, land use, grid resolution, and PBL parameterization in WRF simulation over a complex open-pit mining facility. Section 4.1 gives details of the field meteorological campaign and the experimental observations. Section 4.2 represents the details for the WRF model including domain configurations, initialization, topography, land use, grid resolution, PBL turbulence parameterization, and evaluation. Section 4.3 shows the results of different atmospheric properties over the mining facility as a function of topography, grid resolution, and turbulence parameterization. Moreover, the seasonal variations of various meteorological properties, as well as initialization datasets are also discussed in this section. Some parts of this chapter (Section 4.1 to 4.3.2) are submitted to the Journal of Applied Meteorology and Climatology (JAMC) as a manuscript.

**Chapter 5:** It provides the conclusions, recommendations, and further works based on the findings of the WRF simulations and comparison to observations in this work.

**Appendices:** They describe the operating procedures of the PA-5 SODAR, show the

spatial and temporal distributions of different atmospheric properties over the mining facility, and provide the source codes to extract and perform statistical analysis of the WRF outputs.

## Chapter 2

# Weather Research and Forecasting (WRF) Model

The variants of the WRF model are known as Advanced Research WRF (ARW) and Non-hydrostatic Mesoscale Model (NMM). The ARW solver, also known as Eulerian Mass (EM) solver was developed by National Center for Atmospheric Research (NCAR) using a height and pressure based terrain-following coordinate system, while the NMM solver was expanded by National Centers for Environmental Prediction (NCEP) based on the Eta model, and later as non-hydrostatic mesoscale model [56, 57]. Finally, these versions with diverse physics packages developed at NCAR, NCEP, and elsewhere, have been included into the WRF software architecture framework [29].

### 2.1 WRF System Components

Figure 2.1 shows the main components of the WRF system. The WRF simulation is executed by using the ARW solver with other components of the WRF system including physical schemes, dynamics, initialization, and data ingestion packages. The ARW and NMM solver share most of the physical packages, while the specific compatibility varies according to the schemes [120]. The topographical information is set up through the WRF Preprocessing System (WPS) to initialize the simulation. The ingestion, reformation, and interpolation of forecast or real-time atmospheric datasets are also performed to generate lateral boundary conditions in the nested domains. Finally, the dynamic solvers and physical interface and packages execute the WRF simulation. The idealized simulations can also be performed in a simplified setting by varying various parameters and initial conditions. There are twelve

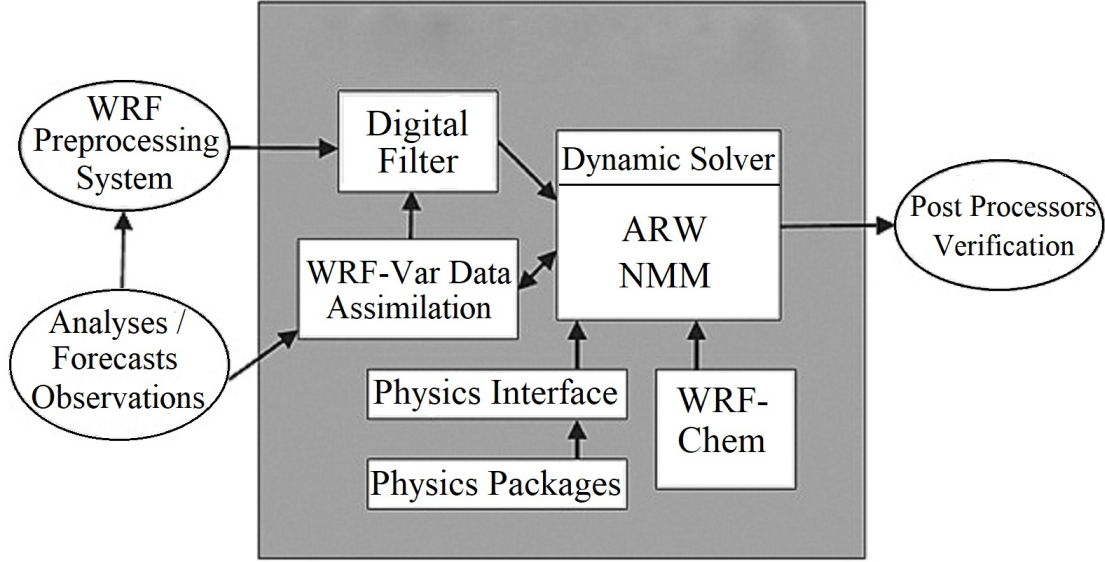


Figure 2.1: WRF system components [106].

idealized scenarios in WRF, some of which include baroclinic waves, supercell convection, Large-Eddy Simulation (LES), and tropical cyclones [97].

### 2.1.1 WRF Preprocessing System

WRF Preprocessing System (WPS) is mainly used for real-data simulations including some functions such as domain creation and terrestrial and meteorological datasets interpolation from another model to the simulation domain. The feature includes meteorological data from various datasets around the world, land use datasets based on USGS and MODIS category, map projections based on Polar Stereographic, Lambert-Conformal, Mercator, and Latitude-Longitude projections, way of nesting, and user interface to force other static data as boundary conditions in WRF [91].

### 2.1.2 WRF-Var System

WRF-Var system is combined with the WRF software structure by representing the horizontal component of background inaccuracy through looping filters and power spectra. The non-separable background error and control variables are flexible to choose, which are evaluated through the National Meteorological Center (NMC) methods of average forecast differences [106]. The WRF Data Assimilation System (WRFDA) features 3DVAR and 4DVAR

approaches as well as the hybrid variational approach. An extensive variety of direct and indirect observation types can be assimilated from the surface and higher elevation datasets to satellite-based measurements [120].

### **2.1.3 Physics and Dynamics Options**

WRF integrates various physics options that can be coupled during the simulation. The options usually vary, from simple and efficient to sophisticated and from low to high computational expense, and from advanced schemes to current operational schemes. The physics options include microphysics, radiation, cloud fraction, land use, lake, Planetary Boundary Layer (PBL), cumulus parameterization, convection, and other physical parameters. Moreover, there are several dynamics options in WRF such as diffusion, kinetic energy, damping, and advection [91].

### **2.1.4 ARW Solver**

The ARW solver applies non-hydrostatic, fully compressible Euler equations using 2nd or 3rd order Runge-Kutta schemes with smaller time steps. The velocity components  $u$  and  $v$  are based on the Cartesian coordinate system while  $w$  is the vertical velocity. The prognostic variables include 3D wind velocity, potential temperature, geopotential, and pressure of the dry air. ARW has the capability of digital filtering initialization where real data can be implemented as three dimensional or idealized data in one, two, or three dimensions. The lateral boundary conditions are periodic, open, and symmetric with free slip conditions at the bottom of the boundary. The simulations can be run by using the nudging option including one or two-way interactive nesting methods [106].

### **2.1.5 WRF-Chem**

WRF-Chem is coupled with Chemistry which simulates the biogenic and anthropogenic emissions, mixing, and chemical transformation of trace gases and aerosols simultaneously in the atmosphere. It is employed for investigating air quality, field program analysis, and cloud-scale interactions as well as for dispersion forecasts. The chemistry and dynamics are combined in WRF-Chem at every timestep that is required for exploring aerosol interactions for air quality research [16].

## 2.2 Governing Equations

The compressible and non-hydrostatic Euler equations are solved and integrated with the Advanced Research WRF (ARW) solver. It follows the ideology where the equations are formed in flux scheme using conservation of mass, momentum, energy, and species [87].

### 2.2.1 Vertical Coordinates

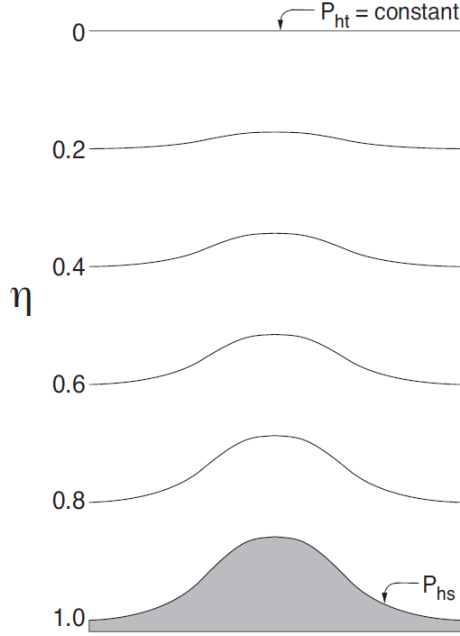


Figure 2.2: Vertical coordinate system of Advanced Research WRF (ARW) [106].

A terrain-following mass vertical coordinate system is applied to develop the ARW equations denoted by  $\eta$  and defined as

$$\eta = \frac{p_h - p_{ht}}{\mu}, \quad (2.1)$$

$$\mu = p_{hs} - p_{ht}. \quad (2.2)$$

Here,  $p_h$  denotes the hydrostatic component of the normalized pressure (unitless) while  $p_{hs}$  and  $p_{ht}$  refer to the normalized pressure (unitless) at surface level and top of the model domain, respectively [67, 106]. The vertical coordinate also known as mass vertical coordinate  $\eta$  varies from a value of 1 at the surface to 0 at the upper boundary of the model domain.



As normalized mass per unit area (unitless) within the column in the model domain at  $x$  and  $y$  coordinates are denoted by  $\mu(x, y)$ , the applicable flux form variables are

$$\mathbf{V} = \mu \mathbf{v} = (U, V, W), \quad (2.3)$$

$$\Omega = \mu \dot{\eta}, \quad (2.4)$$

$$\Theta = \mu \theta, \quad (2.5)$$

where  $\mathbf{v} = (u, v, w)$  are the covariant velocities in the two horizontal and vertical directions, respectively,  $\theta$  is the potential temperature, and  $\dot{\eta}$  is the contra-variant vertical velocity.

### 2.2.2 Euler Equations in ARW Solver

Using the variables defined above, the flux-form Euler equations can be written as follows [67, 106].

#### Equation of State

$$p = \left( \frac{R_d \theta}{p_0 \alpha} \right)^\gamma \quad (2.6)$$

Where,  $p$  is a normalized pressure using a reference pressure  $p_0$ ,  $R_d$  is the dry air gas constant,  $\gamma = 1.4$  is the heat capacity ratio for dry air, and  $\alpha = \frac{1}{\rho}$  is the inverse density.

#### Conservation of Mass

$$\frac{\partial \mu}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0 \quad (2.7)$$

Here,  $U$ ,  $V$ , and  $W$  are the mean velocity components in the  $x$ ,  $y$ , and  $z$  directions respectively.

#### Conservation of Momentum

$$\frac{\partial U}{\partial t} + \frac{\partial U}{\partial x} u + \frac{\partial V}{\partial y} u + \frac{\partial W}{\partial z} u - \frac{\partial(p\phi_\eta)}{\partial x} + \frac{\partial(p\phi_x)}{\partial \eta} = F_U \quad (2.8)$$

$$\frac{\partial V}{\partial t} + \frac{\partial U}{\partial x} v + \frac{\partial V}{\partial y} v + \frac{\partial W}{\partial z} v - \frac{\partial(p\phi_\eta)}{\partial y} + \frac{\partial(p\phi_y)}{\partial \eta} = F_V \quad (2.9)$$

$$\frac{\partial W}{\partial t} + \frac{\partial U}{\partial x} w + \frac{\partial V}{\partial y} w + \frac{\partial W}{\partial z} w - g \left( \frac{\partial p}{\partial \eta} - \mu \right) = F_W \quad (2.10)$$

## Conservation of Energy

$$\frac{\partial \Theta}{\partial t} + \frac{\partial U}{\partial x} \theta + \frac{\partial V}{\partial y} \theta + \frac{\partial W}{\partial z} \theta = F_{\Theta} \quad (2.11)$$

Here,  $F_U$ ,  $F_V$ ,  $F_W$ , and  $F_{\Theta}$  represent the forcing terms arising from model physics, turbulent mixing, spherical projections, and the earth's rotation and the non-conserved variable  $\phi = gz$  is the geopotential.

### 2.2.3 Projections in ARW Solver

Lambert conformal, polar stereographic, Mercator, and latitude-longitude are the four types of projections that are supported in the ARW solver [49]. Lambert conformal, polar stereographic, and Mercator projections are based on isotropic transformation, while latitude-longitude projection is based on anisotropic projection. The projections are implemented using map factors for both in  $x$  and  $y$  directions to accommodate the anisotropy [106]. Figure 2.3 shows the horizontal and vertical grid spacing in the ARW solver.

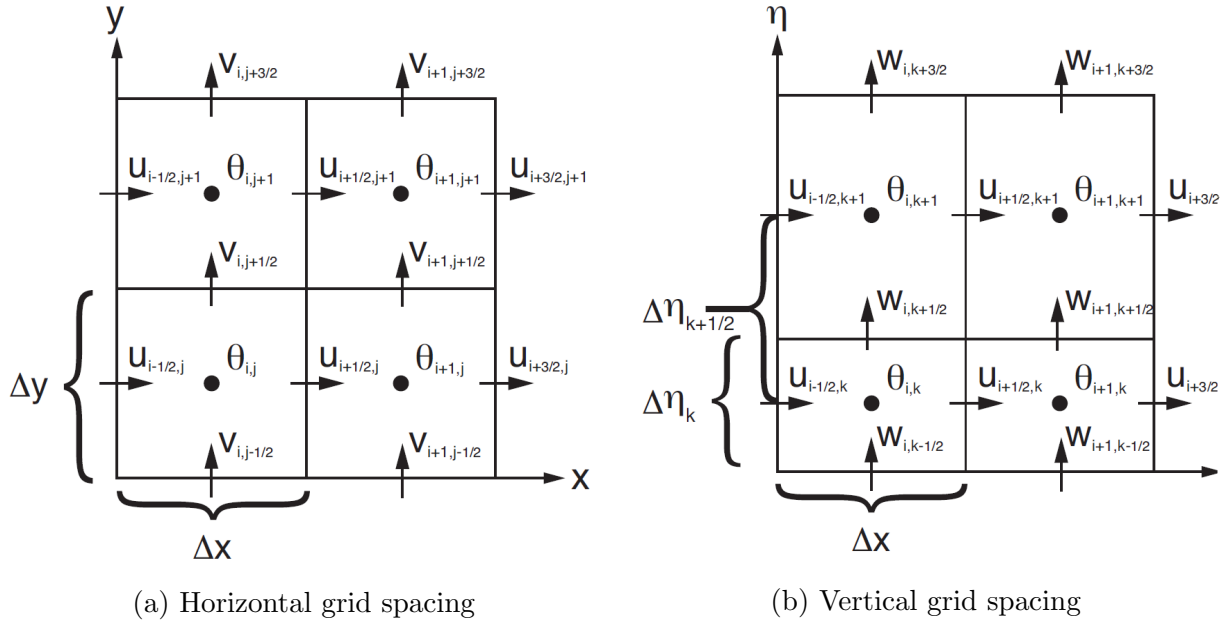


Figure 2.3: Horizontal and vertical grid spacing in ARW solver [106].

The grid spacings in  $x$  and  $y$  directions are constants during computation by the ARW solver. The transformations of the governing equations are performed by defining map scale factors  $m_x$  and  $m_y$  as

$$(m_x, m_y) = \frac{(\Delta x, \Delta y)}{D}. \quad (2.12)$$

Here  $\Delta x$  and  $\Delta y$  represent the grid spacing in  $x$  and  $y$  directions, respectively, while  $D$  represents the distance on the earth.

# Chapter 3

## WRF Simulations of Urban Climate

### 3.1 Experimental Observation

The experimental measurements were conducted in the Guelph Turfgrass Institute from July 14, 2018 to September 4, 2018. Three days, August 03, 04, and 13 in 2018, were chosen based on the availability of experimental datasets to compare with the model results. The following instruments were used to determine horizontal and vertical wind velocity, wind direction, temperature, and relative humidity in that region at surface level and higher elevations.

#### 3.1.1 SOnic Detections And Ranging (SODAR)

A Mini SODAR 4000 series by Atmospheric Systems Corporation was used to measure horizontal and vertical wind velocity and wind direction from 30 m to 200 m with an output frequency of 30 minutes.<sup>1</sup> The SODAR measures the wind properties with a vertical resolution as low as 10 m. Moreover, this SODAR can measure wind speed from 0 to 45 m s<sup>-1</sup>. It can measure wind velocity and wind direction with an accuracy of  $\pm 0.5$  m s<sup>-1</sup> and  $\pm 2^\circ$ . The SODAR consists of two components which include a signal processing unit to determine a variety of atmospheric echo intensities and a speaker detector to monitor the pulse frequencies. The signal processing unit is known as the Acoustic Signal Processor (ASP) having a number of variations including a bank of analog filters with a microprocessor [40]. As the computers are becoming more powerful, additional functions have been assigned to the ASP along with networking, diagnostic responsibilities, satellite communication, and controller features. Moreover, different speaker emitters and detectors are engaged to accomplish a

---

<sup>1</sup><http://www.minisodar.com/sodar/>

monitoring task that is most often associated with the maximum effective sampling altitude. This SODAR is perfectly suited for applications such as wind energy, wind profiling in the airport, and air pollution monitoring [27].

The SODAR works by transmitting sound waves upward into the surroundings and exposing the Doppler shift in the backscatter signal. The movement of the atmosphere is formed by the wind flow and turbulence that is generated due to mechanical and thermal forces. Thermal turbulence is due to the temperature differences while mechanical turbulence is caused by airflow over obstacles on the surface [45]. There are three beams in a SODAR to measure the 3D wind velocity. When the acoustic signal is transmitted from these beams, its energy is scattered in all directions by the turbulent refractive index that changes at all heights. Despite the fact of unique scattering patterns resulting from turbulence, some of the acoustic signals are returned towards the electric transducers [10]. This process provides a continuous-time report of echo strength associated with the turbulence intensity. The height profile of different measurements can be determined using

$$z = \frac{ct}{2}. \quad (3.1)$$

Here,  $c$  is the velocity of sound,  $t$  is the transmitting time, and  $z$  is the height of a measure of turbulence intensity [11]. As a mono-static SODAR, the SODAR system uses the same antenna for transmitting and receiving, and the scattering angle between the eddies and the SODAR antenna is  $180^\circ$  [19].

After transmitting the pulse, due to the atmospheric absorption, the echo signals become too weak so that it can not be detected properly by the electric transducers. The components of wind are estimated at this time when the acoustic pulse is transmitted in a different direction. The volume occupied by the transmitted acoustic signal increases as it develops vertically due to the shape of the conical beam. The echo from the height is obtained after calculating traveling distance and analyzing the received signal [116]. The effective volume over which the wind averaging takes place can be determined by using

$$\forall = \frac{c\tau}{2}\pi\frac{z\Delta\varphi^2}{2}. \quad (3.2)$$

Here,  $\tau$  is the acoustic pulse duration and  $\Delta\varphi$  is the beam width. The wind speed and wind direction estimate a vertical spatial resolution having a vertical extent determined by  $\frac{c\tau}{2}$  and a horizontal spatial resolution around  $\frac{z\Delta\varphi}{2}$  [11]. The scattered acoustic power interpreted by the SODAR is given by

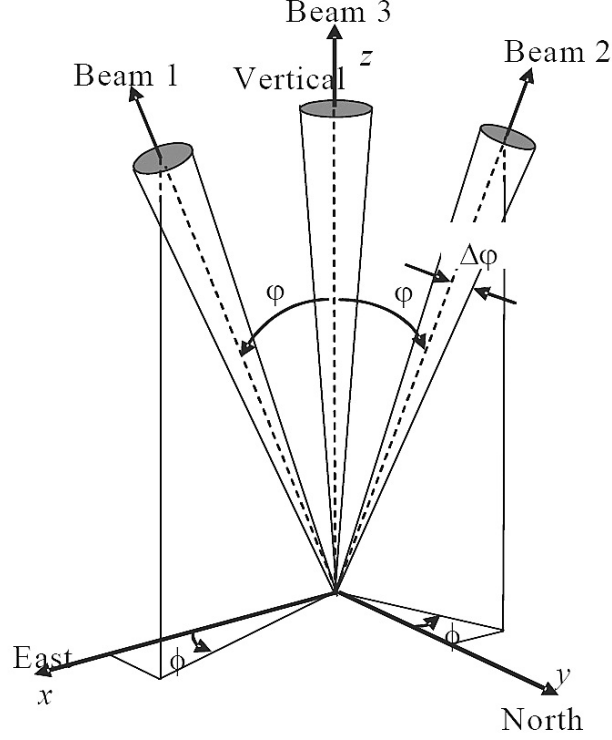


Figure 3.1: Working principle of three beam SODAR [10].

$$\sigma_s = \frac{1}{8} \frac{k^4}{\kappa^{\frac{11}{3}}} \cos^2 \varphi \left[ 0.033 \frac{C_T^2}{T^2} + 0.76 \cos^2 \left( \frac{\varphi}{2} \right) \frac{C_V^2}{\pi c^2} \right], \quad (3.3)$$

where  $T$  is the ambient temperature,  $k$  is the wave number, and  $C_T^2$  and  $C_V^2$  are the temperature and velocity structure functions respectively [12]. The transducers create an almost spherical wave through the antennas to measure the wind vector. Due to sound interference between the distinct speakers, the sample has a tendency to have narrower beams of higher power [34]. This acoustic profiler transmits and receives sound waves within a specific frequency band. Since the return signal strength is inversely proportional with the height, the weaker signals from greater heights are simply lost due to turbulence and other background noise. Hence, it is crucial to identify the potential noise sources as well as to measure the noise level in order to reduce the bias of the SODAR system [42].

The content in the reflected signal includes background acoustic noise, Doppler-shifted echo signals at frequencies close to the emitted frequency  $f_T$ , and echoes from close-by solid objects in the surrounding at the frequency  $f_T$ . The received signal is derived by a Doppler Fast Fourier Transform (FFT) spectrum at discrete frequencies

$$f_i = i \frac{f_s}{N_f}, \quad (3.4)$$

$$i = -\frac{N_f}{2} + 1, -\frac{N_f}{2} + 2, \dots, \frac{N_f}{2} - 1, \frac{N_f}{2}, \quad (3.5)$$

where  $N_f$  is the number of points in the spectrum and  $f_s$  is the sampling frequency [11]. Finally, the SODAR is calibrated to ensure appropriate calculation of wind and turbulence properties [94]. The sensible wind profiles are simulated by transforming the accurate SODAR frequency and amplitude through an atmospheric scattering model. The transmitted pulses are analyzed by an algorithm which calculates the pseudo-atmospheric reaction in real time [17].

### 3.1.2 Weather Station

Guelph Turfgrass weather station hourly datasets were used to observe the temperature and relative humidity at 2 m. The Guelph Turfgrass Institute station that is identified with the World Meteorological Organization (WMO) identification number 71833. This data is accessible with an hourly temporal resolution. Temperature data is measured at 2 m while wind data is measured at 10 m above the ground [5].

## 3.2 WRF Model

A particular distribution of WRF is used in this study, titled Unified Environmental Modelling System (UEMS) version 18.1.1.<sup>2</sup> The installation, configuration, and execution of UEMS has been immensely facilitated to encourage its use within operational, private, and university forecasting and research communities. The UEMS allows for the acquisition of multiple initialization datasets via the Non-hydrostatic Forecast System (NFS) and Hyper Text Transfer Protocol (HTTP). The system is semi-intelligent in that it can determine which datasets are available for ingestion at a given time [106].

### 3.2.1 Domain Configurations

The sensitivity tests of WRF have been conducted with different grid configurations including the way of nesting, grid resolution, time-invariant data resolution and geographical shifts

---

<sup>2</sup><http://strc.comet.ucar.edu/software/uems/>

in the domain centre. The minimum grid resolutions of the nested domains are at least four grid cells from the boundary of parent domain [120]. For computational stability, the magnitude of timesteps (in seconds) is suggested a maximum 6 times the magnitude of the coarsest grid distance [106]. Figure 3.2 and Table 3.1 illustrate the size and configuration of the domains, respectively, where domain 1 is the largest domain and domain 5 is the smallest one. The domains consist of five nested levels. The numerical grid resolutions in the horizontal directions also increase from domain 1 to domain 5.

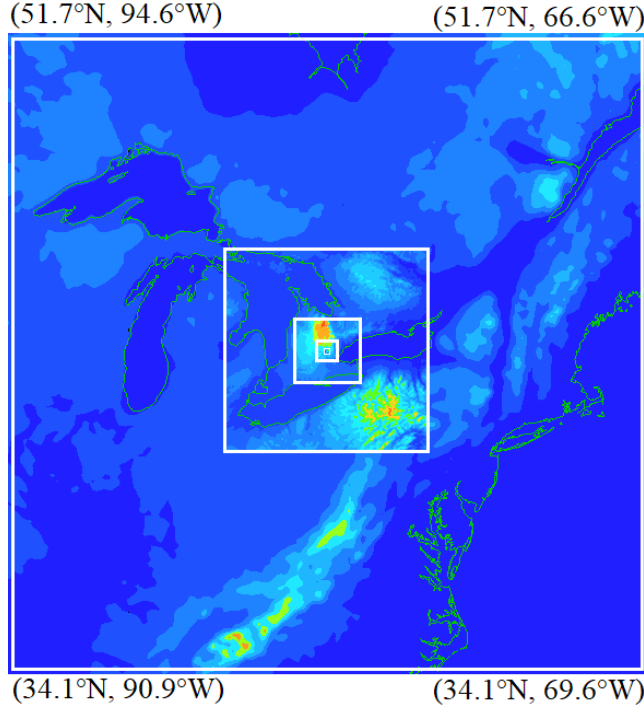


Figure 3.2: WRF simulation domains with latitude and longitude centred at Guelph, Ontario, Canada.

### 3.2.2 Initialization

The WRF simulations are run for three different days on August 3, 4 and 13 in 2018, for 36 hours each including a spin-up time of 12 hours. The Global Forecasting System (GFS) dataset has been used in the simulation that provides initial and time-varying boundary conditions to WRF at 3 hours time resolution and  $0.5^\circ$  spatial resolution [39, 106, 126].<sup>3</sup>

<sup>3</sup><https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>



Table 3.1: Domain configurations and associated parameters.

Parameter	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5
Domain size (km $\times$ km)	2000	632.7	197.6	61.4	18.5
Horizontal grid resolution (km)	10	3.33	1.11	0.37	0.12
Number of grid elements	200	190	178	166	154
Top of the domain (km)	25	25	25	25	25
Vertical levels	45	45	45	45	45
Output frequency (min)	60	60	60	60	60
GTPO 30s topography resolution (m)	900	900	900	900	900
SRTM 1s topography resolution (m)	-	-	-	30	30

In WRF, the time-varying meteorological initialization fields provide data at different topographic resolutions.

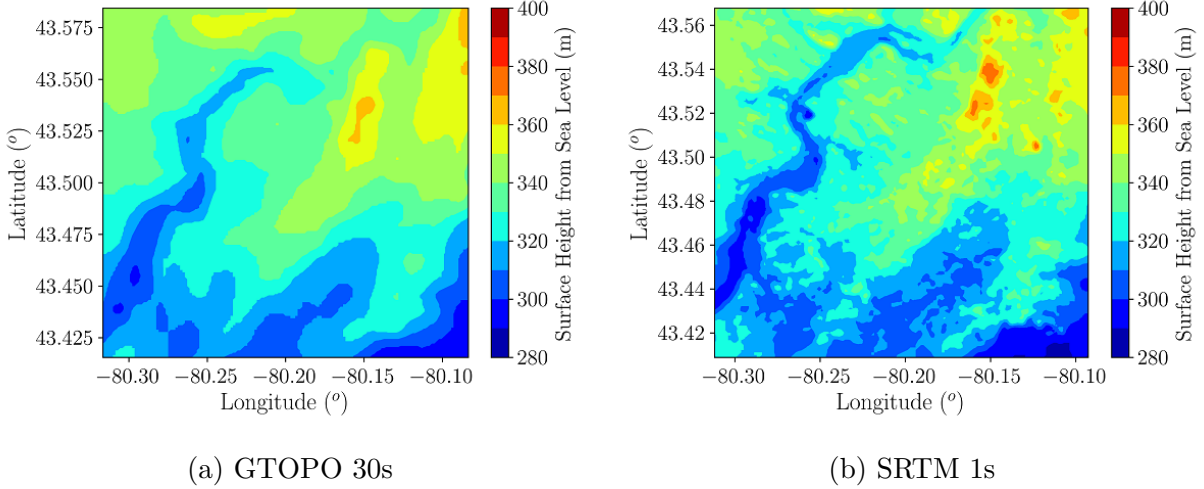


Figure 3.3: Surface height from sea level: GTPO 30s has low resolution; SRTM 1s has high resolutions.

The Global 30 Arc-Second (GTPO 30s) and the Shuttle Radar Topography Mission 1 Arc-Second (SRTM 1s) datasets are used to modify the resolution of topography in domains 4 and 5. GTPO 30s is a global dataset which covers the full extent of latitude from 90 degrees south to 90 degrees north, as well as from 180 degrees west to 180 degrees east. As a low topographic resolution initialization, the GTPO 30s has a topographic resolution around 900 m  $\times$  900 m. On the other hand, the Shuttle Radar Topography Mission (SRTM) is obtained through digital elevation models on a near-global scale to generate a high-resolution

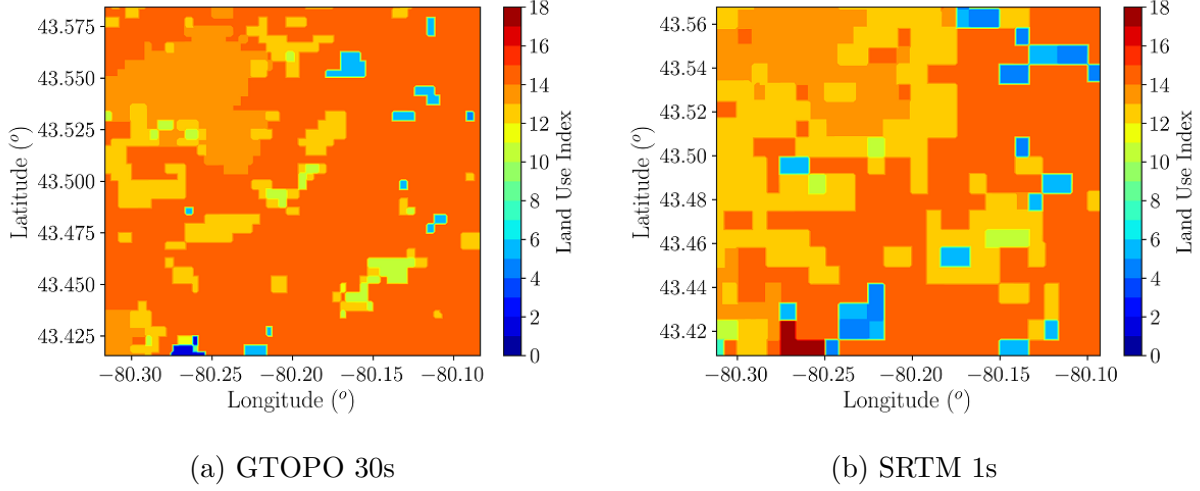


Figure 3.4: Land use classification of GTOPO 30s and SRTM 1s.

digital topographic database of the earth. The SRTM 1s dataset has a topographic resolution of about  $30 \text{ m} \times 30 \text{ m}$ , which is 30 times higher than the GTOPO 30s dataset [38, 72]. Moreover, time-invariant data such as land water masks, vegetation, land use, and albedo have been obtained from the NCAR and MODIS databases. The surface height of domain 5 from sea level and land use classifications for both GTOPO 30 s and SRTM 1s datasets are shown in Figures 3.3 and 3.4, respectively.

### 3.2.3 Planetary Boundary Layer Parameterization

The planetary boundary layer (PBL) scheme implemented in the model plays a decisive role on the accuracy of reanalyzed state and affects the flow within the PBL as the wind varies according to the stability of the atmosphere. Researchers have studied, performing a sensitivity test of the WRF model, and found that the Yonsei University (YSU) scheme shows improvement over the other PBL schemes of WRF [26]. Hence, YSU scheme is applied in the simulations, which uses identical profile functions for momentum and heat assuming turbulent Prandtl number as a constant [24].

### 3.2.4 Model Evaluation

The results of the WRF model should be verified because of the uncertainty in model implementation for any specific location. Hence, experimental datasets from Mini SODAR 4000

series located in the Turfgrass area at Guelph are used to evaluate the WRF simulations in higher elevation wind for every 4 hours while Guelph Turfgrass weather station's hourly dataset is used for the surface level comparison. A quantitative comparison between the experimental observations and WRF model is performed by determining the Fractional Bias (FB) and the Normalized Mean Square Error (NMSE) defined by [4]

$$FB = \frac{\sum_{i=1}^n O_i - \sum_{i=1}^n M_i}{0.5(\sum_{i=1}^n O_i + \sum_{i=1}^n M_i)}, \quad (3.6)$$

$$NMSE = \frac{\sum_{i=1}^n (O_i - M_i)^2}{(\sum_{i=1}^n O_i)(\sum_{i=1}^n M_i)}. \quad (3.7)$$

Here,  $O$  and  $M$  represent the median of experimental observations and WRF outputs at every hour or every four hours, respectively, while  $n$  represents the number of samples. Moreover, as the wind direction is a circular variable, deviations in wind direction as predicted by WRF and the observations are reported as the absolute bias or difference between simulations and observations. For each comparison between the observation ( $O$ ) and the model ( $M$ ), the bias is calculated using

$$\Delta\phi = |O - M| \% 180. \quad (3.8)$$

Here all angles are in degrees and the remainder of a positive difference is taken by dividing this difference with  $180^\circ$  to ensure that the difference is always between 0 and  $180^\circ$  [82]. However, for multiple comparisons, an average value is taken for the wind bias.

### 3.3 Results and Discussion

The comparison between WRF simulations in domain 5 and the experimental observations on August 3, 4 and 13, 2018 is evaluated in this discussion. In each time interval, the median value of every property is considered to perform statistical analysis because the mean value may not be a fair representation of the data due to the mean value being easily influenced by the outliers in the data.

#### 3.3.1 Horizontal Wind Velocity

Horizontal wind speed is a fundamental meteorological property caused by air moving from high to low pressures. Figure 3.5 shows the fractional bias of horizontal wind speed at 10

m (near surface level), 70 m (Mini SODAR) and 170 m (Mini SODAR). These altitudes are pressure heights.<sup>4</sup> In these figures, the error bars represent the Normalized Mean Square Error (NMSE).

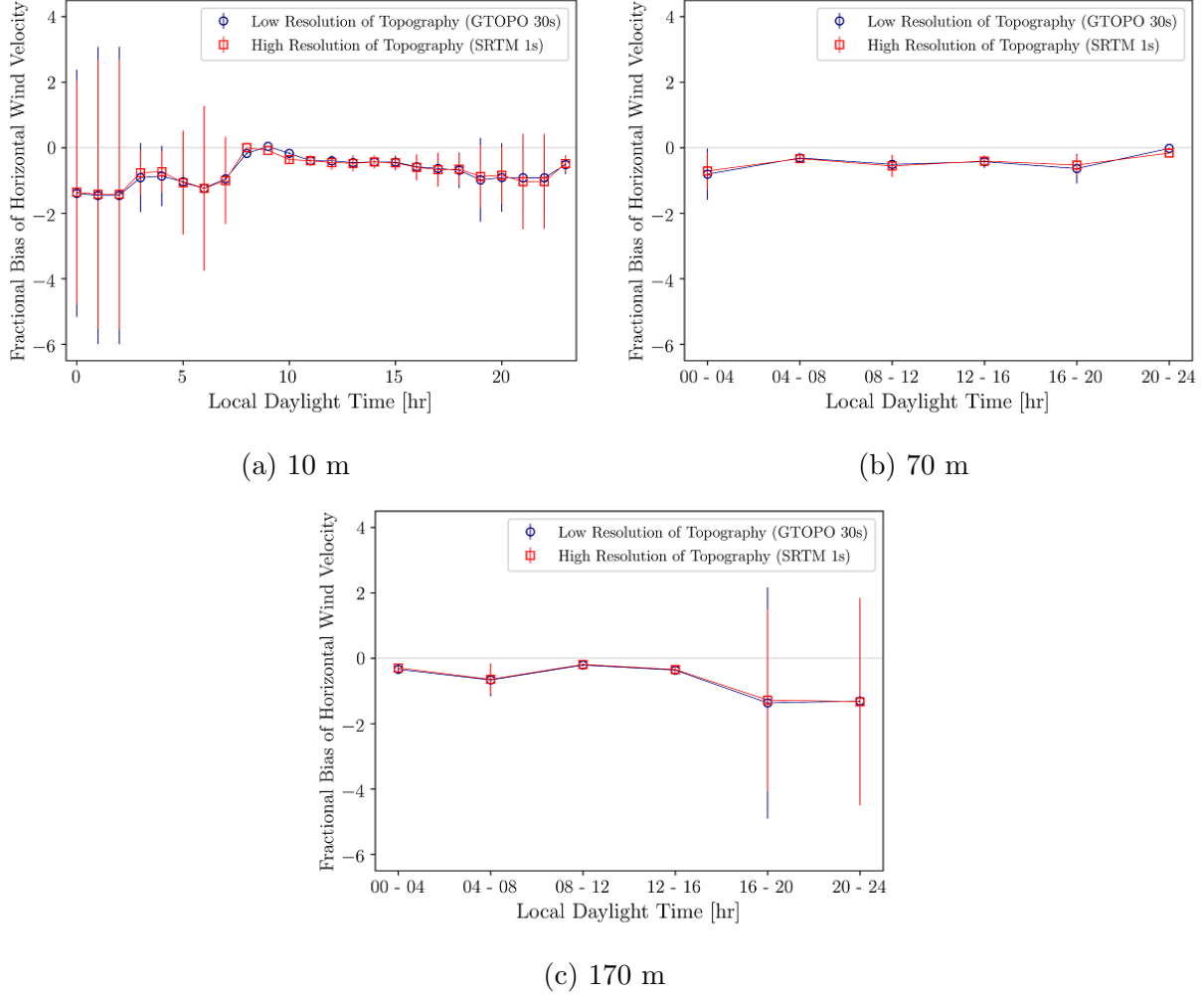


Figure 3.5: Effect of topography on horizontal wind velocity at 10 m, 70 m, and 170 m; The error bars represent the Normalized Mean Square Error (NMSE).

It is observed that in convective hours, when the atmosphere is thermally unstable, WRF shows less fractional bias and error than the thermally stable conditions at late night and early morning. There are no significant differences between low and high topographic resolution WRF simulations although at 10 m elevation high topographic resolution WRF

<sup>4</sup>[www.weather.gov/epz/wxcalc.pressurealtitude](http://www.weather.gov/epz/wxcalc.pressurealtitude)

simulations sometimes show marginally better agreement than the low topographic resolution WRF simulations. Low altitude wind speeds at surface level are influenced by surface roughness and atmospheric thermal stability [25]. Low topographic resolution WRF simulations do not account for realistic surface roughness in the topography and therefore exhibit unwanted bias in wind speed compared to high topographic resolution WRF simulations.

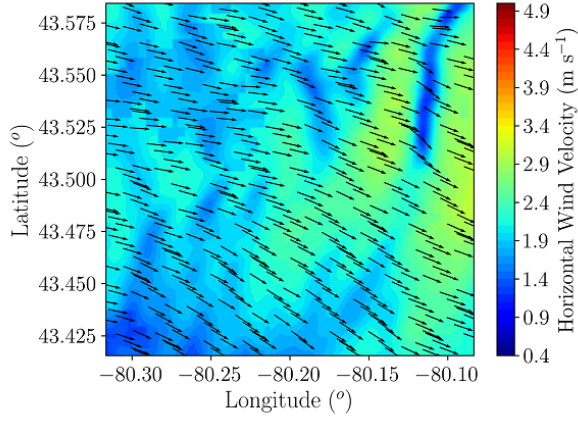
Figures 3.6 and 3.7 show the horizontal wind velocity in domain 5 at 10 m for different types of topography on August 4, 2018 during thermally stable and unstable conditions. From this study, it is found that there are some differences in horizontal wind velocity and wind direction at 0200 and 0600 LDT although at 1400 and 1800 LDT these properties are almost the same both for GTOPO and SRTM WRF simulations. GTOPO WRF simulations exhibit higher wind velocity than STRM cases in thermally stable condition. Moreover, the wind direction also changes from east to south-east in GTOPO WRF cases while the wind direction is towards the east side at most of the times in SRTM WRF simulations during thermally stable hours.

The horizontal wind velocity profile is shown in Figure 3.8 both for GTOPO and SRTM WRF simulations. In support of surface-level contour plots of wind speed, when the atmosphere is stable the wind velocity is lower during late night and early morning. On the other hand, the horizontal velocity at lower altitudes becomes higher in convective hours due to the warming of air when the atmosphere is thermally unstable.

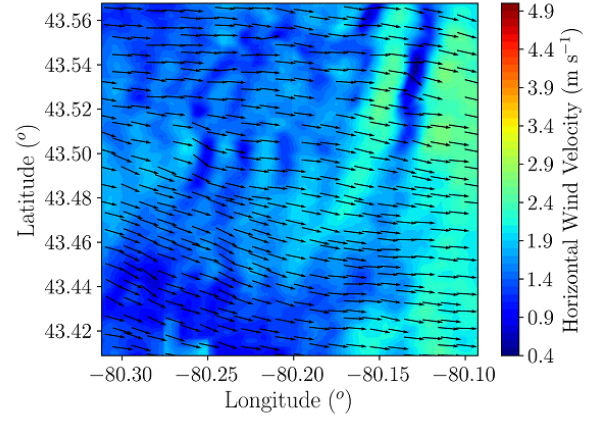
### 3.3.2 Vertical Wind Velocity

Vertical wind velocity is a key factor for cloud development, precipitation, and development of weather systems [86]. Figure 3.9 shows the fractional bias of vertical wind speed at 70 m (Mini SODAR) and 170 m (Mini SODAR) respectively. It is also noted that the 10 m observation does not include vertical wind speed. In these figures, the error bars represent the Normalized Mean Square Error (NMSE).

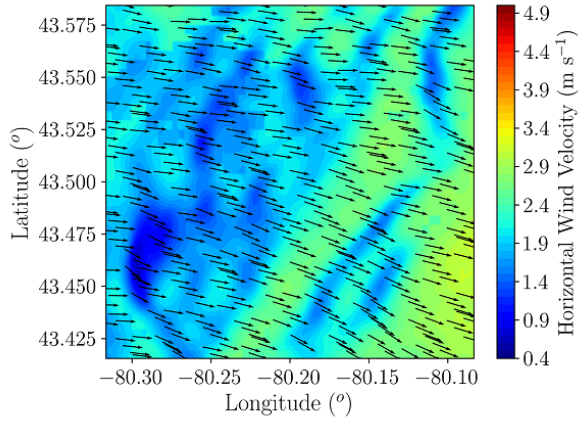
The present results of the vertical velocity profile show better agreements with Mini SODAR at thermally unstable condition compared to the stable condition. However, there are large differences between low and high topographic resolution WRF simulations at stable conditions and with low topographic resolution WRF simulations resulting in more bias and error. Nevertheless, it should be borne in mind that, as a sonic instrument, the Mini SODAR has limitations in measuring profiles of vertical wind accurately in higher altitudes which may affect the experimental measurements.



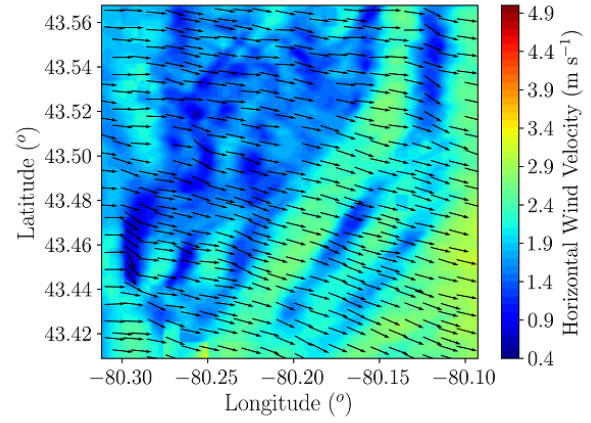
(a) GTOPO: 0200 LDT



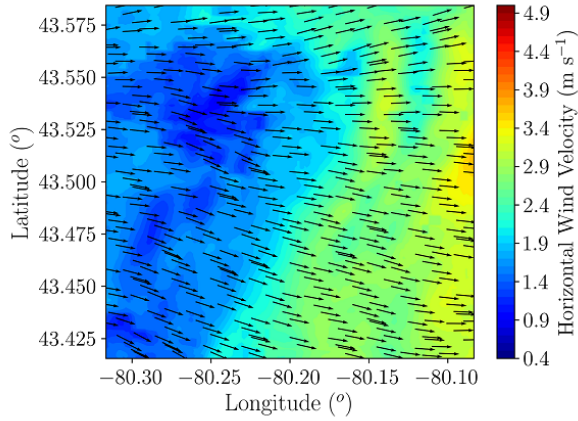
(b) SRTM: 0200 LDT



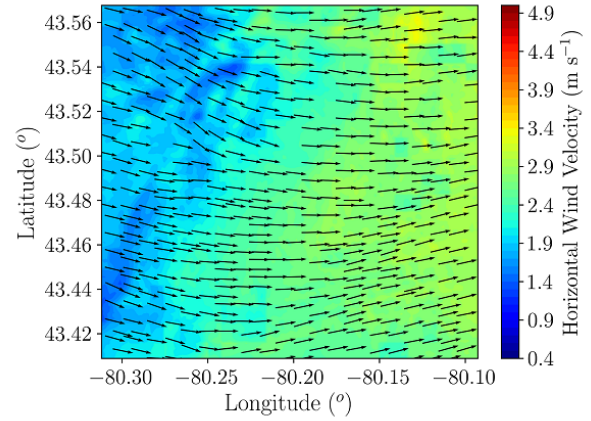
(c) GTOPO: 0600 LDT



(d) SRTM: 0600 LDT



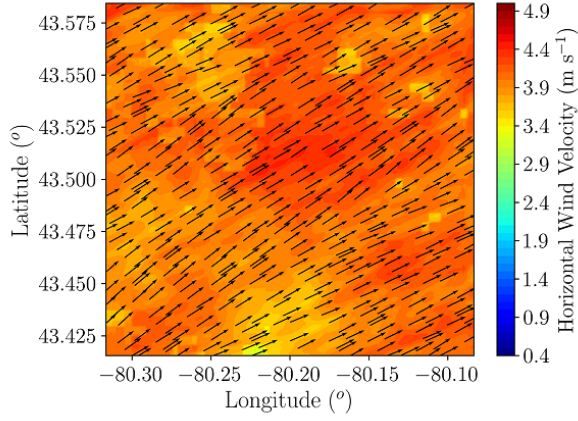
(e) GTOPO: 1000 LDT



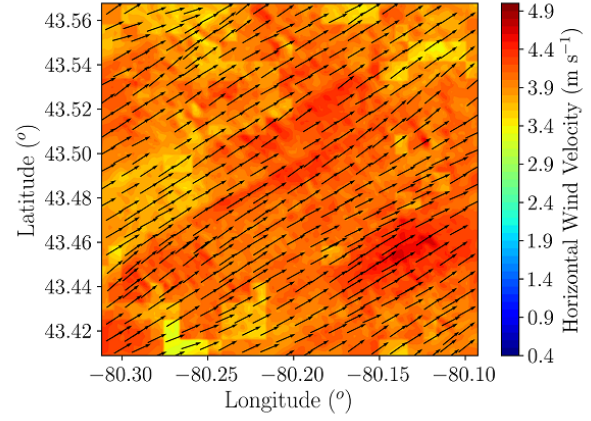
(f) SRTM: 1000 LDT

Figure 3.6: Effect of topography on 10-m horizontal wind velocity and direction at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT).

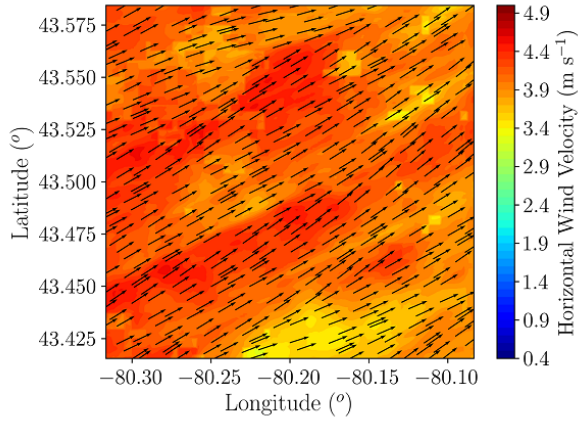




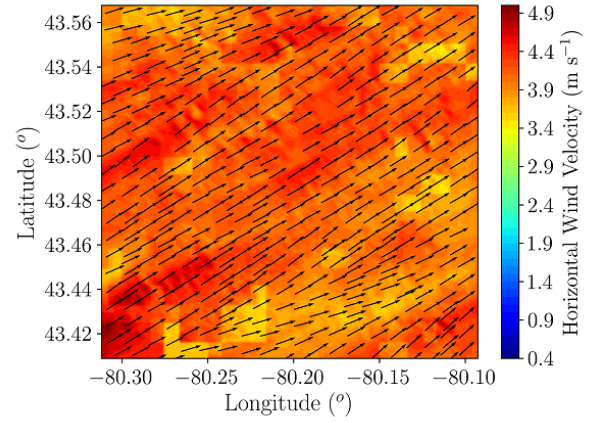
(a) GTOPO: 1400 LDT



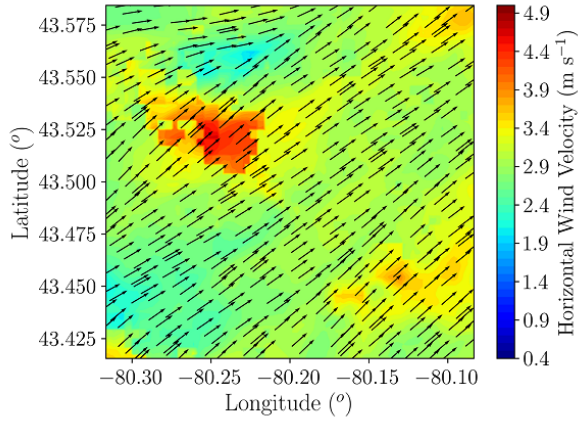
(b) SRTM: 1400 LDT



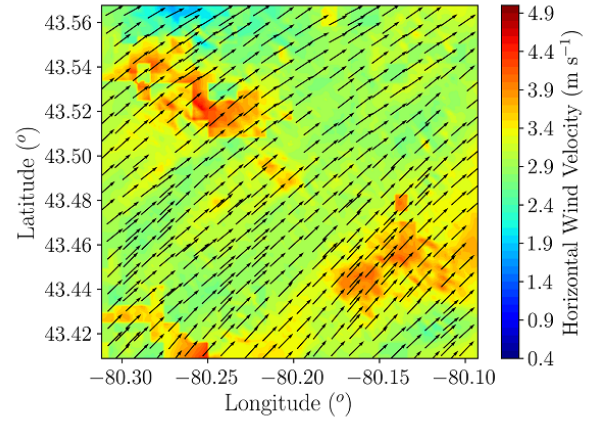
(c) GTOPO: 1800 LDT



(d) SRTM: 1800 LDT

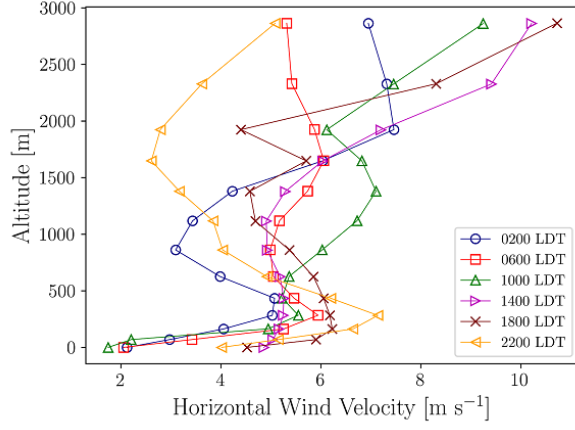


(e) GTOPO: 2200 LDT

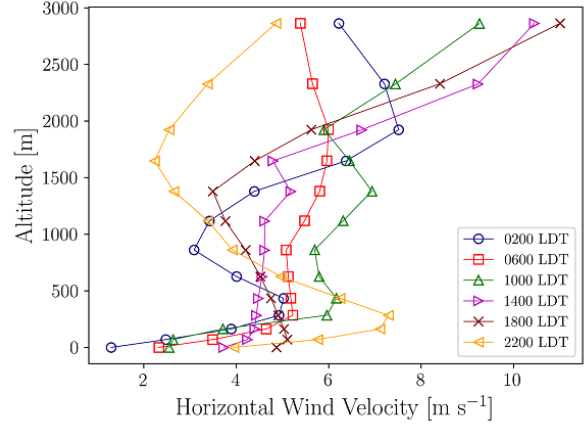


(f) SRTM: 2200 LDT

Figure 3.7: Effect of topography on 10-m horizontal wind velocity magnitude and direction at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT).

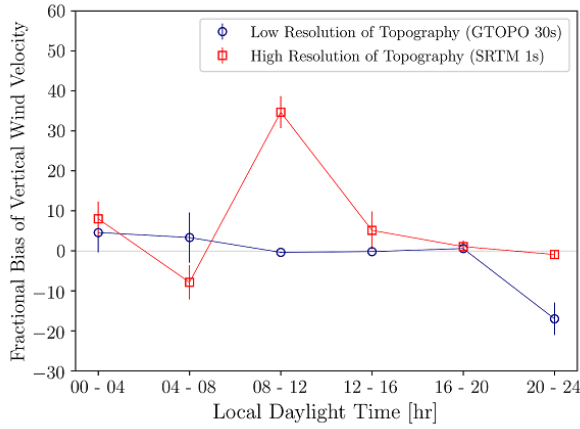


(a) GTOPO

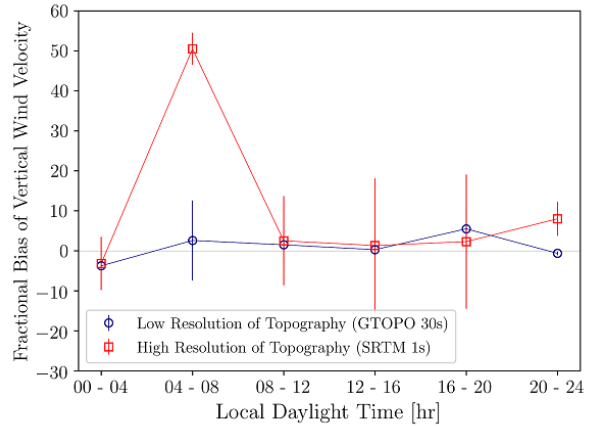


(b) SRTM

Figure 3.8: Horizontal wind velocity profile based on pressure height on August 4, 2018.



(a) 70 m



(b) 170 m

Figure 3.9: Effect of topography on vertical wind velocity at 70 m and 170 m; The error bars represent the Normalized Mean Square Error (NMSE).



### 3.3.3 Wind Direction

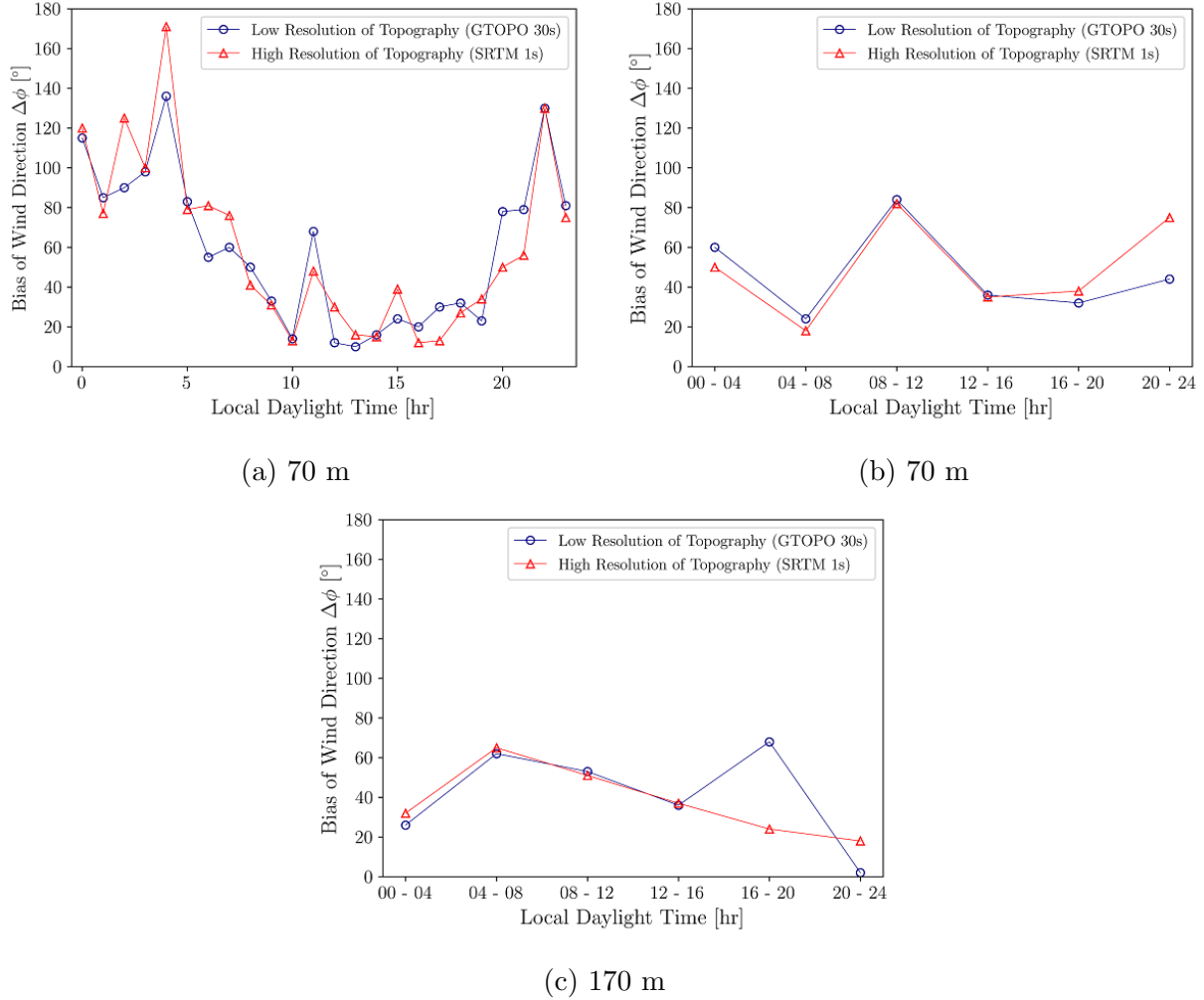


Figure 3.10: Effect of topography on wind direction at 70 m and 170 m showing the median values of wind direction bias for each hour.

The median values of wind direction bias at 10 m (near surface level), 70 m (Mini SODAR) and 170 m (Mini SODAR) are shown in Figure 3.10. It has been identified that there is an extensive bias of wind direction in thermally stable conditions while the bias is less in the unstable condition. In stable condition, the wind flows very slowly which causes more complexity to determine the exact wind direction for a specific time. Moreover, the bias of wind direction is more near the surface level than the higher altitudes. This is due to the surface roughness, and shear stress which results in frequent changes of wind

directions. However, there is no significant difference in wind direction between low and high topographic resolution WRF simulations during most of the times.

### 3.3.4 Potential Temperature

Potential temperature is an important parameter which is not affected by the physical lifting or sinking associated with the flow over solid obstacles or turbulence in the atmosphere [80].

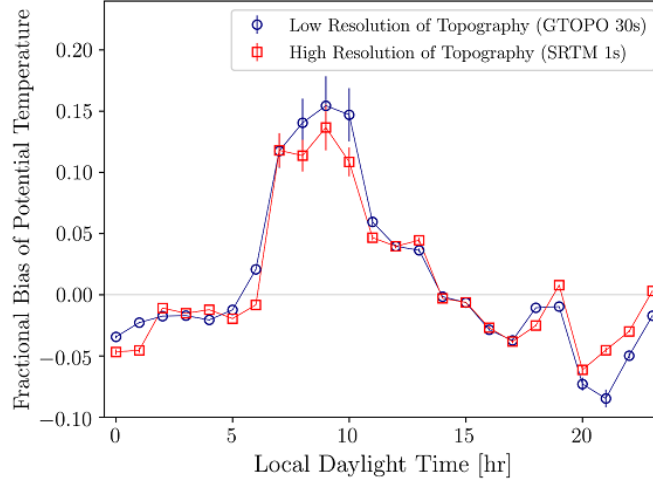
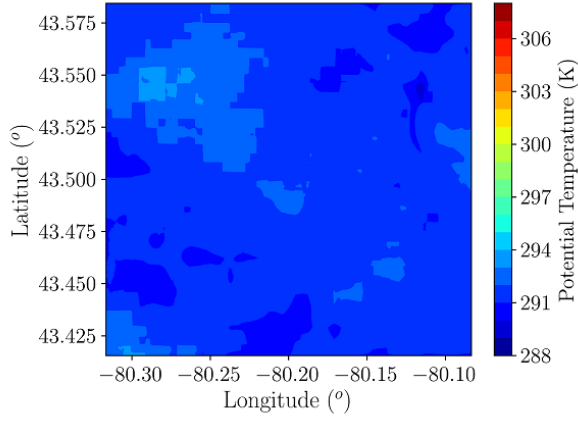


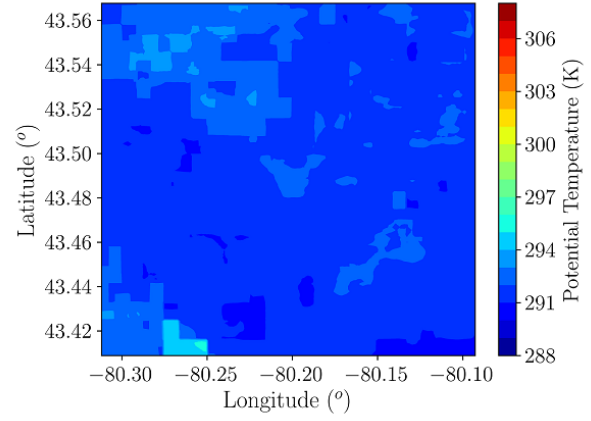
Figure 3.11: Fractional bias of potential temperature at 2 m; The error bars represent the Normalized Mean Square Error (NMSE).

Figure 3.11 shows the fractional bias of potential temperature at 2 m (near surface level) in domain 5. In terms of predicting the potential temperature, WRF shows better agreements at the stable condition. Moreover, it has been observed that the bias is negative in the stable condition which means WRF overpredicts potential temperature during this time. In contrast, WRF underestimates the potential temperature at the unstable condition as the fractional bias shows positive trend during convective hours. However, the bias is almost the same in most of the times for low and high topographic resolution WRF outputs.

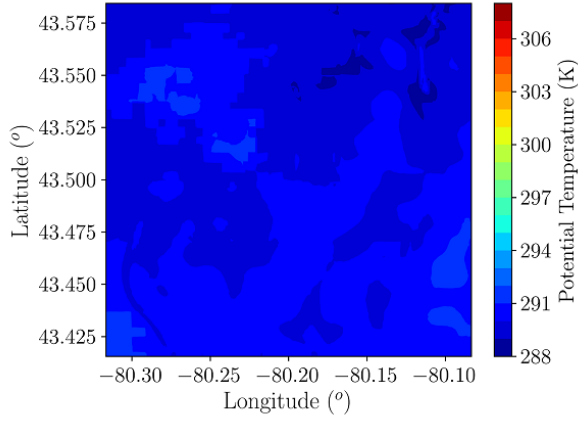
Figures 3.12 to 3.14 illustrate the surface level potential temperature contour plots and potential temperature profiles both in stable and unstable conditions. The spatial distribution of potential temperature shows similar agreements at the stable conditions although there are more deviations of potential temperature between low and high topographic resolutions for the unstable condition. It appears that the surface level temperature is better well-mixed for high topographical resolution during convective hours. This can explain the



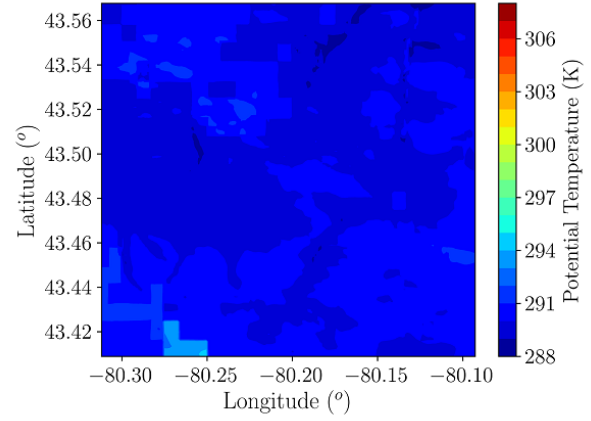
(a) GTOPO: 0200 LDT



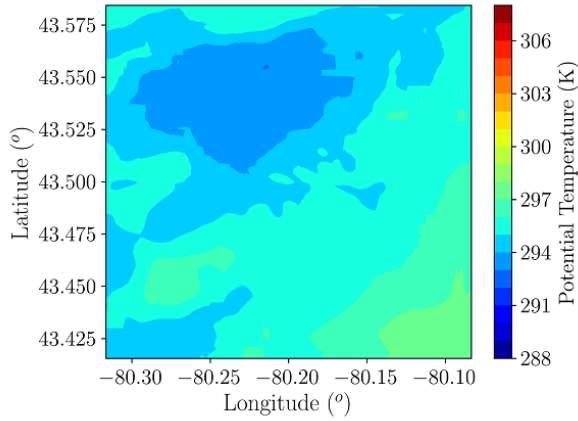
(b) SRTM: 0200 LDT



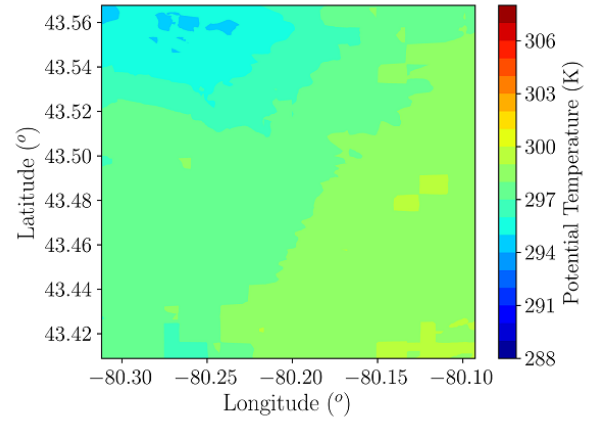
(c) GTOPO: 0600 LDT



(d) SRTM: 0600 LDT

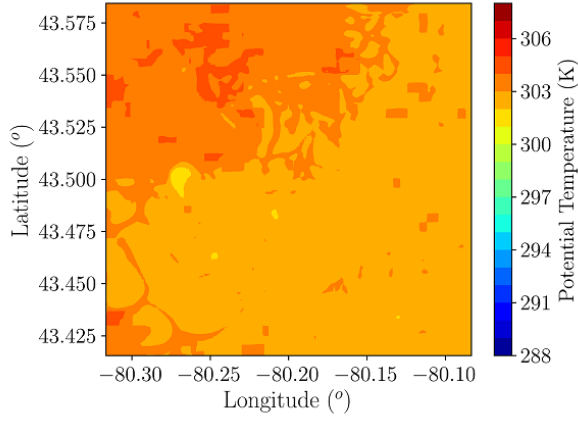


(e) GTOPO: 1000 LDT

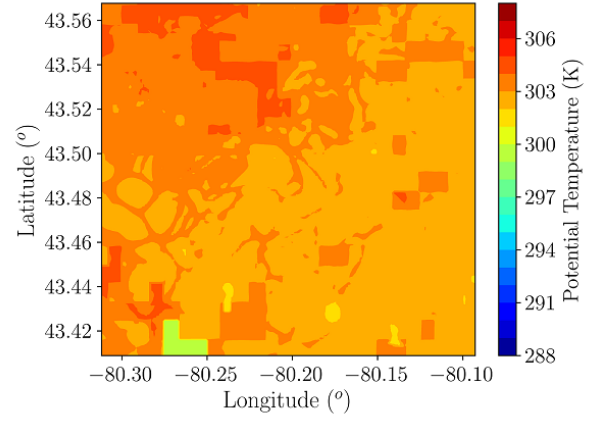


(f) SRTM: 1000 LDT

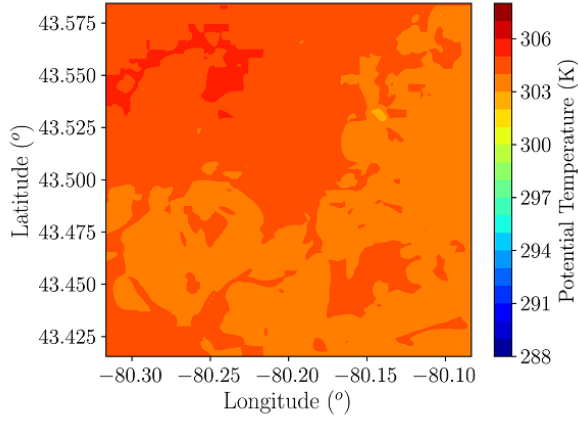
Figure 3.12: Effect of topography on 2-m potential temperature at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT).



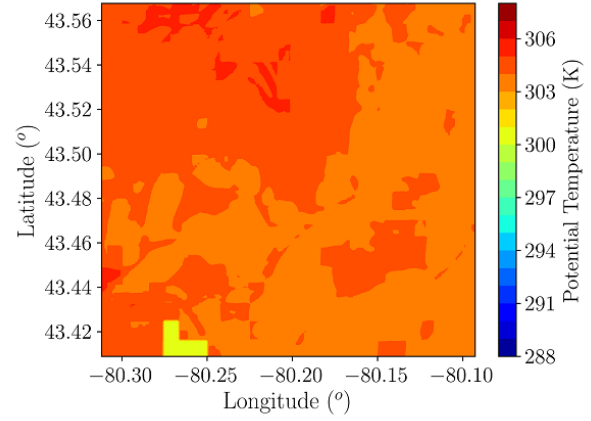
(a) GTOPO: 1400 LDT



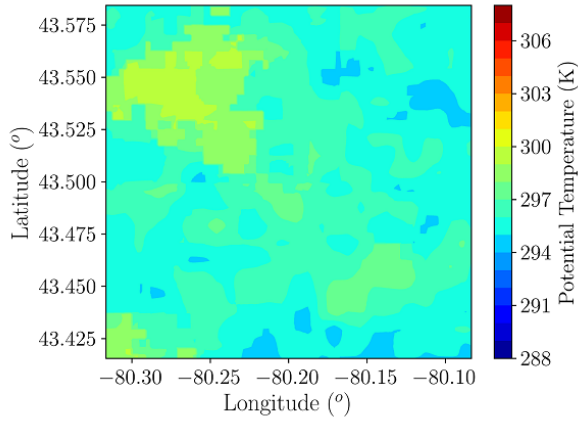
(b) SRTM: 1400 LDT



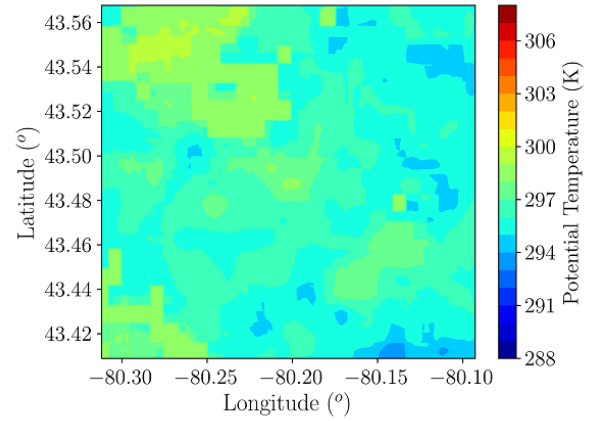
(c) GTOPO: 1800 LDT



(d) SRTM: 1800 LDT



(e) GTOPO: 2200 LDT



(f) SRTM: 2200 LDT

Figure 3.13: Effect of topography on 2-m potential temperature at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT).

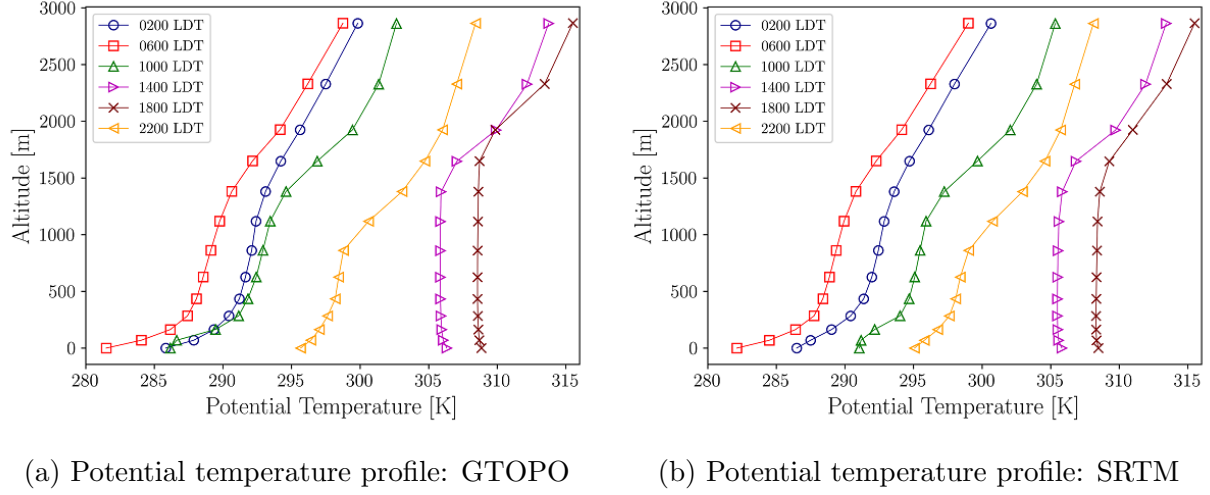


Figure 3.14: Potential temperature profile based on pressure height on August 4, 2018.

difference in the results during convective hours because surface roughness variations can possibly be responsible for a higher degree of surface-level mixing. In the potential temperature profile, when the atmosphere is stable at night and early morning, there is positive potential temperature gradient while the potential temperature gradient is negative at lower altitudes in thermally unstable condition especially during the afternoon hours.

### 3.3.5 Relative Humidity

Besides water content, relative humidity depends on temperature and the pressure of any region. The same amount of water vapor results in higher relative humidity in cool air than warm air.

Figures 3.15 to 3.17 show the fractional bias of relative humidity at 2 m (near surface level) and the spatial distributions of relative humidity at stable and unstable conditions in domain 5 for low and high topographic resolutions. WRF shows a negative correlation for the fractional bias of relative humidity and temperature. WRF underpredicts the relative humidity at the stable condition as the bias is positive while it overestimates the relative humidity at the unstable condition. However, there is no significant difference of relative humidity between low and high topographic resolution WRF outputs in most of the times although at 1000 LDT the humidity of SRTM WRF simulation is comparatively lower than the GTOPO WRF simulation.

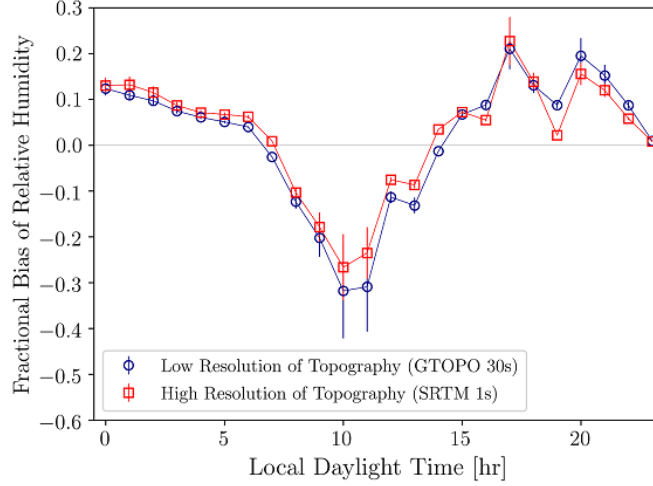


Figure 3.15: Fractional bias of relative humidity at 2 m. The error bars represent the Normalized Mean Square Error (NMSE).

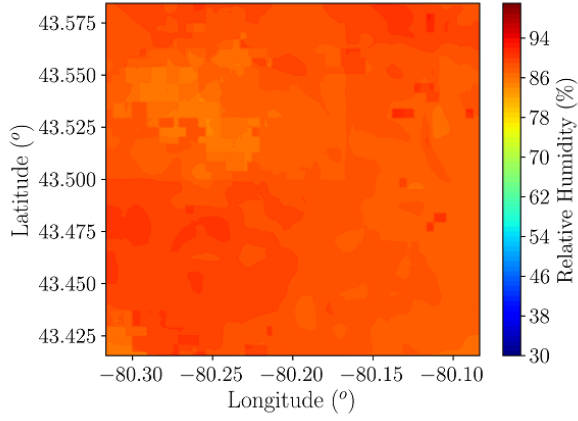
### 3.3.6 Sensible Vertical Turbulent Heat Flux

The turbulent vertical heat flux involves the eddy-induced fluxes of heat. At the surface level, the small-scale turbulence determines the vertical heat flux, while the large-scale turbulence heat flux is zero. With the increase of height, the large-scale turbulence heat flux increases and reaches a maximum value [90].

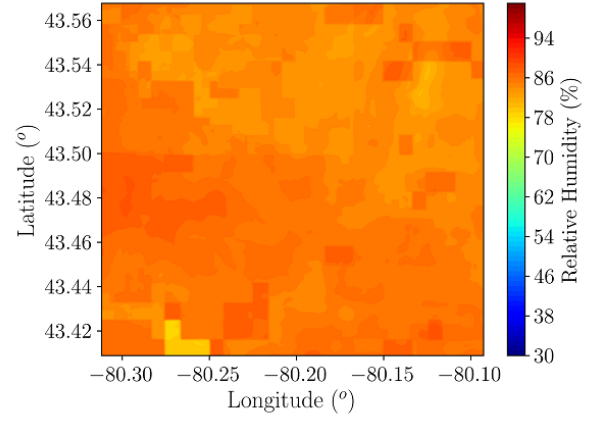
WRF shows fewer variations of vertical turbulent heat flux in both low and high topographic resolution simulations in thermally stable condition. However, the differences are more under the thermally unstable condition, when the heat flux changes rapidly with time. From Figures 3.18 and 3.19, it is observed that high topographic resolution WRF simulations predict higher heat flux than the low topographic resolution WRF simulations. Moreover, in support of the potential temperature spatial distribution contours, the heat flux is also higher in SRTM WRF simulation at 1000 LDT similar to the potential temperature although at thermally stable hours there are no significant differences.

## 3.4 Error Analysis

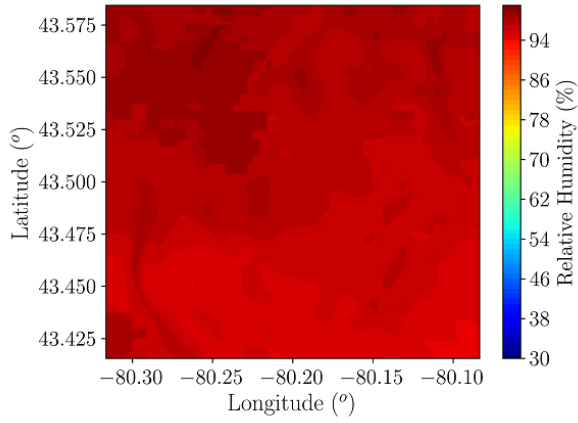
The experimental data from the Mini SODAR and Guelph Turfgrass weather station are used to evaluate the WRF simulations for those specific locations in domain 5. A quantitative comparison between the experimental observations and WRF model is also performed by



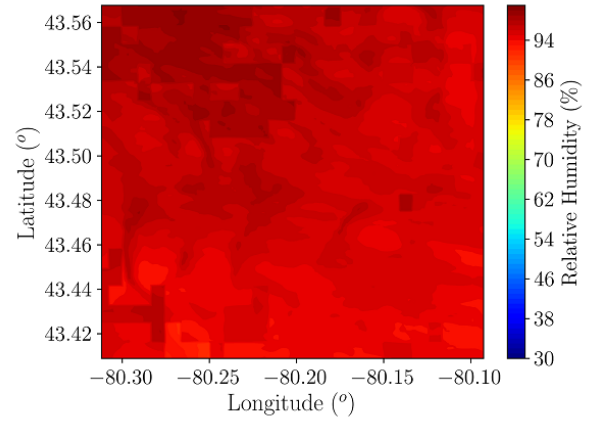
(a) GTOPO: 0200 LDT



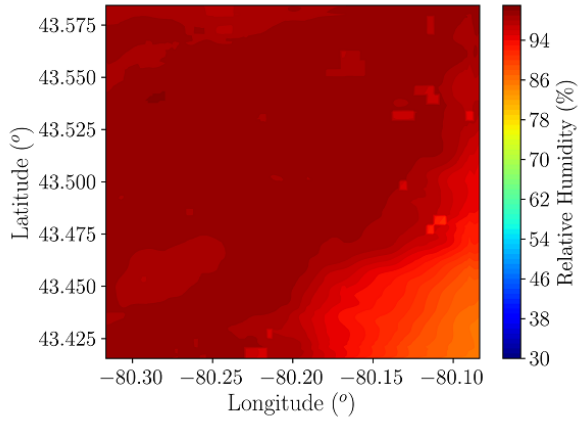
(b) SRTM: 0200 LDT



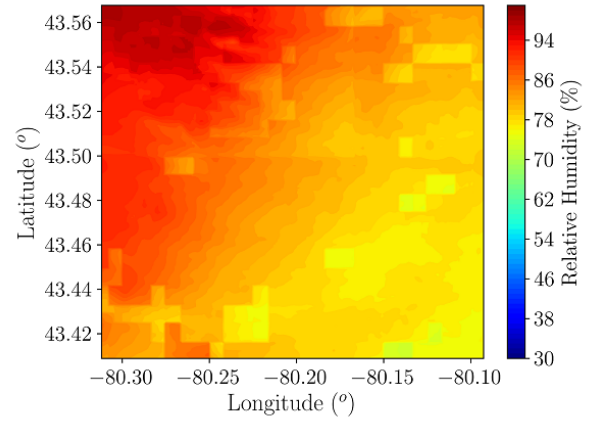
(c) GTOPO: 0600 LDT



(d) SRTM: 0600 LDT

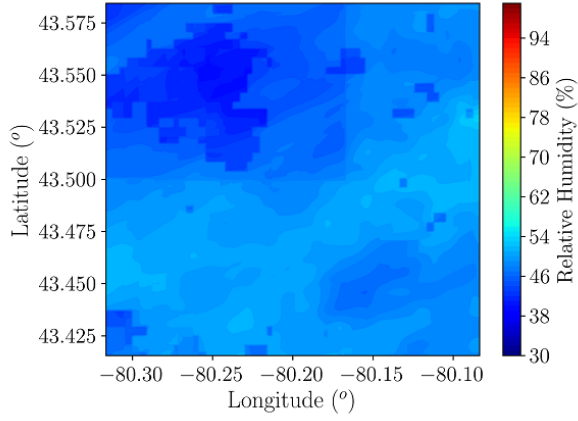


(e) GTOPO: 1000 LDT

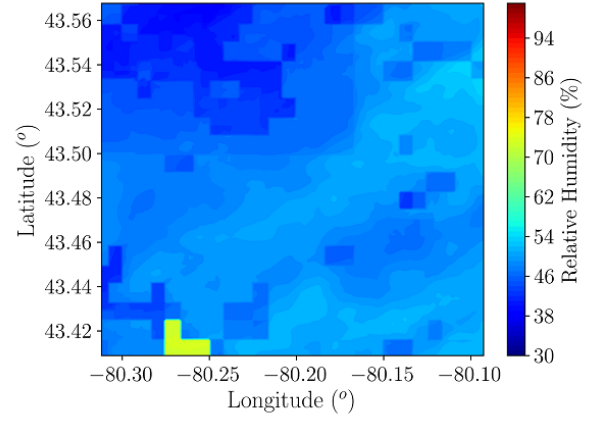


(f) SRTM: 1000 LDT

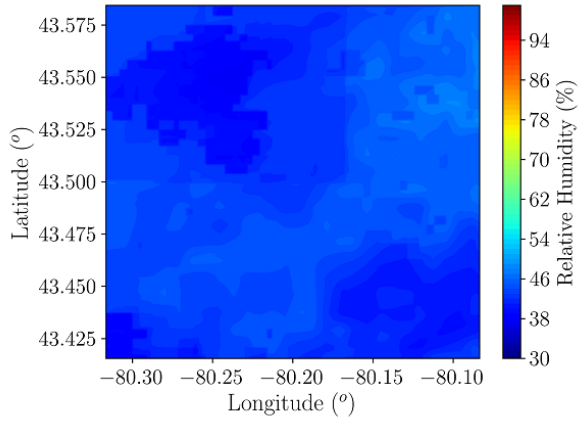
Figure 3.16: Effect of topography on 2-m relative humidity at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT).



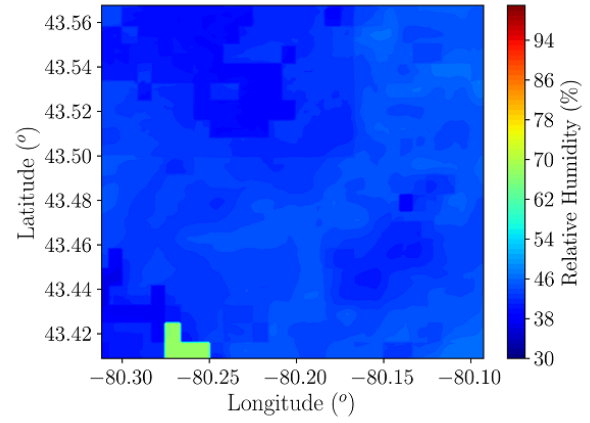
(a) GTOPO: 1400 LDT



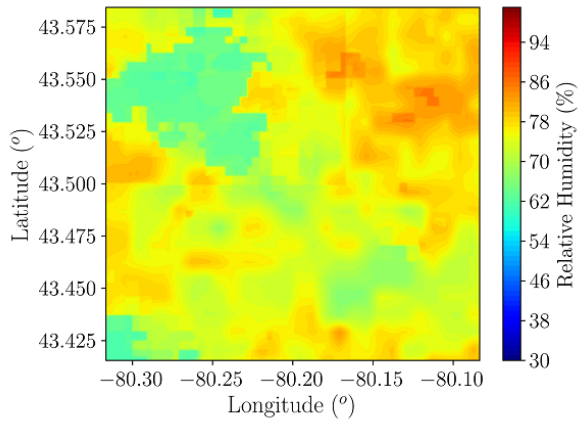
(b) SRTM: 1400 LDT



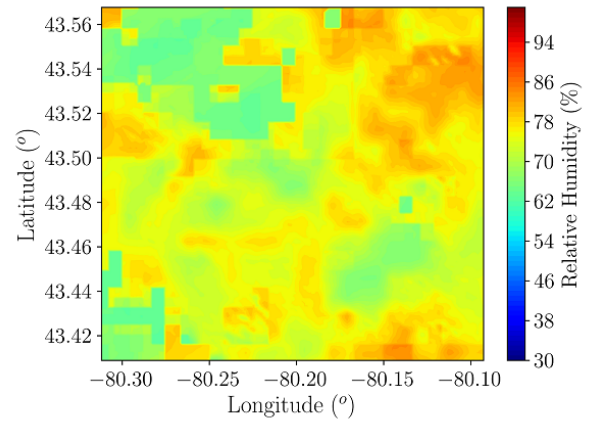
(c) GTOPO: 1800 LDT



(d) SRTM: 1800 LDT



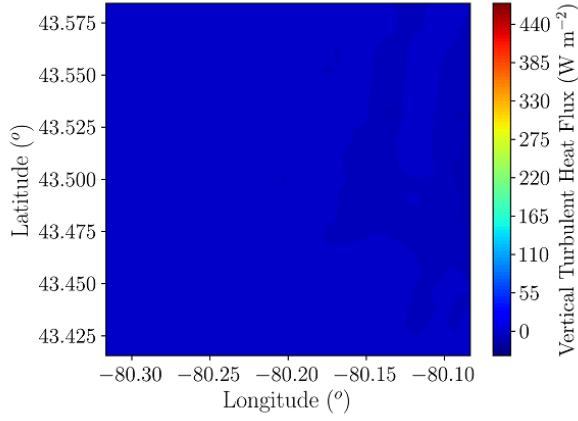
(e) GTOPO: 2200 LDT



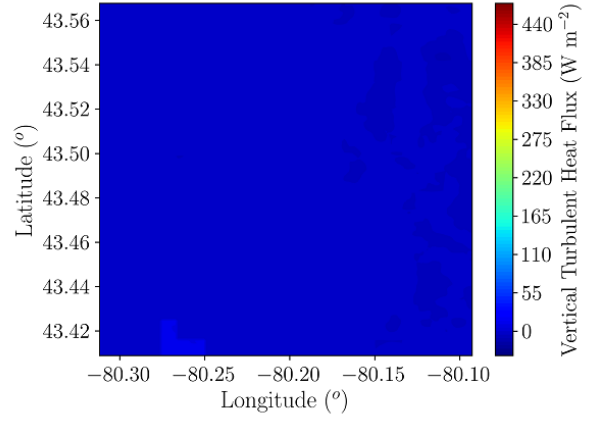
(f) SRTM: 2200 LDT

Figure 3.17: Effect of topography on 2-m relative humidity at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT).

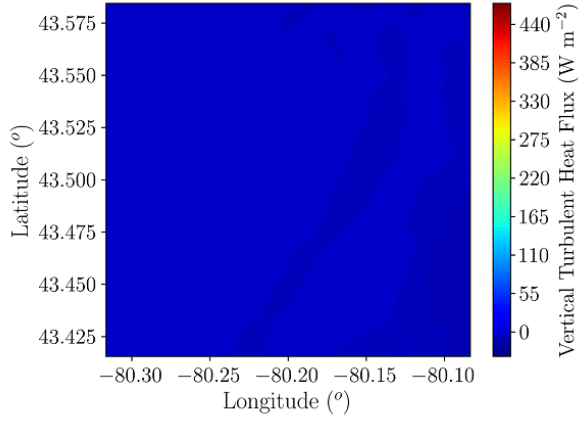




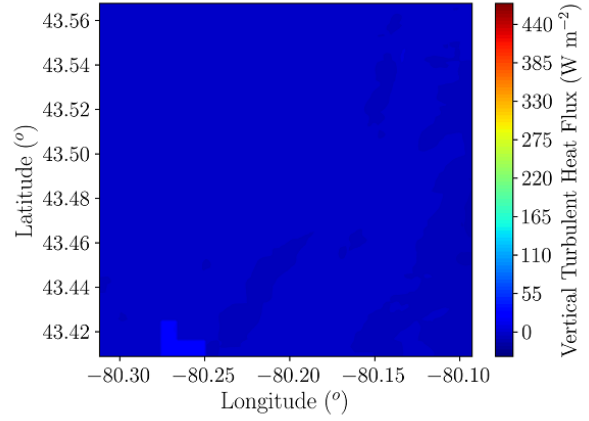
(a) GTOPO: 0200 LDT



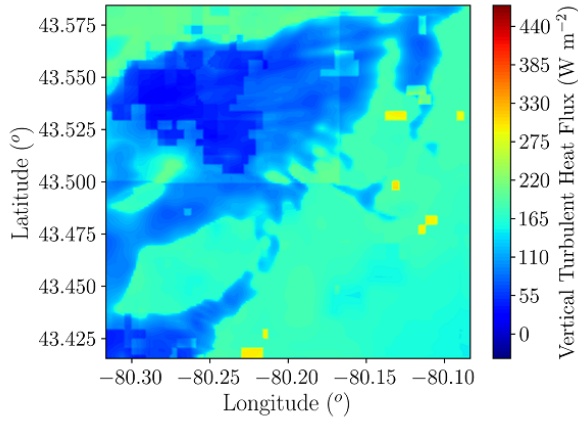
(b) SRTM: 0200 LDT



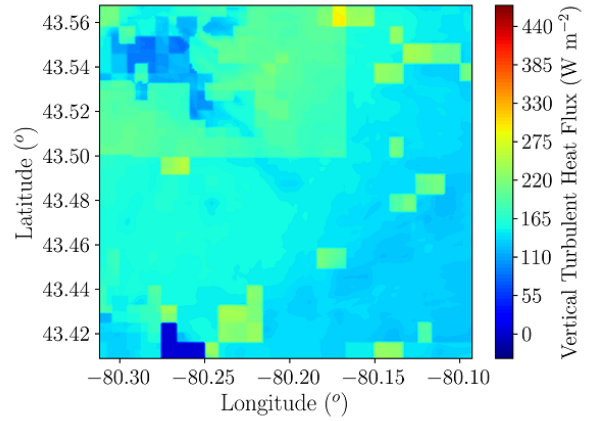
(c) GTOPO: 0600 LDT



(d) SRTM: 0600 LDT

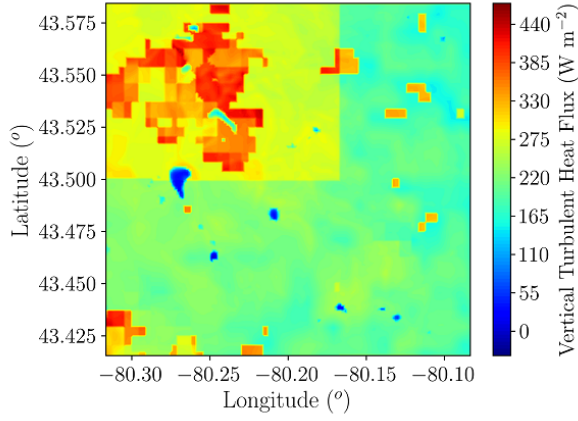


(e) GTOPO: 1000 LDT

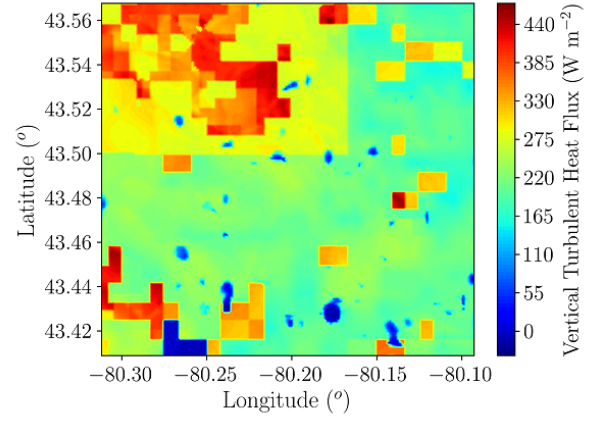


(f) SRTM: 1000 LDT

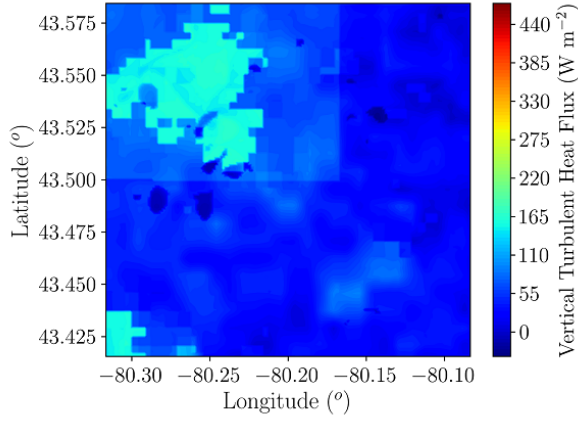
Figure 3.18: Effect of topography on 10-m vertical turbulent heat flux at 0200, 0600, and 1000 LDT on August 4, 2018; times in Local Daylight Time (LDT).



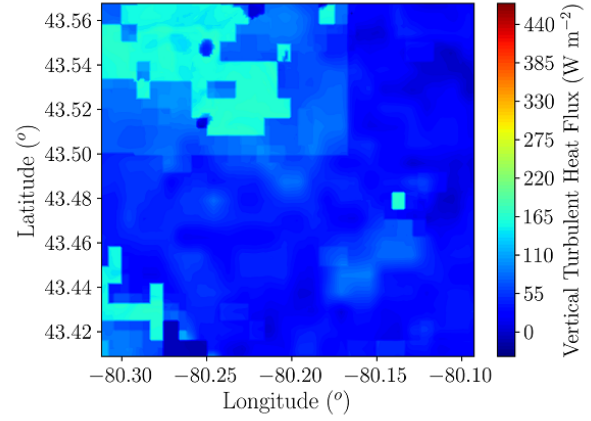
(a) GTOPO: 1400 LDT



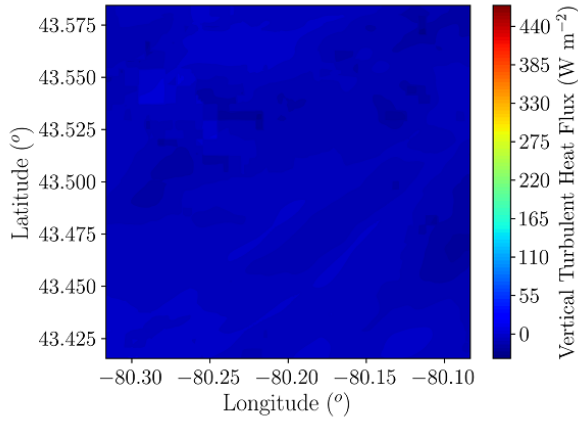
(b) SRTM: 1400 LDT



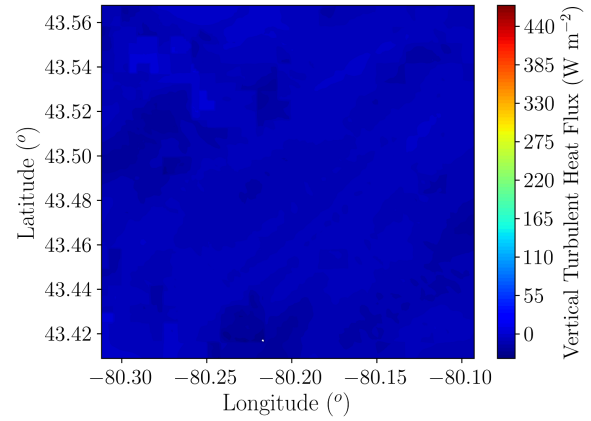
(c) GTOPO: 1800 LDT



(d) SRTM: 1800 LDT



(e) GTOPO: 2200 LDT



(f) SRTM: 2200 LDT

Figure 3.19: Effect of topography on 10-m vertical turbulent heat flux at 1400, 1800, and 2200 LDT on August 4, 2018; times in Local Daylight Time (LDT).

determining the Fractional Bias (FB) and the Normalized Mean Square Error (NMSE). Table 3.2 shows the FB and NMSE of different parameters to compare WRF outputs with experimental observations.

Table 3.2: Effect of topography showing the FB and NMSE of different atmospheric properties.

Parameter	FB		NMSE	
	GTOPO	SRTM	GTOPO	SRTM
Horizontal wind velocity at 10 m	-0.661	-0.662	0.491	0.492
Horizontal wind velocity from SODAR	-0.363	-0.345	0.137	0.123
Vertical wind velocity from SODAR	0.071	0.289	0.005	0.085
Potential temperature at 2 m	0.012	0.011	0.0001	0.0001
Relative humidity at 2 m	0.017	0.029	0.0003	0.0009

It is identified that although there are small differences in predicting horizontal wind velocity at surface level and higher altitudes, the differences between low and high topographic resolution WRF simulations are significant in predicting the vertical wind velocity. Low topographic resolution simulations realize better results than high topographic resolution simulations as far as vertical wind is concerned. WRF overpredicts the horizontal wind velocity both at surface level and higher elevation while it underpredicts the vertical velocity at the measured elevations. Moreover, the bias of horizontal wind velocity is also higher at surface level compared to higher altitudes. WRF shows a very high correlation in fractional bias between potential temperature and relative humidity. High topographic resolution simulations produce slightly better results than low topographic resolution simulations in predicting the potential temperature while low topographic resolution simulations show better agreements with experimental observation in predicting the relative humidity.

### 3.5 Conclusions

The sensitivity tests of low and high topographic resolution WRF simulations are performed to observe the urban climate both in the surface level and higher altitudes. The numerical results are compared to the experimental observations of a Mini SODAR and Guelph Turfgrass weather station to evaluate the WRF outputs. Overall, the model has presented a better performance in predicting heat-related properties such as temperature and relative

humidity compared to the momentum-related properties in agreement with a previous study [28]. It is also determined that low topographic resolution simulations show reasonable agreement with observed relative humidity while high topographic resolution simulations predict the potential temperature more accurately at surface level. On the other hand, the error of horizontal wind velocity is less with high topographic resolution WRF simulations while low topographic resolution WRF simulations deliver better results in terms of predicting the vertical wind velocity. It is also found that predicting the surface level wind velocity is one of the main limitations of WRF as the fractional bias is almost double compared to the bias of wind velocity at higher elevations. As far as the wind direction is concerned, it is also found that predictions of surface-level wind direction are less reliable, especially during thermally stable conditions, compared to wind directions at higher altitudes. However, as the atmosphere is calm during thermally stable hours, the winds are slower, which may also cause inaccurate measurement of wind directions and result in a large bias of wind direction in thermally stable conditions.

# Chapter 4

## WRF Simulations over a Complex Mining Facility

### 4.1 Experimental Observation

The simulations were run on a desktop with 24 Central Processing Units (CPUs). Due to model computational cost, no more than three days of simulations were possible for the Summer and Winter seasons. The experimental measurements were performed in an open-pit mining facility in northern Canada in May 2018 and March 2019. As shown in Figure 4.1, the facility includes a tailings pond, which is an area of refused mining waste where the waterborne refuse material is pumped, and most of the outlets of the pumps are located near the barren area on the East side of the pond. Open-pit mining excavations are primarily conducted over the mine area.

For the Summer campaign, three days, May 18, 24, and 30 in 2018, were chosen for observation-model comparison based on the availability of experimental datasets. To ensure that these three days were representative of weather conditions in May 2018, the horizontal wind velocity and air temperature during those days were compared to statistics of weather conditions observed during the entire May 2018 as measured by the Wood Buffalo Environmental Association (WBEA) weather station indicated in Figure 4.1. Figure 4.2 shows the horizontal wind velocity and temperature during these days in comparison to the median, 10th percentile, and 90th percentile of hourly observations during May 2018. Based on this investigation, the observations in those three days are reasonably close to the monthly range.

For the Winter campaign, three days, March 16, 18, and 21 in 2019, were chosen to

compare the WRF model with the experimental measurements. Comparison to monthly statistics was not performed in this case. Based on the findings of a rigorous sensitivity study conducted for the Summer campaign, WRF was executed with the optimum configurations for the Winter campaign. The following instruments were used to determine horizontal and vertical wind velocities, temperature, vertical potential temperature gradient (potential temperature lapse rate), relative humidity, and sensible heat flux at surface level and higher elevations in the mining facility. Figure 4.1 shows the location of experimental observations in the mining facility during the Summer season.

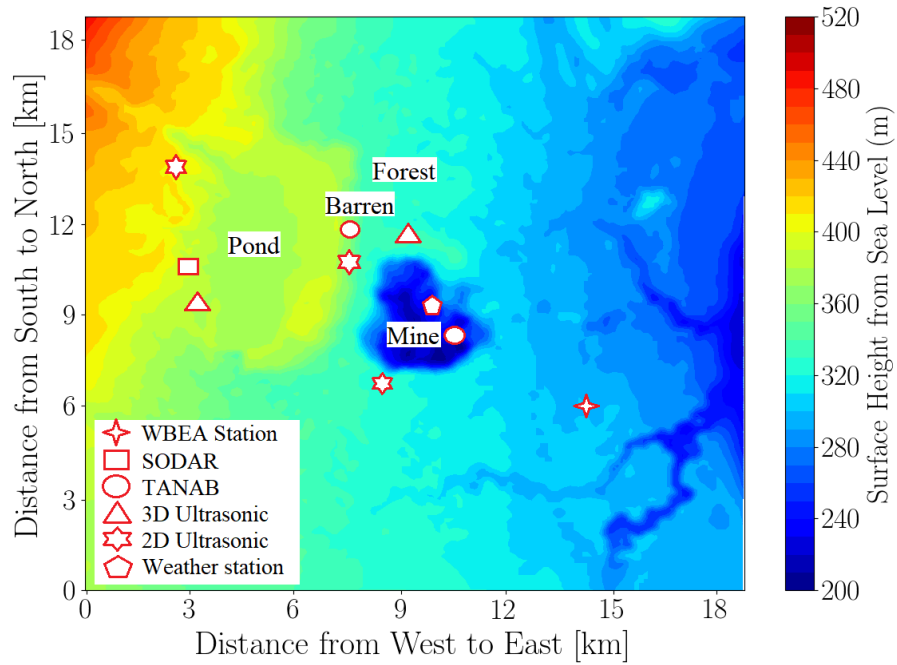
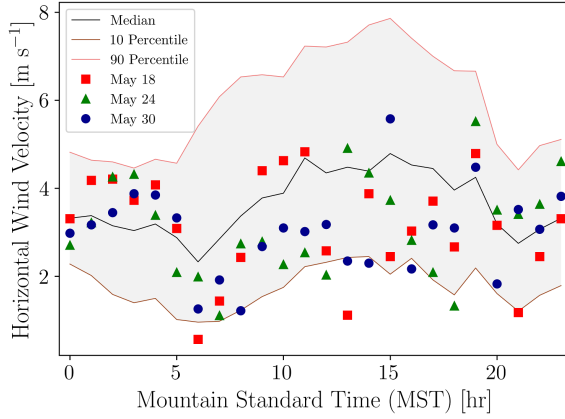


Figure 4.1: Locations of the meteorological instruments deployed at the mining facility for model comparison in the Summer campaign; Figure color coded with surface height above sea level.

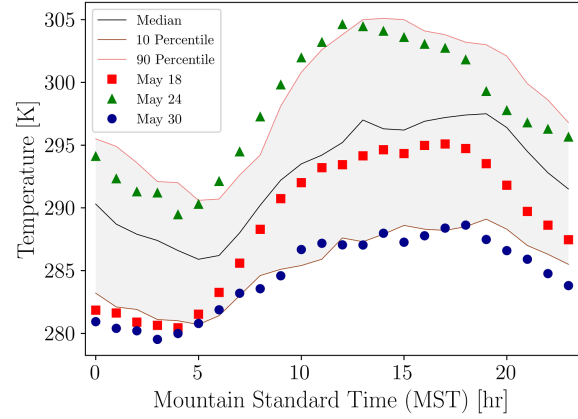
#### 4.1.1 SOnic Detections And Ranging (SODAR)

A PA-5 SODAR by Remtech Inc. was used to measure horizontal and vertical wind velocity and wind direction from 100 m to 2700 m with 100-m vertical resolution with an output frequency of 60 minutes.<sup>1</sup> This acoustic wind profiler has the capability of measuring wind

<sup>1</sup><http://www.remtechinc.com>



(a) Horizontal wind velocity at 10 m



(b) Temperature at 2 m

Figure 4.2: Horizontal wind velocity (a) and temperature (b) distribution for the entire month of May in 2018 as measured by the Wood Buffalo Environmental Association (WBEA) weather station indicated in Figure 4.1.

speed from 0 to 40 m s<sup>-1</sup>. The SODAR is perfectly suited for regional wind studies for pollutant dispersion analysis, meteorological studies, and military applications. It was located on the west side of a tailings pond.

Table 4.1: Measured parameters by PA-5 SODAR.

Parameter	Parameter description	Units
ALT	Altitude	m
CT	Echo strength	-
SPEED	Horizontal wind speed	cm s <sup>-1</sup>
DIR	Wind direction	degrees
S DIR	Standard deviation of wind direction	degrees
W	Vertical wind speed	cm s <sup>-1</sup>
SW	Standard deviation of vertical wind speed	cm s <sup>-1</sup>
SU	Standard deviation of horizontal wind speed (along wind)	cm s <sup>-1</sup>
SV	Standard deviation of horizontal wind speed (cross wind)	cm s <sup>-1</sup>
INVMI	Mixing height	m
-9999	No measurement available	-

The measured parameters by the PA-5 SODAR system are shown in Table 4.1. The SODAR works by emitting sound waves in three directions up through the atmosphere. The

sound signal sent from the transducers into the air are echoed by the air and returned to the SODAR. The time delay from when the sound signal was transmitted to when it was received indicates the sampling height based on the speed of sound. Moreover, wind speed causes a change in frequency and this change in frequency, combined with the known angles of the three sound beams, is interpreted to produce the wind profiles of the atmosphere [21, 22]. Rainfall also generates acoustic noise that has some impacts on the transmitted and received signal of the transducers [18]. The upper limit of rainfall rate for PA-5 SODAR is typically 1 to 2 cm per hour which depends on the atmospheric conditions. Moreover, the SODAR system can detect snow as snowflakes are bigger than raindrops. Furthermore, the acoustic noise due to snowflakes impacting on the transducers is negligible [2]. However, the antenna heating is applied for 2 to 3 hours to clean the transducers if there is a snow storm during observation. The ratio between the backscattered signal and the received ambient acoustic noise decreases when the surrounding noise of the SODAR increases [35]. This SODAR can measure in environments with up to 65 dBA of noise and there is an inaccuracy of measurement above this sound level. The minimum horizontal distance of an obstacle should be equal to its height to reduce the impact of ground clutter [99].

#### **4.1.2 Tethered And Navigated Air Blimp (TANAB)**

The customized Tethered And Navigated Air Blimp (TANAB) consists of fixed and variable payloads. The fixed payload is comprised of a helium balloon, the navigation tether, a tether reel, and a gondola platform housing multiple sensors. The variable payload is comprised of microclimate sensors such as the TriSonica<sup>TM</sup> Mini weather station by Applied Technologies, Inc. to measure 3D wind profiles, wind direction, pressure, temperature, and relative humidity with a frequency of 10 Hz.<sup>2</sup> It is capable of measuring the wind speed from 0 to 30 m s<sup>-1</sup>, temperature between -25 to 80 °C, pressure between 50 to 115 kPa, and relative humidity between 0 to 100 %. The measurement resolution of this mini weather station is 0.1 m s<sup>-1</sup> for wind speed, 1° for wind direction, and 0.1 °C for temperature. Moreover, the accuracy of measurement for wind speed is  $\pm 0.1$  m s<sup>-1</sup>, for wind direction is  $\pm 1^\circ$ , and for temperature is  $\pm 2$  °C. This mini weather station is ideal for applications that require a miniature, lightweight, and low-velocity anemometer, and it is suitable particularly for airborne systems. The TANAB was launched multiple times up to an altitude of 200 m from surface level both in the mine (on May 18 and 24, 2018) and near the pond (on May 30,

---

<sup>2</sup><http://www.apptech.com>



2018). Each launch took about one hour. Meteorological variables were statistically sampled every five minutes to produce medians of wind speed and potential temperature as a function of time of day and height.

### 4.1.3 3D Ultrasonic Anemometer

Two CSAT 3B ultrasonic anemometers from Campbell Scientific Inc. were used to measure the 3D wind components and temperature at 10 Hz as recommended by [5].<sup>3</sup> The anemometer has the capability of measuring wind speed up to  $30 \text{ m s}^{-1}$  and temperature from  $-30$  to  $50 \text{ }^{\circ}\text{C}$ . The measurement resolutions of these ultrasonic anemometers are  $0.001 \text{ m s}^{-1}$  for horizontal wind,  $0.0005 \text{ m s}^{-1}$  for vertical wind,  $0.058^{\circ}$  for wind direction, and  $0.002 \text{ }^{\circ}\text{C}$  for temperature. Moreover, the accuracies are  $\pm 0.08 \text{ m s}^{-1}$  for horizontal wind,  $\pm 0.04 \text{ m s}^{-1}$  for vertical wind, and  $\pm 10^{\circ}$  for wind direction. These ultrasonic anemometers were located to the north side of the mine and south-west of the pond at 10 m elevation on meteorological towers.

### 4.1.4 2D Ultrasonic Anemometers

Three R. M. YOUNG Model 86004 2D ultrasonic anemometers from R. M. YOUNG Company were operated to measure wind velocity and wind direction at 10 Hz.<sup>4</sup> These anemometers can measure wind speed in a range from 0 to  $65 \text{ m s}^{-1}$  with an accuracy of  $\pm 0.2 \text{ m s}^{-1}$  and wind direction with an accuracy of  $\pm 2^{\circ}$ . The measurement resolution of wind speed is  $0.01 \text{ m s}^{-1}$  and wind direction is  $0.1^{\circ}$ . These sensors offer high performance and low power consumption in a compact size. These 2D ultrasonic anemometers were located to the south-west of the mine, east, and north-west of the pond. They were installed at 10 m elevation on meteorological towers.

### 4.1.5 Weather Stations

Two weather stations were also operational. For the first weather station, a Gill 3-cup anemometer and model 41382 Relative Humidity/Temperature Probe from R. M. YOUNG Company were set up to measure wind velocity, wind direction, temperature, and relative humidity at 1 Hz.<sup>5</sup> The temperature and relative humidity probe can measure temperature

---

<sup>3</sup><https://www.campbellsci.ca>

<sup>4</sup><http://www.youngusa.com>

<sup>5</sup><http://www.youngusa.com>

from  $-50$  to  $50$  °C with an accuracy of  $\pm 0.3$  °C and relative humidity between 0 to 100 % with an accuracy of  $\pm 1$  %. The cup anemometer and the relative humidity/temperature probe were located at the north side of mine on a trailer which was 2 m above the ground. The second weather station was located at the south-east corner of the mining facility. The meteorological data from this station were downloaded from the Wood Buffalo Environmental Association (WBEA) website.<sup>6</sup>

### 4.1.6 LIDAR Topographical Measurements

Light Detection And Ranging (LIDAR) observations were made by Clean Harbors Company to determine the actual topography of the mining facility.<sup>7</sup> Airborne Imaging was performed to complete calibration on every LIDAR acquisition flight and data was acquired over a calibration site flown with at least two passes in opposite directions before and after the flight. Any error in the attitude of the aircraft (roll, pitch, and heading) was observed and corrected for within system specifications. To statistically quantify the accuracy, the LIDAR elevations were compared with independently surveyed ground points. This LIDAR observation has an accuracy of  $\pm 1$  m in the horizontal direction and  $\pm 0.1$  m in the vertical direction.

## 4.2 WRF Model

### 4.2.1 Domain Configurations

WRF has been tested with various configurations including the way of nesting, grid resolution, time-invariant data resolution, and geographical shifts in the domain centre. The minimum grid distance of the nested domains should be at least four grid cells from the parent boundary. Figure 4.3 and Table 4.2 illustrate the size and configuration of the domains, respectively, where domain 1 is the largest domain and domain 5 is the smallest one. The domains consist of five nested levels. The largest domain (1) covers parts of Alberta, British Columbia, Saskatchewan, and Manitoba provinces. It also covers parts of Yukon, Northwest, and Nunavut Territories. The smallest domain (5) contains primarily the mining facility.

---

<sup>6</sup><https://wbea.org/>

<sup>7</sup><http://www.airborneimaginginc.com>

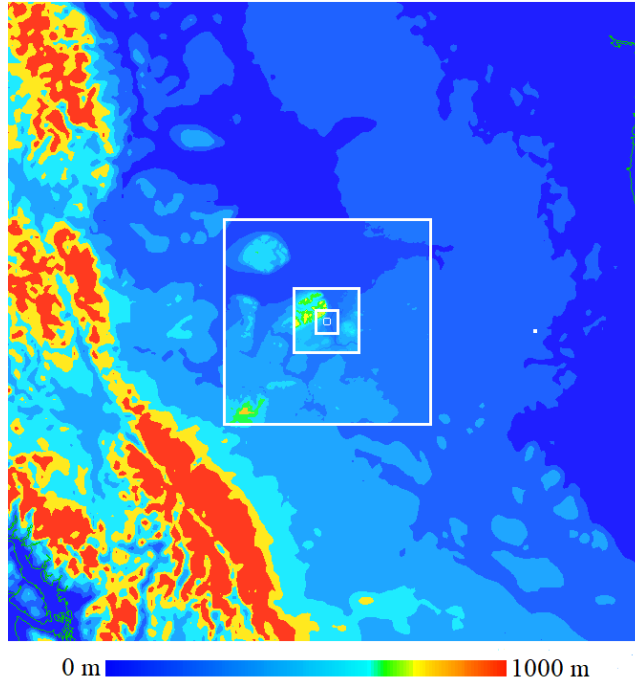


Figure 4.3: WRF simulation domains centred at the mining facility colour coded with surface height above sea level.

Table 4.2: Domain configurations and associated parameters.

Parameter	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5
Domain size (km $\times$ km)	2000	632.7	197.6	61.4	18.5
Low horizontal resolution (km)	15	5	1.67	0.56	0.18
Mid horizontal resolution (km)	10	3.33	1.11	0.37	0.12
High horizontal resolution (km)	7	2.33	0.77	0.26	0.09
Grid elements for low horizontal resolution	133	127	119	110	100
Grid elements for mid horizontal resolution	200	190	178	166	154
Grid elements for high horizontal resolution	285	271	256	244	232
Top of the domain (km)	25	25	25	25	25
Vertical levels for low vertical resolution	45	45	45	45	45
Vertical levels for mid vertical resolution	90	90	90	90	90
Vertical levels for high vertical resolution	120	120	120	120	120
Output frequency (min)	60	60	60	60	60
GTOPO 30s topographical resolution (m)	900	900	900	900	900
SRTM 1s/LIDAR topographical resolutions (m)	-	-	-	30	30

### 4.2.2 Initialization

For the Summer campaign, the WRF simulations are executed for three different days on May 18, 24, and 30 in 2018, for 36 hours in each day including a spin-up time of 12 hours. Different configurations have been studied in sensitivity tests varying the topographic resolution, land use, horizontal grid resolution, number of vertical levels, and PBL parameterization. The simulations were run with the non-hydrostatic ARW solver while one-way nesting method was chosen for better stability [102]. WRF provides the option for obtaining time-varying meteorological initialization fields from different sources. The Global Forecasting System (GFS) dataset has been used for the simulations to provide initial and time-varying boundary conditions to WRF at 3-hour time resolution and  $0.5^\circ$  spatial resolution [39, 106, 126].<sup>8</sup> The other implemented physics are Lin et al. [70] scheme for microphysics, Multi-scale Kain-Fritsch [133] scheme for cumulus physics, and Dudhia [41] scheme for longwave and shortwave radiations. For computational stability, the magnitude of timestep satisfies a conservative Courant number choice for the coarsest grid and is further divided in subsequent finer nested grids by the same ratio as the horizontal grid refinement. A ratio of  $\frac{1}{3}$  for both timestep and horizontal resolution in successive nested grids was used as suggested [106]. For example, at high horizontal and mid vertical resolutions, the model timesteps were in the order of 200 s to 2 s for domains 1 to 5, respectively.

### Topography and Land Use

As shown in Figure 4.4, the default Global 30 Arc-Second (GTOPO 30s) and the Shuttle Radar Topography Mission 1 Arc-Second (SRTM 1s) datasets are used to modify the resolution of topography in domains 4 and 5. GTOPO 30s is a global dataset covering the full extent of the earth. As a low-resolution topographic initialization, the GTOPO 30s dataset has a topographic horizontal resolution of 900 m by 900 m. On the other hand, the SRTM 1s dataset contains digital elevation models on a near-global scale to generate the most complete high-resolution digital topographic database of the earth. The SRTM 1s dataset has a topographic resolution of 30 m by 30 m [38, 72]. Moreover, in this study, the SRTM 1s dataset is corrected with the LIDAR topographical measurements to include the actual topography of the mining facility in WRF simulations at a resolution of 1 m by 1 m.

Land Use (LU) classifications in WRF describe surface properties such as temperature, roughness, permeability, moisture, etc. that are important drivers for atmospheric dynamics.

---

<sup>8</sup><https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>

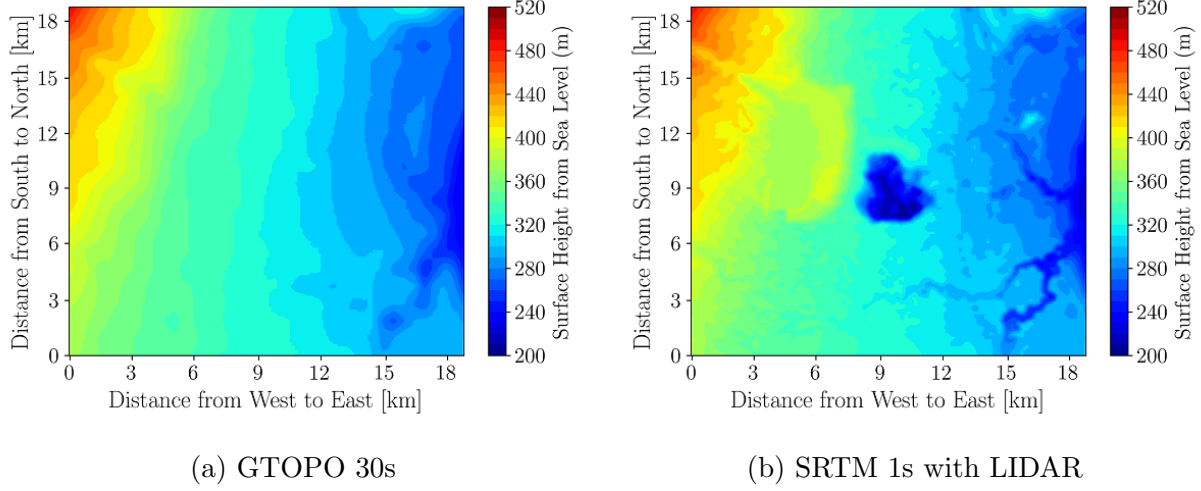
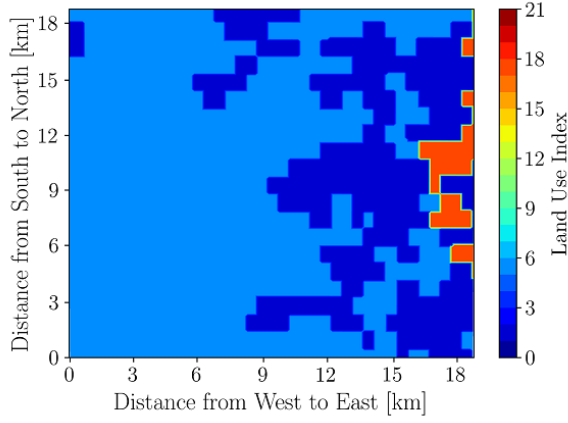
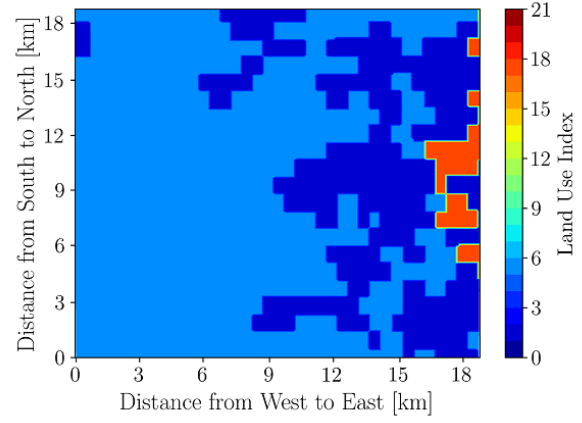


Figure 4.4: Surface height from sea level: GTOPO 30s is the non-modified land; SRTM 1s with LIDAR contains land modifications showing the mine and the tailings pond.

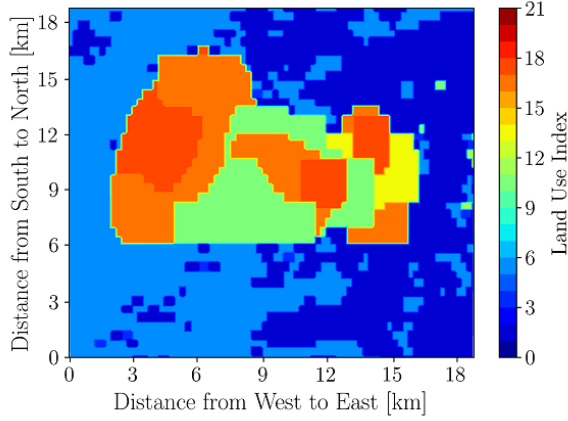
In the mining facility, changes in land surface must be correctly accounted for to be able to realistically simulate atmospheric transport and dynamics. LU classification has significant impact on meteorological simulations [28, 124]. Three LU classifications are considered for the simulations. In the first classification, the default WRF LU from a MODerate Resolution Imaging Spectroradiometer (MODIS) 30s data product (Modified 20-category IGBP-MODIS land use) is considered. This LU does not observe the existence of the mining facility. In the second classification, WRF LU is overwritten with MODIS 15s data product (Modified 20-category IGBP-MODIS land use) reflecting the existence of the mining facility. The following classifications have been used: pond (water = 17), mine (barren = 16), plant and lodge (urban and built-up = 13), and other (grassland = 10). This modification overwrites the water temperature by the nearest lake temperature [44]. In the third classification, WRF LU is overwritten with MODIS 30s data product (Modified 21-category IGBP-MODIS land use), also reflecting the existence of the mining facility. The following classifications have been used: pond (lake = 21), mine (barren = 16), plant and lodge (urban and built-up = 13), and other (grassland = 10) [44]. With this option, however, a one-dimensional mass and energy balance scheme for the pond is considered to model it as a lake [48, 111]. A depth of 50 m was considered and the lake model accounted for 25 model layers including up to 5 layers for snow, 10 layers for water, and 10 layers for soil. LU changes in WRF were only considered for the topography case considering modifications with the SRTM-LIDAR



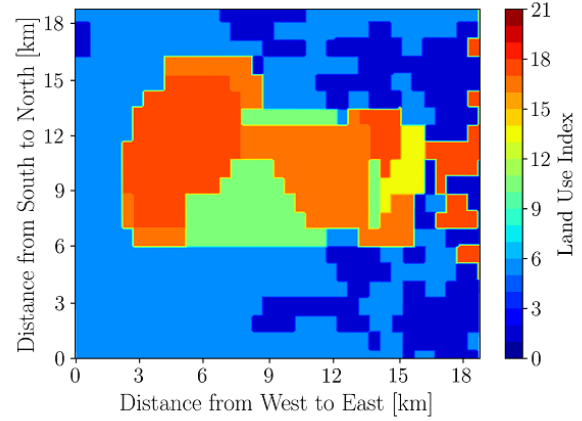
(a) GTOPO 30s non-modified land use



(b) SRTM-LIDAR non-modified land use



(c) SRTM-LIDAR with modified land use



(d) SRTM-LIDAR with modified land use and lake model

Figure 4.5: Topography and land use classifications for WRF simulations.

topography datasets. The aerodynamic roughness length throughout the 5 domains was also appropriately forced by specifying the land use classification. Figure 4.5 shows the three LU configurations for WRF simulations. Note that only MODIS 30s data product allowed the inclusion of the lake model, so it must have been considered over the MODIS 15s data product.

### **Planetary Boundary Layer Parameterization**

The planetary boundary layer (PBL) scheme implemented in the model plays a decisive role in the accuracy of the flow within the PBL as the wind varies according to the stability of the atmosphere. Two PBL parameterization options are considered. Researchers have performed sensitivity tests of the WRF model and found that the Yonsei University (YSU) scheme shows improvement over other PBL schemes of WRF [26]. Hence, YSU scheme is applied as the first option. For the second option, the Large-Eddy Simulation (LES) is used assuming that the resolved turbulent and terrain-driven eddies are mainly responsible for the bulk of the vertical and horizontal transfer of momentum, heat, water vapor, and atmospheric constituents with the PBL [31, 112].

### **Grid Resolution**

One of the main objectives of enhancing horizontal and vertical grid resolutions is to reach a grid resolution that will be adequate to explicitly resolve atmospheric dynamics within the PBL [9, 96]. Hence, the grid spacing is changed both in horizontal and vertical directions to find the optimum grid resolution in this study. The grid resolutions for YSU WRF simulations in literature are varied from 0.44 km to 4 km [26, 83]. On the other hand, high horizontal grid spacing (typically below 0.1 km) is implemented for WRF-LES simulations to resolve the energetic scales of turbulence more accurately [79, 119]. Hence, for consistency between YSU WRF and WRF-LES simulations, the horizontal grid spacing for low, mid, and high resolutions in domain 5 are 0.18, 0.12, and 0.09 km, respectively. On the other hand, the default 45 vertical levels are considered as low vertical level, wherein, 90 and 120 vertical levels are implemented to obtain mid and high vertical resolutions, respectively. Moreover, the low, mid, and high vertical levels provide 13, 17, and 21 vertical levels below 2700 m elevation, respectively. Grid resolution details for all domains are provided in Table 4.2.

### 4.2.3 Evaluation

The results of the WRF model should be evaluated against observations because of the uncertainty in model implementation for any specific application. Hence, experimental datasets from SODAR are used to evaluate the WRF simulations at higher elevation wind speed and direction, while TANAB, 2D/3D ultrasonic anemometers, and weather station datasets are used for comparison of the WRF simulation results to surface-level observations. Note that since TANAB flies in low altitudes up to 200 m, it is considered a surface level instrument. A quantitative comparison between the WRF model and experimental observations is performed by determining the bias and Root Mean Square Error (RMSE) defined by

$$BIAS = \frac{\sum_{i=1}^n (M_i - O_i)}{n}, \quad (4.1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (M_i - O_i)^2}{n}}. \quad (4.2)$$

Here,  $M_i$  and  $O_i$  represent the WRF results and experimental observations, respectively, on an hourly basis for all three days of analysis combined, and  $n$  represents the sample size. Moreover, as the wind direction is a circular variable, differences of wind direction between WRF and the observations are always reported within  $\pm 180^\circ$  as shown in Figure 4.6 [82].

A two-sample statistical estimation test is also performed to calculate the range for the difference in bias for a given atmospheric property when running the model under different configurations. This inferential test quantifies the change in bias and therefore quantifies any model improvement or deterioration as a result of changing model configurations. If  $\bar{x}_1$  and  $s_1^2$ , and  $\bar{x}_2$  and  $s_2^2$  are the average and standard deviation of independent samples of size  $n_1$  and  $n_2$ , respectively, an approximate  $(1 - \alpha)$  100% confidence interval for  $\mu_1 - \mu_2$ , i.e. the difference between the means of biases, is

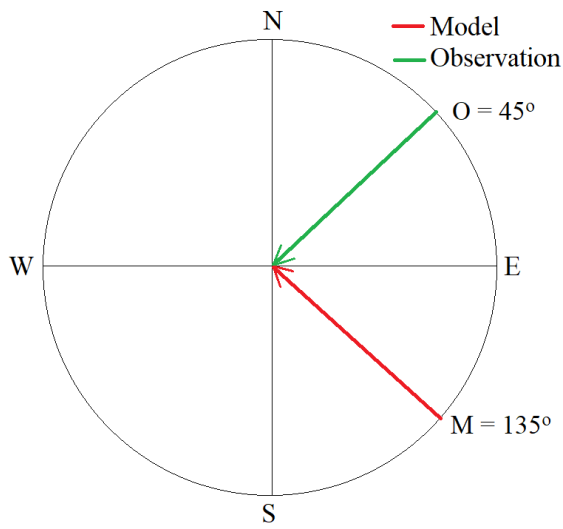
$$(\bar{x}_1 - \bar{x}_2) - t_{\alpha/2} \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} < \mu_1 - \mu_2 < (\bar{x}_1 - \bar{x}_2) + t_{\alpha/2} \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}, \quad (4.3)$$

where  $t_{\alpha/2}$  is the  $t$ -value with degrees of freedom

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}}, \quad (4.4)$$

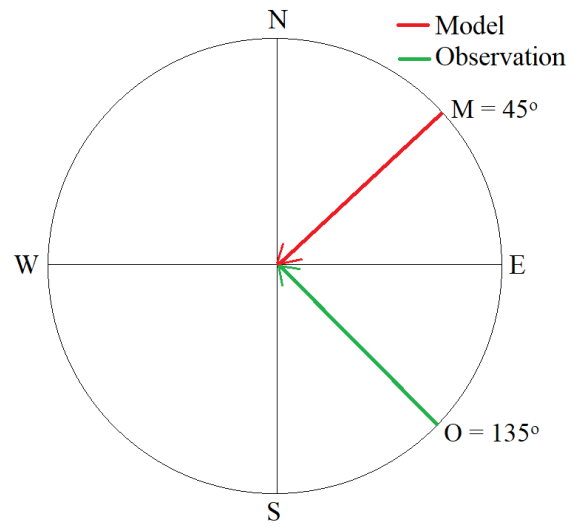
leaving an area  $\alpha/2$  to the right of a  $t$  probability distribution function [118].





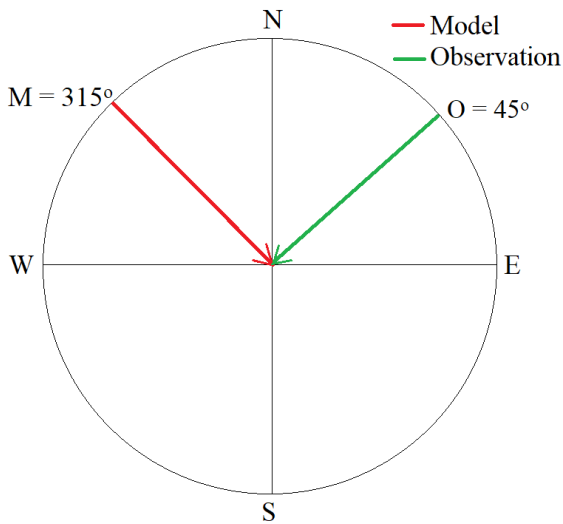
$$M - O = (135 - 45)^\circ = 90^\circ$$

(a) Case 1: Bias  $\leq 180^\circ$



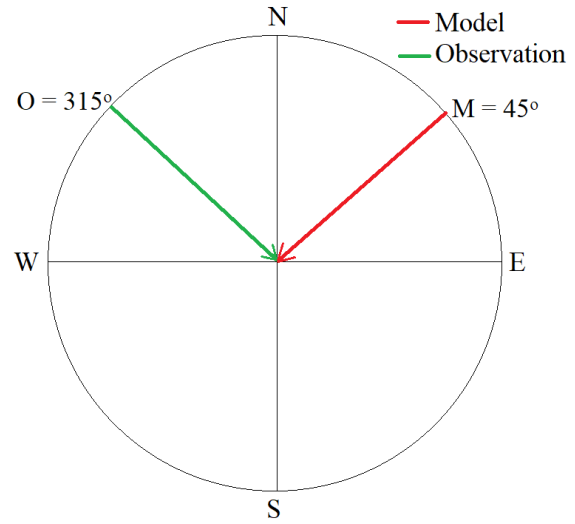
$$M - O = (45 - 135)^\circ = -90^\circ$$

(b) Case 2: Bias  $\geq -180^\circ$



$$M - O = (315 - 45)^\circ = 270^\circ \rightarrow (270 - 360)^\circ = -90^\circ$$

(c) Case 3: Bias  $> 180^\circ$



$$M - O = (45 - 315)^\circ = -270^\circ \rightarrow (-270 + 360)^\circ = 90^\circ$$

(d) Case 4: Bias  $< -180^\circ$

Figure 4.6: Wind direction bias calculation for different cases to report the bias within  $\pm 180^\circ$ .

## 4.3 Results and Discussion

The comparison between WRF simulations in domain 5 and the experimental observations in the Summer and Winter campaigns are evaluated in this discussion. The simulations associated with the Summer campaign are rigorously examined with respect to the effects of topography and land use modifications. Subsequently, the effects of horizontal/vertical resolutions and planetary boundary layer parameterization are discussed. For statistical analysis, both hourly statistics and overall error statistics are calculated. For hourly statistics, the median values of every property for the three simulation dates and observations are considered. This is motivated by the fact that the mean value may not be a fair representation of the data due to the mean value being easily influenced by the outliers in the data. For the overall statistics, the bias and RMSE of every property at every diurnal time and multiple observation locations are computed. The analysis is followed by examining the difference between GFS forecast and reanalysis forcing data as well as the performance of the model associated with the Winter campaign.

### 4.3.1 Effect of Topography and Land Use

The effects of topography, land use, and lake model are discussed in this section. For this investigation, mid horizontal/vertical resolutions WRF simulations are run to minimize computational expense. Moreover, the Yonsei University (YSU) scheme is chosen for PBL parameterization because LES cases sometimes diverged during running the WRF simulations on the mid horizontal/vertical resolutions.<sup>9</sup> The bias and RMSE of different properties are shown in Table 4.3. Moreover, the two-sample estimation tests among different cases are also presented in Table 4.4 showing a range for the difference in bias for pairs of model configurations.

#### Wind Velocity

Figure 4.7 shows the horizontal wind velocity in domain 5 near the surface level for different types of topography and land use configurations. It should be noted that the TANAB system is considered as a surface level measurement because it only probes the atmospheric boundary layer up to low altitudes typically under 200 m.

---

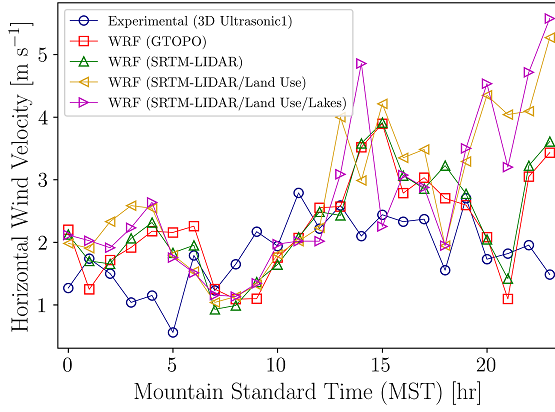
<sup>9</sup>The model did not run to completion, and it crashed before the end of simulation time.

Table 4.3: Effects of topography and land use change on atmospheric dynamics; the table shows the bias and RMSE of different atmospheric properties between the WRF model and field observations.

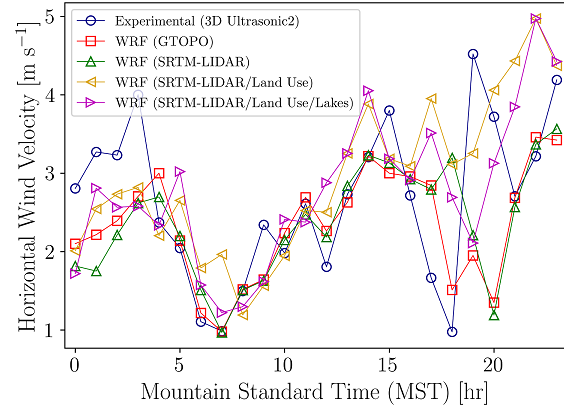
Parameter	Errors	GTOPO	SRTM-LIDAR	SRTM-LIDAR/LU	SRTM-LIDAR/LU/Lakes
Horizontal wind velocity at 10 m ( $\text{m s}^{-1}$ )	BIAS	-0.29	-0.28	0.04	0.02
	RMSE	1.58	1.58	1.53	1.57
Wind direction at 10 m ( $^{\circ}$ )	BIAS	0.33	-1.28	-1.37	-3.39
	RMSE	73.9	74.3	76.1	73.6
Horizontal wind velocity from TANAB ( $\text{m s}^{-1}$ )	BIAS	0.77	0.29	0.47	0.52
	RMSE	1.84	1.52	1.70	1.87
Horizontal wind velocity from SODAR ( $\text{m s}^{-1}$ )	BIAS	0.10	0.15	0.43	0.55
	RMSE	2.14	2.14	2.34	2.34
Wind direction from SODAR ( $^{\circ}$ )	BIAS	2.65	-0.83	-0.47	0.19
	RMSE	75.8	76.5	77.2	76.8
Temperature at 2 m ( $^{\circ}\text{C}$ )	BIAS	-0.66	-0.59	0.04	0.17
	RMSE	2.39	2.30	2.51	2.47
Potential temperature gradient from TANAB ( $\text{K m}^{-1}$ )	BIAS	0.016	0.016	0.016	0.015
	RMSE	0.027	0.027	0.027	0.028
Relative humidity at 2 m (%)	BIAS	0.46	0.22	-2.74	-1.05
	RMSE	9.62	9.51	9.40	9.67
Sensible heat flux at 10 m ( $\text{W m}^{-2}$ )	BIAS	40.4	37.4	12.6	14.8
	RMSE	97.0	96.8	56.4	56.5

Table 4.4: Effects of topography and land use changes on atmospheric dynamics showing the two-sample estimation test ( $\alpha=0.1$ ) among different cases; a range for difference in bias is reported for pairs of model configurations.

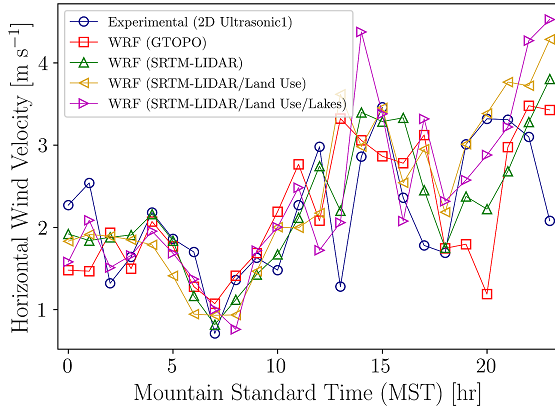
Parameter	GTOPO		GTOPO		GTOPO	
	SRTM-LIDAR		SRTM-LIDAR/LU		SRTM-LIDAR/LU/Lakes	
	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum
Horizontal wind velocity at 10 m ( $\text{m s}^{-1}$ )	-0.44	0.42	-0.75	0.10	-0.74	0.12
Wind direction at 10 m ( $^{\circ}$ )	-18.7	21.9	-18.9	22.3	-16.5	23.9
Horizontal wind velocity from TANAB ( $\text{m s}^{-1}$ )	0.02	0.94	-0.19	0.79	-0.26	0.76
Horizontal wind velocity from SODAR ( $\text{m s}^{-1}$ )	-0.64	0.54	-0.94	0.28	-1.06	0.16
Wind direction from SODAR ( $^{\circ}$ )	-17.4	24.4	-17.9	24.1	-18.5	23.4
Temperature at 2 m ( $^{\circ}\text{C}$ )	-0.71	0.57	-1.37	-0.03	-1.50	-0.16
Potential temperature gradient from TANAB ( $\text{K m}^{-1}$ )	-0.007	0.007	-0.007	0.007	-0.007	0.008
Relative humidity at 2 m (%)	-2.38	2.86	0.59	5.81	-1.13	4.15
Sensible heat flux at 10 m ( $\text{W m}^{-2}$ )	-23.6	29.6	6.01	49.6	3.84	47.4



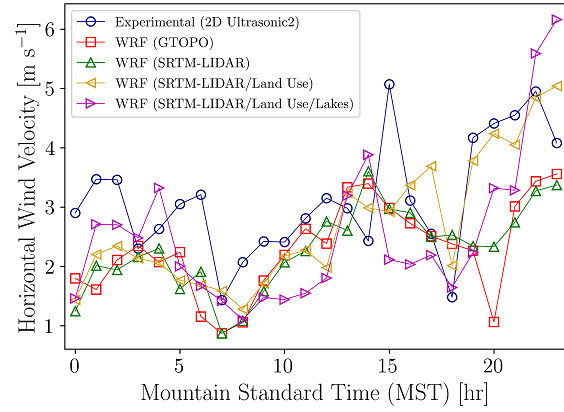
(a) 3D Ultrasonic 1



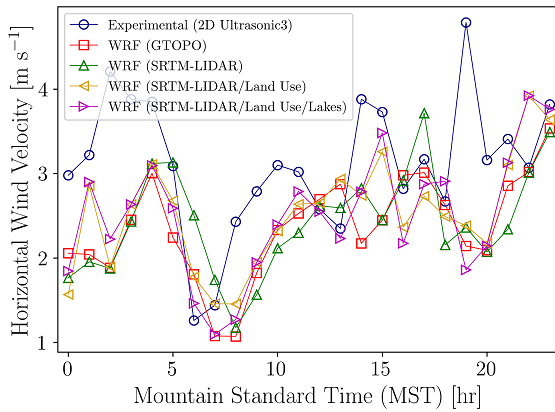
(b) 3D Ultrasonic 2



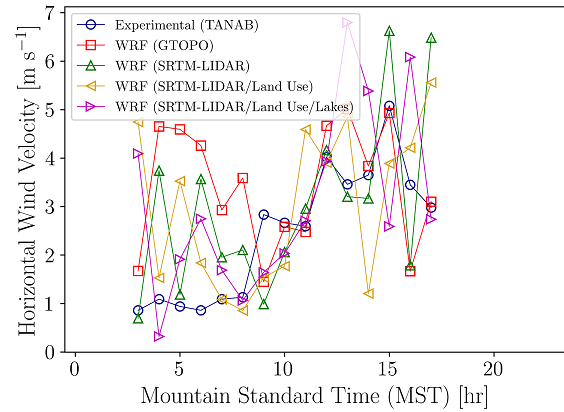
(c) 2D Ultrasonic 1



(d) 2D Ultrasonic 2



(e) 2D Ultrasonic 3



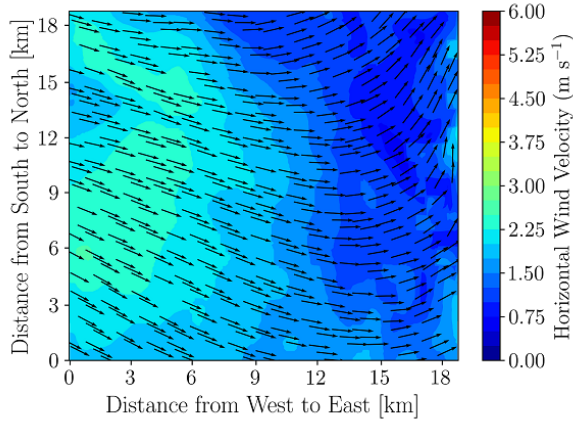
(f) TANAB

Figure 4.7: Effects of topography and land use changes on horizontal wind velocity near surface level showing the median values of model and observation for each hour; here the instruments were set up at 10 m elevations from the surface except for the TANAB.

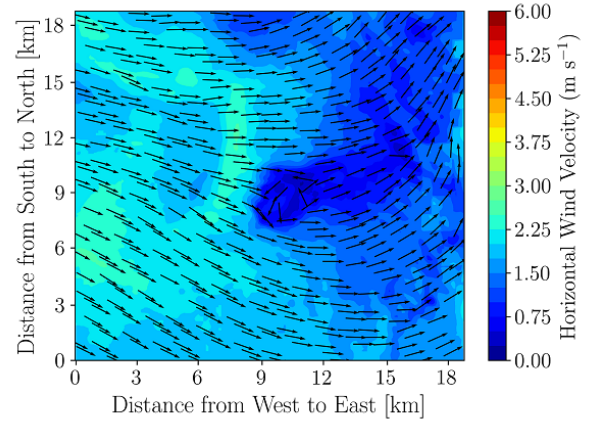
It is observed that there is no major difference of wind speed predictions when comparing to 2D/3D ultrasonic anemometers at 10 m although at midnight and early morning the WRF case with SRTM-LIDAR topography, modified land use, and lake model shows better agreement with the experimental observation. However, the model shows the improvement of wind velocity prediction with SRTM-LIDAR topography when compared to TANAB observations with errors reduced by 0.02 to 0.94 m s<sup>-1</sup> ( $\alpha = 0.1$ ). As the variations of wind speed at low altitudes depend on the surface roughness and atmospheric stability [25], the WRF case without the mine topography cannot perform accurate simulations. Moreover, in convective hours when the atmosphere is unstable, WRF shows less bias than the thermally stable conditions at late night and early morning, in agreement with findings of [109].

Figures 4.8, 4.9 and B.1 to B.4 show the horizontal wind velocity in domain 5 at 10 m for different types of topography and land use configurations on May 18, 2018 during thermally stable and unstable conditions. There are some circulations over the mine cavity at 0200 MST in most of the WRF simulations accounting for the mine terrain. As a result, winds are slower in that area and exhibit circulation [75]. Meanwhile, during nighttime the temperature of the pond is more than the surrounding land surfaces and the slightly warmer pond temperatures drive an airflow mechanism in cases accounting for land use modification and lakes model. However, at 1400 MST when the atmosphere is mostly unstable, there are no major differences of wind velocity and wind direction among various topographies although some circulations are observed over the pond area for the case with SRTM-LIDAR topography and land use modification in WRF.

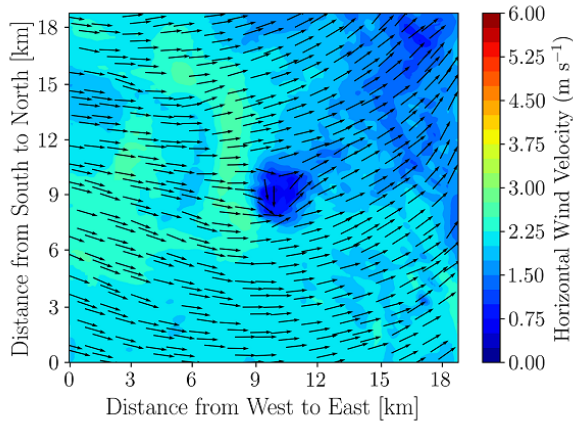
Meteorological complexities in the site as a function of space and time can be best demonstrated using vertical profiles of wind velocity at different locations such as forest, pond, mine, and barren areas. For this purpose, four profiles were considered for further analysis of the WRF simulation results. Two profiles were considered at the centre of the mine and the pond to represent atmospheric columns over these areas. A forest profile just north of the mine was considered. In addition, another profile in between the mine and pond was chosen to represent the developed barren areas of the site. Figure 4.10 shows the horizontal wind velocity profiles in domain 5 for SRTM-LIDAR with land use and lake model at different locations on May 18, 2018. In agreement with surface level contour plots of wind speed, the mine is more calm than other locations at night time and early morning hours when the atmosphere is thermally stable. On the other hand, the horizontal wind velocity at lower altitudes increases in convective hours when the atmosphere is thermally unstable. There is a difference between the magnitude of wind speed at surface level and



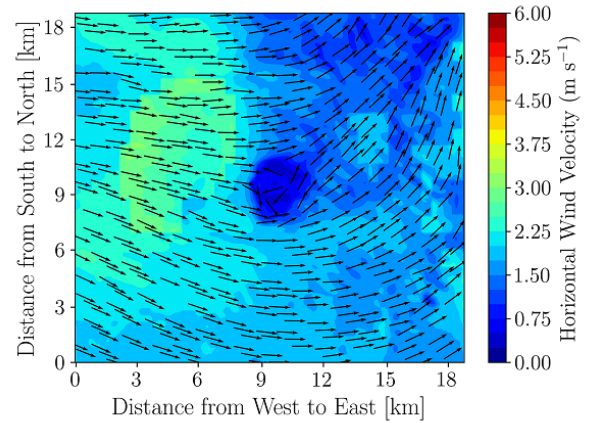
(a) GTOPO



(b) SRTM-LIDAR

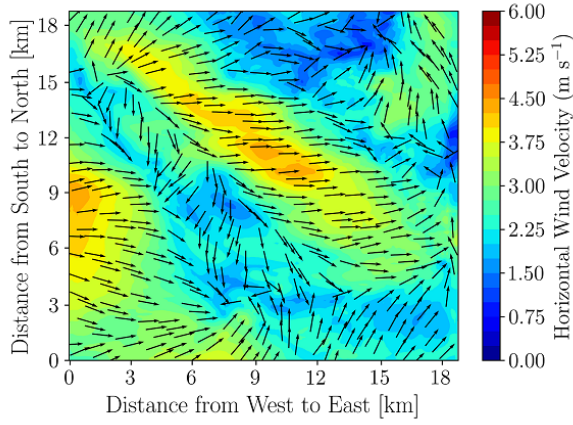


(c) SRTM-LIDAR with land use

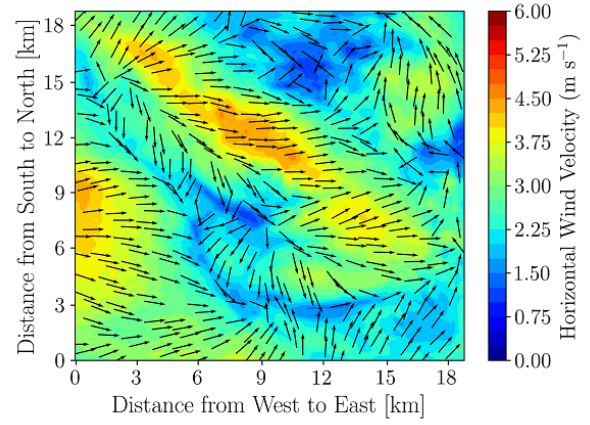


(d) SRTM-LIDAR with land use and lake model

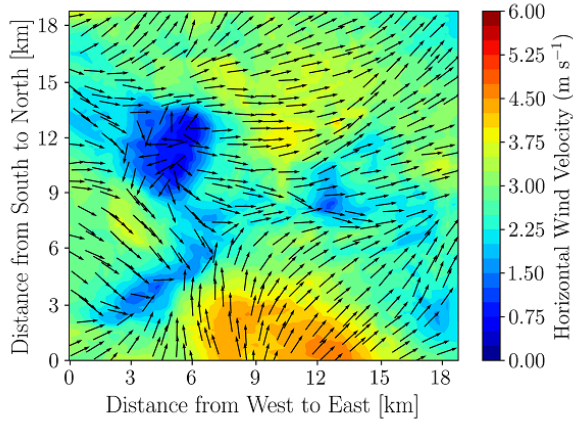
Figure 4.8: Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 0200 MST on May 18, 2018; times in Mountain Standard Time (MST).



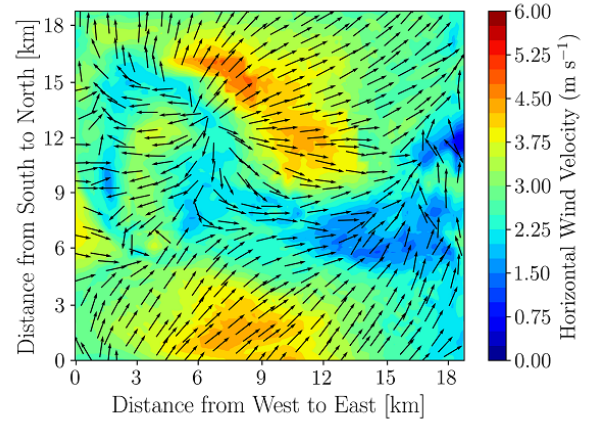
(a) GTOPO



(b) SRTM-LIDAR

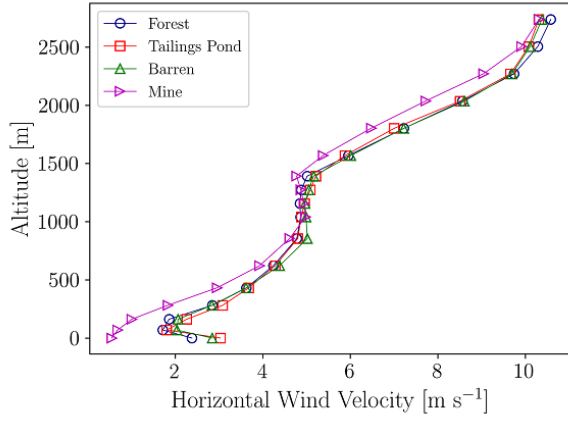


(c) SRTM-LIDAR with land use

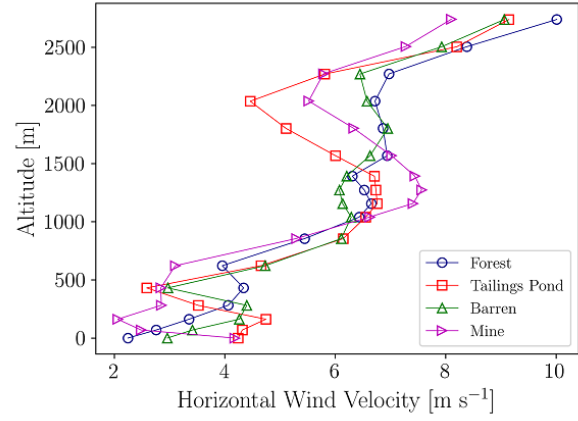


(d) SRTM-LIDAR with land use and lake model

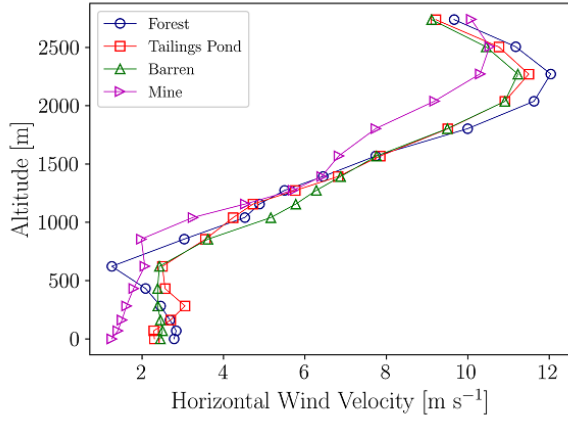
Figure 4.9: Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 1400 MST on May 18, 2018; times in Mountain Standard Time (MST).



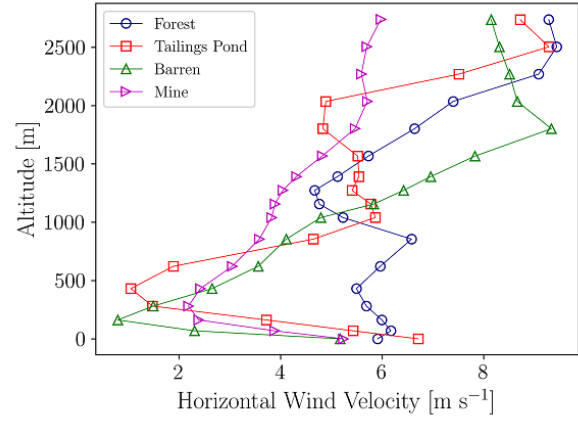
(a) Horizontal wind velocity at 0200 MST



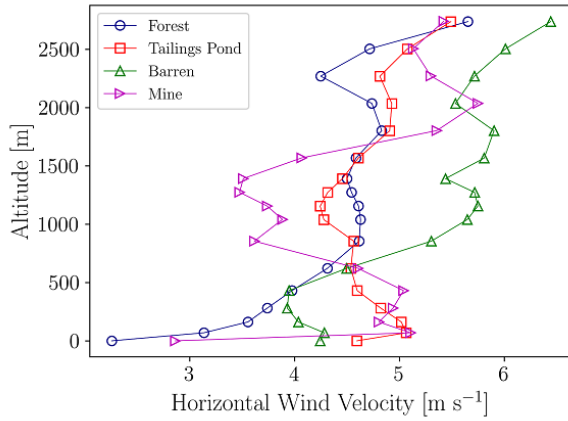
(b) Horizontal wind velocity at 0600 MST



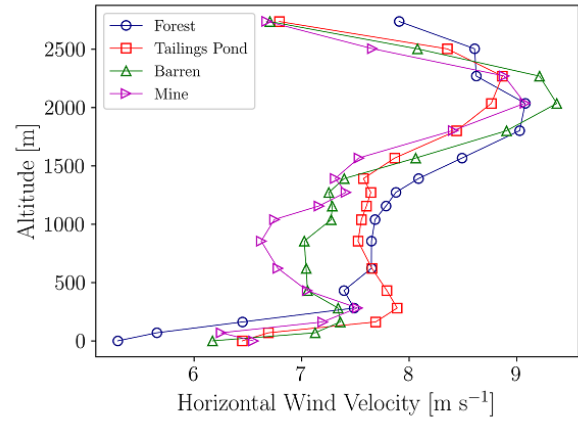
(c) Horizontal wind velocity at 1000 MST



(d) Horizontal wind velocity at 1400 MST



(e) Horizontal wind velocity at 1800 MST



(f) Horizontal wind velocity at 2200 MST

Figure 4.10: Horizontal wind velocity profiles based on pressure height at different locations on May 18, 2018; times in Mountain Standard Time (MST).



higher altitudes within the lower portion of the PBL during the thermally stable conditions. On the other hand, during the thermally unstable condition, a more or less vertically uniform wind speed is observed throughout the lower portion of the PBL.

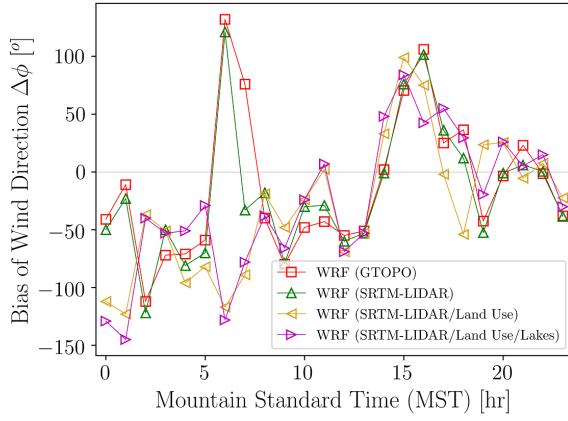
## Wind Direction

Figure 4.11 shows the median values of wind direction bias in domain 5 near the surface level for different types of topography and land use configurations. There are no major differences in wind direction among different WRF simulations. As the weather station inside the mine did not measure wind direction, so the circulation of winds in the mine could not be confirmed experimentally. However, there is a higher bias of wind direction in thermally stable conditions while the bias is less during thermally unstable conditions. In stable conditions, the wind flows very slowly which causes more complexity to determine the exact wind direction for a specific time. Also, it is well known that PBL parameterizations are less accurate for thermally stable conditions, so less accurate results were expected [7]. In addition, the bias is large in most of the sensors during thermally neutral conditions when atmospheric stability alternates from stable to unstable or unstable to stable. Similar conclusions were reached where WRF simulations were compared with surface-level observations by Talbot et al. [112].

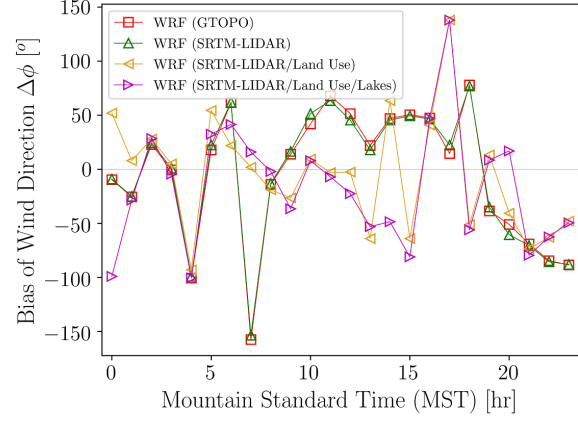
Figure 4.12 shows the wind direction profiles in domain 5 for SRTM-LIDAR topography with land use modification and lake model at different locations on May 18, 2018. The wind direction over the mine area is significantly different from other locations at 0200 MST due to the circulation occurring in that region near the surface level. The deviations of wind direction are the greatest at lower altitudes both in stable and unstable conditions. Surface friction due to topography plays an important role in the speed and direction of surface winds. However, the effect of friction for topography usually does not extend to higher elevations explaining the more or less similar wind directions for the four profiles at higher altitudes.

## Temperature

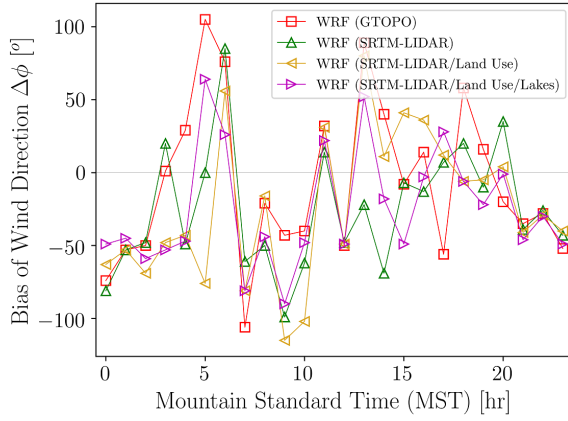
Figure 4.13 shows the temperature at 2 m in domain 5 for different types of topography and land use. The results show that WRF exhibits lower error in predicting temperature than wind speed and wind direction [28]. Moreover, WRF shows reasonable agreements both in stable and unstable conditions. However, SRTM-LIDAR topography cases with land use



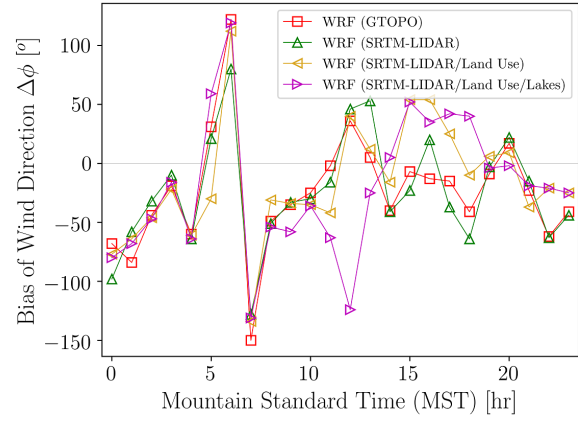
(a) 3D Ultrasonic 1



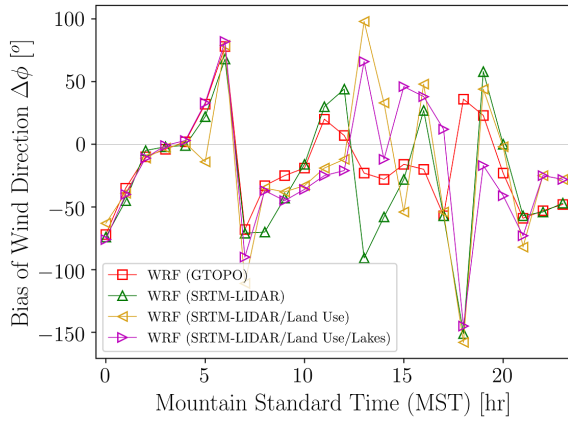
(b) 3D Ultrasonic 2



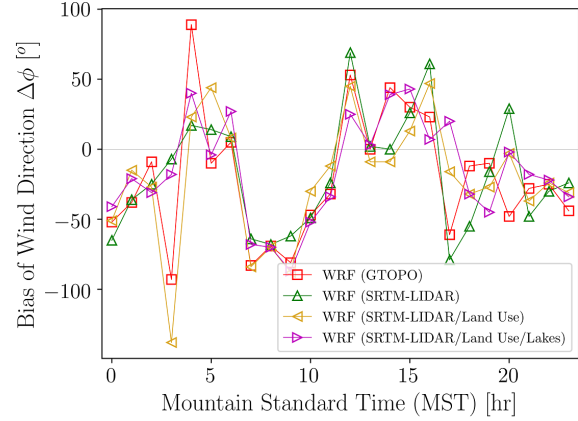
(c) 2D Ultrasonic 1



(d) 2D Ultrasonic 2

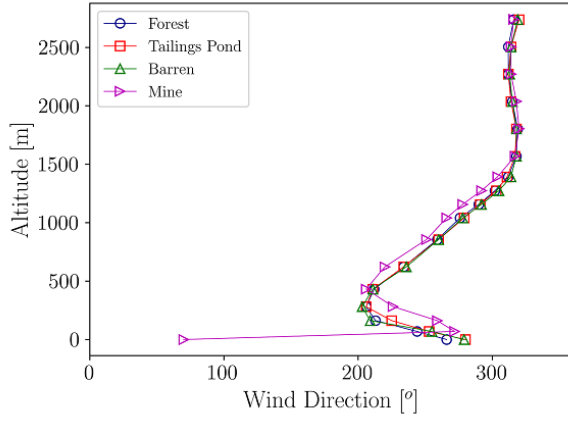


(e) 2D Ultrasonic 3

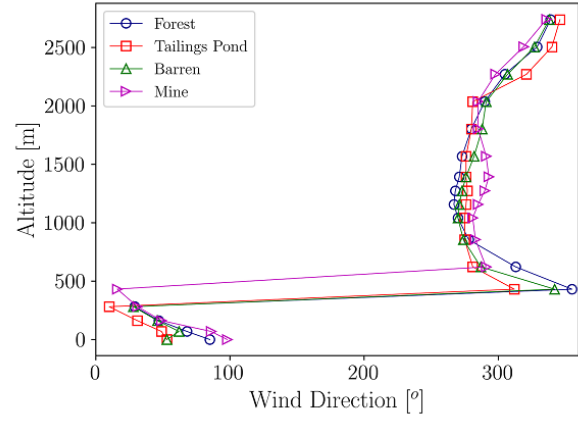


(f) WBEA Station

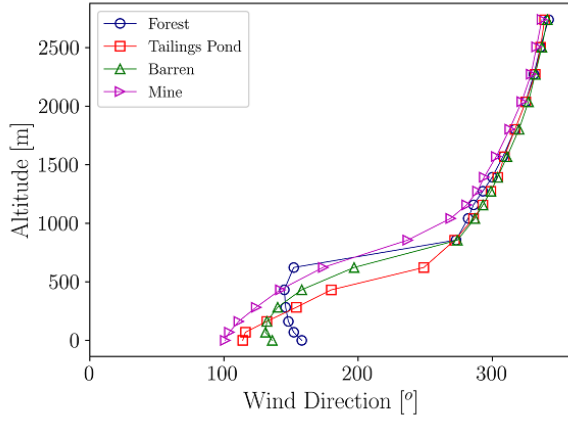
Figure 4.11: Effects of topography and land use changes on bias of wind direction at 10 m elevations from the surface showing the median values of wind direction bias for each hour.



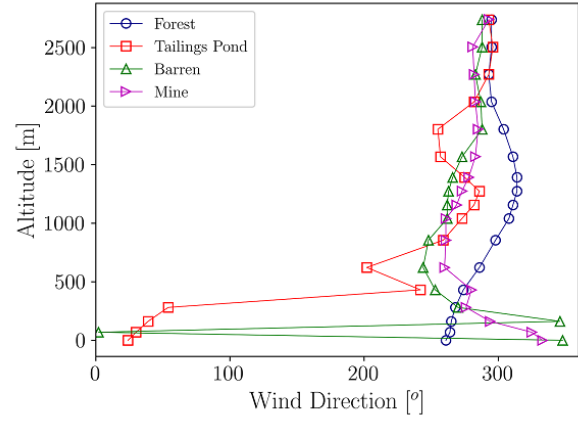
(a) Wind direction at 0200 MST



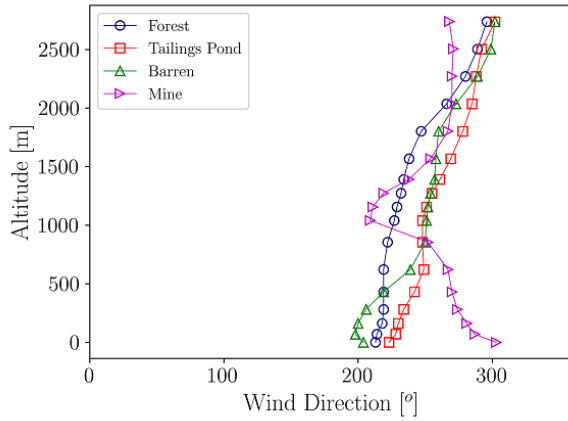
(b) Wind direction at 0600 MST



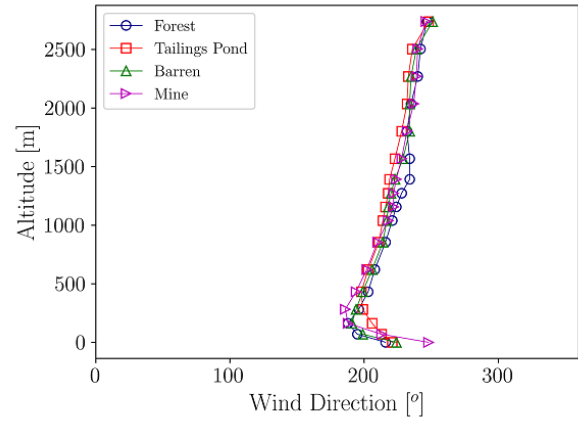
(c) Wind direction at 1000 MST



(d) Wind direction at 1400 MST



(e) Wind direction at 1800 MST



(f) Wind direction at 2200 MST

Figure 4.12: Wind direction profiles based on pressure height at different locations on May 18, 2018; times in Mountain Standard Time (MST).

modification and lakes model do not improve the predictions of temperature, in agreement with a previous study [51].

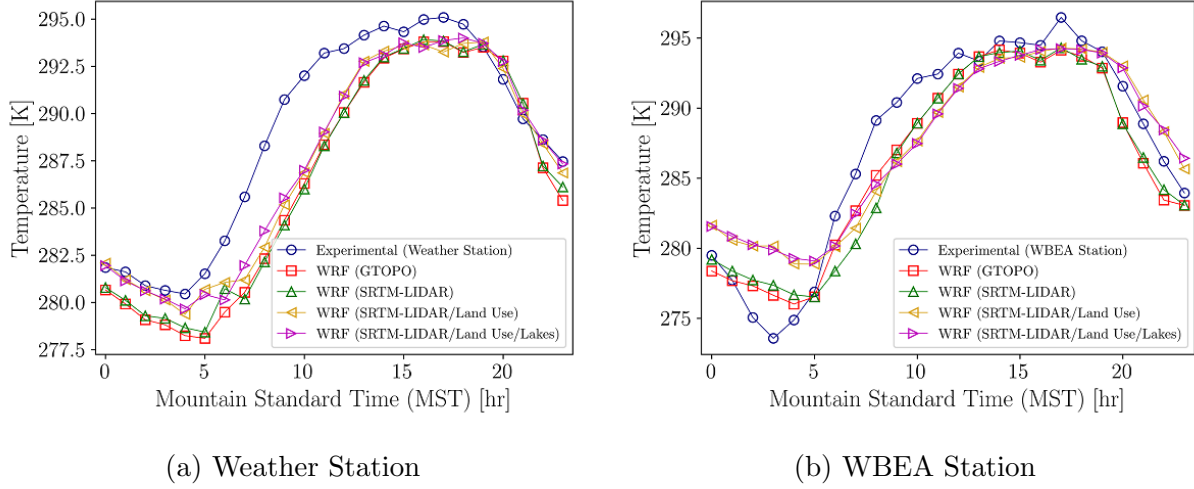
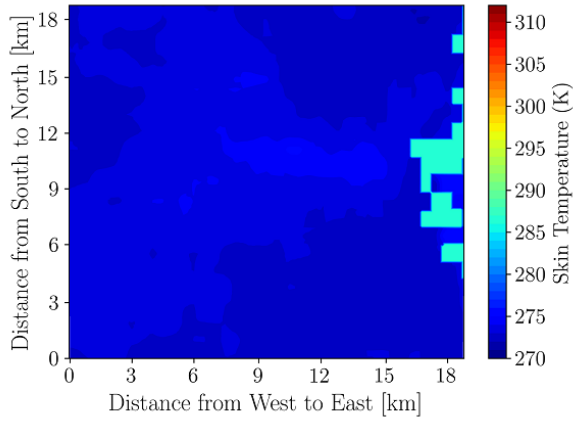
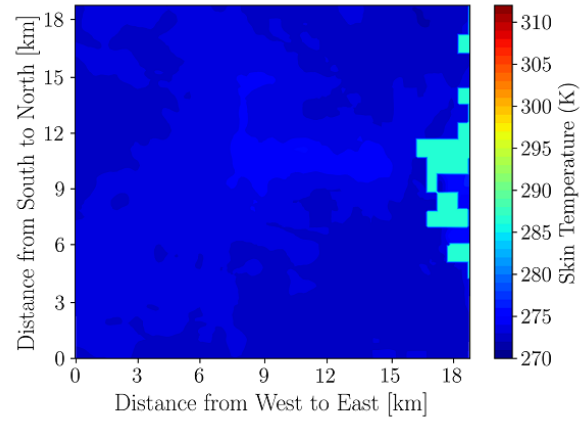


Figure 4.13: Effect of topography and land use changes on temperature at 2 m showing the median values of model and observation for each hour.

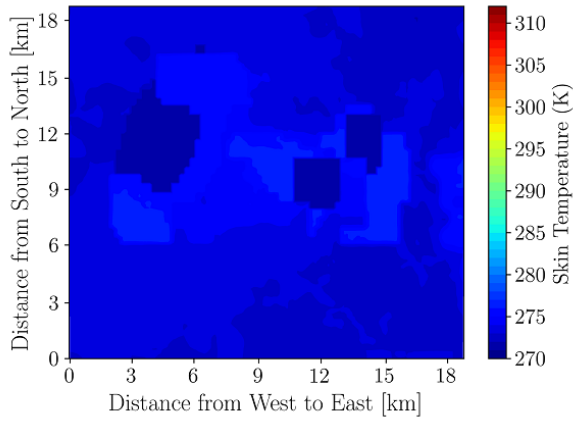
The spatial distributions of surface skin temperature and potential temperature at 2 m both in stable and unstable conditions on May 18, 2018 are shown in Figures 4.14 to 4.17 and B.5 to B.12. Examining surface level contour plots of skin temperature and temperature at 2 m provides a convenient way to understand the impact of altered topography and land use on meteorology. Two mechanisms are considered to explain the observed effects: net radiation heat transfer between the earth surface and the sky, and the heat capacity of the earth surface. The land is always emitting longwave radiation upward to the atmosphere and space, but during the day this loss of longwave radiation from the surface is more than offset by incoming shortwave (solar) radiation. Given the low heat capacity of the land, this results in warm surface temperatures during the day and cold surface temperatures during the night. For water bodies, given the higher heat capacity, the temperature variation exhibits a lower amplitude diurnal cycle. This results in warmer water temperatures at night and colder water temperatures during the day compared to the surrounding land surface temperatures [41]. These phenomena can only be simulated in WRF when accounting for SRTM-LIDAR topography, land use, and lake model [51]. Hence, over midnight (0200 MST) in thermally stable hours, the pond surface exhibits a warmer temperature compared to the surrounding areas. It is also found that when the lake model is not used, WRF overwrites the pond



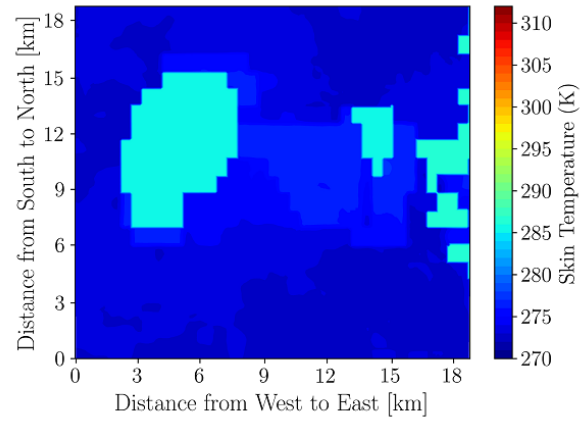
(a) GTOPO



(b) SRTM-LIDAR

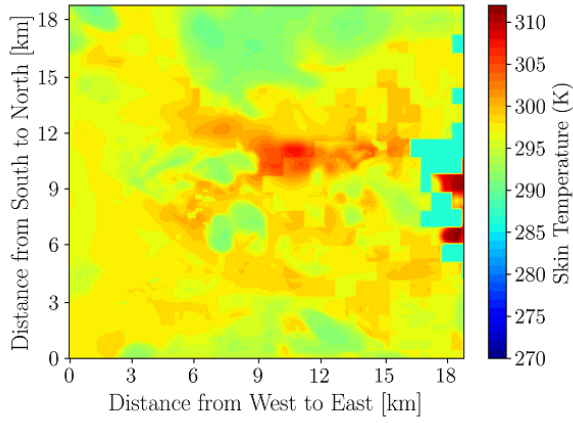


(c) SRTM-LIDAR with land use

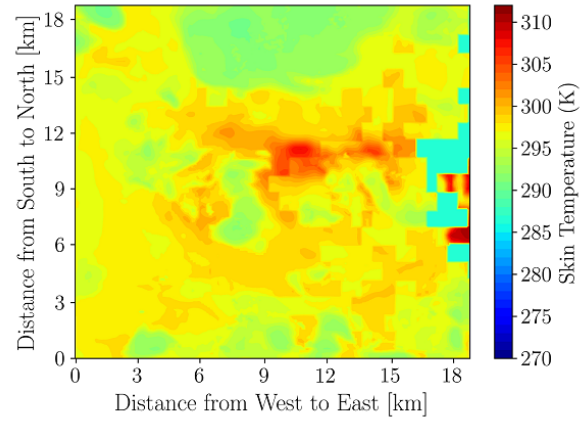


(d) SRTM-LIDAR with land use and lake model

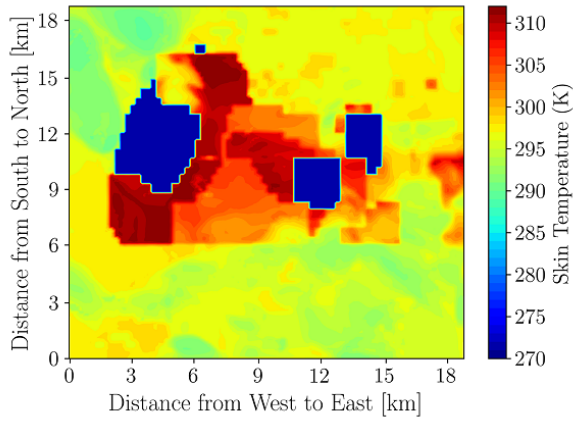
Figure 4.14: Effect of topography and land use changes on surface skin temperature at 0200 MST on May 18, 2018.



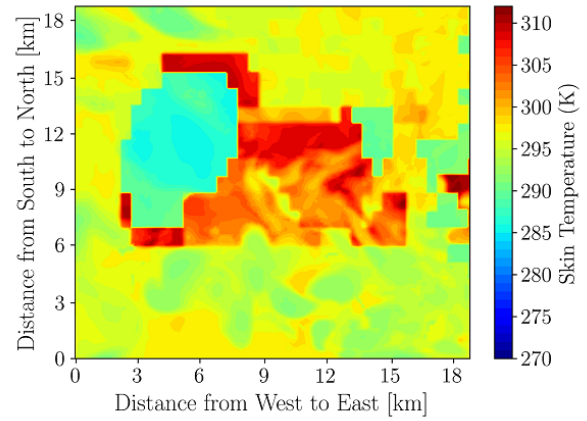
(a) GTOPO



(b) SRTM-LIDAR

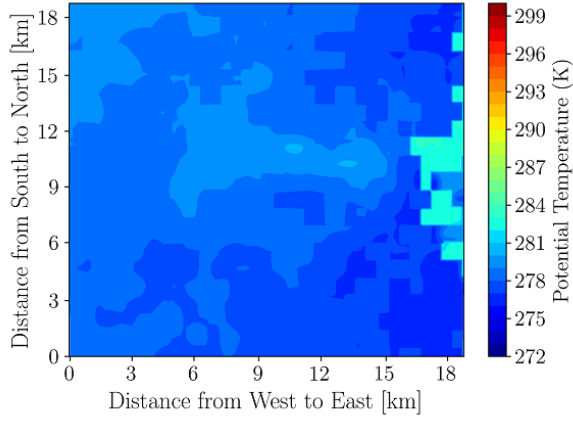


(c) SRTM-LIDAR with land use

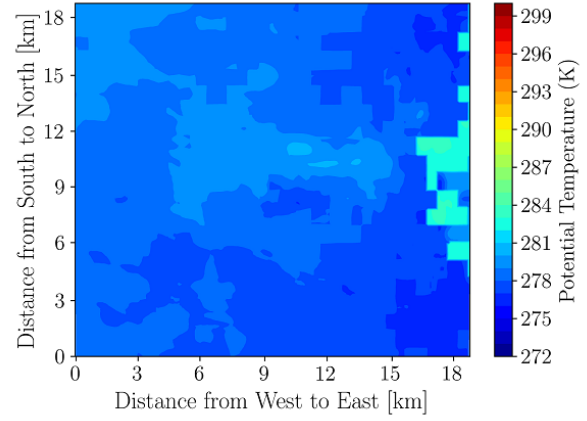


(d) SRTM-LIDAR with land use and lake model

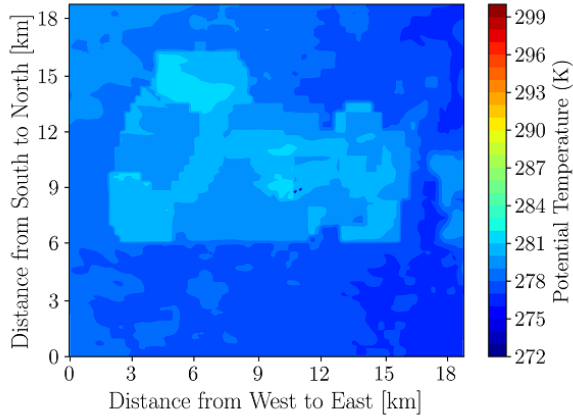
Figure 4.15: Effect of topography and land use changes on surface skin temperature at 1400 MST on May 18, 2018.



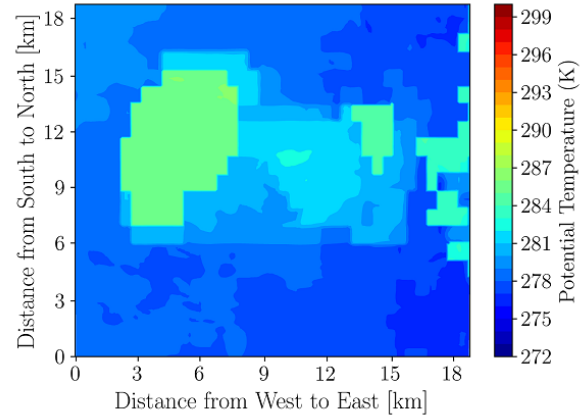
(a) GTOPO



(b) SRTM-LIDAR

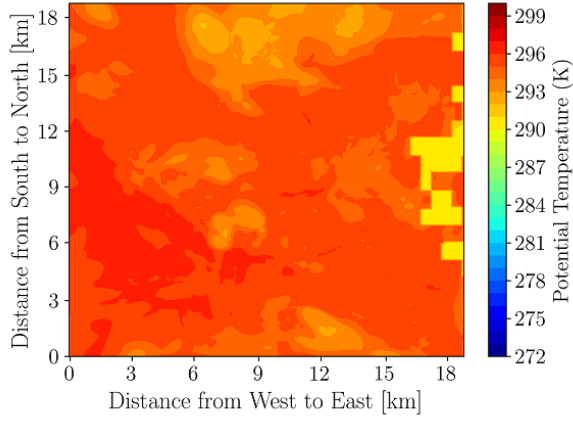


(c) SRTM-LIDAR with land use

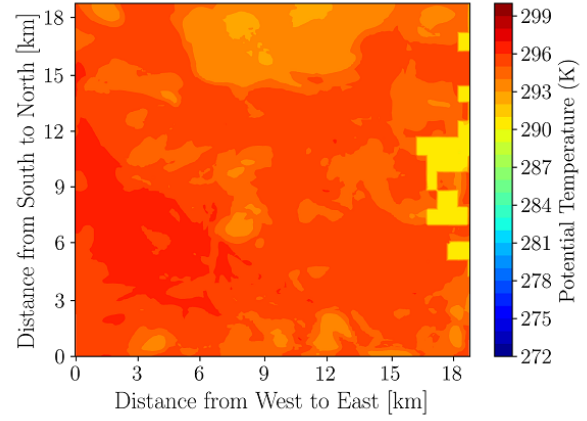


(d) SRTM-LIDAR with land use and lake model

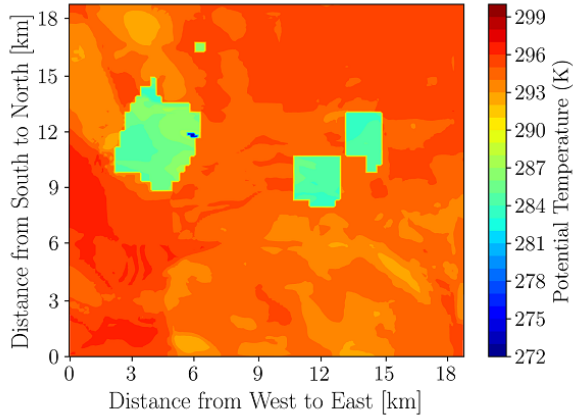
Figure 4.16: Effect of topography and land use changes on 2-m potential temperature at 0200 MST on May 18, 2018.



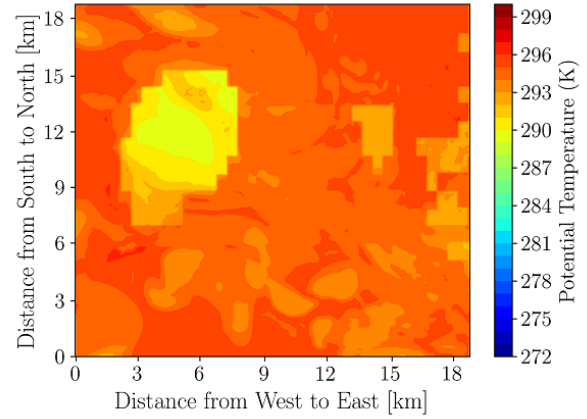
(a) GTOPO



(b) SRTM-LIDAR



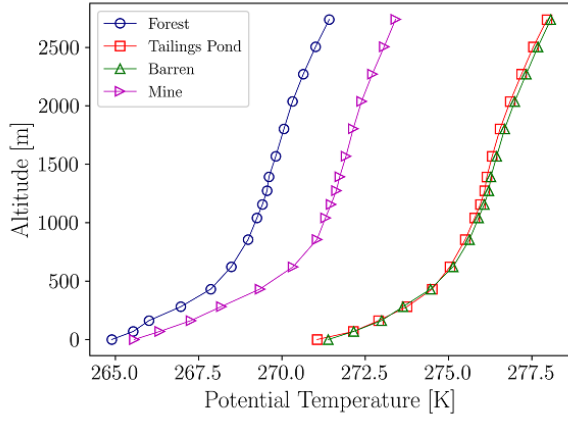
(c) SRTM-LIDAR with land use



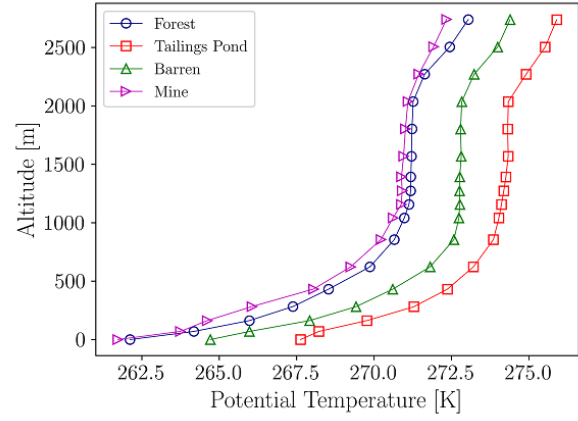
(d) SRTM-LIDAR with land use and lake model

Figure 4.17: Effect of topography and land use changes on 2-m potential temperature at 1400 MST on May 18, 2018.

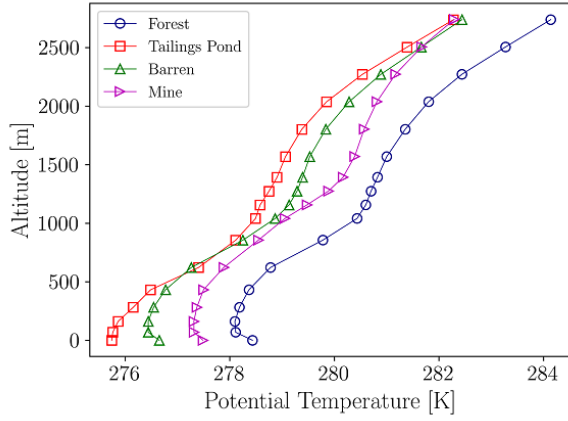




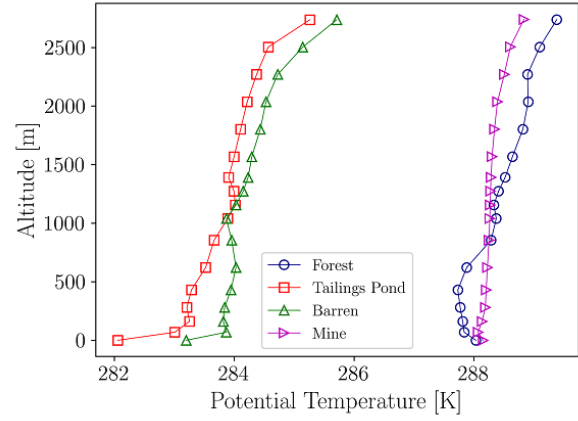
(a) Potential temperature at 0200 MST



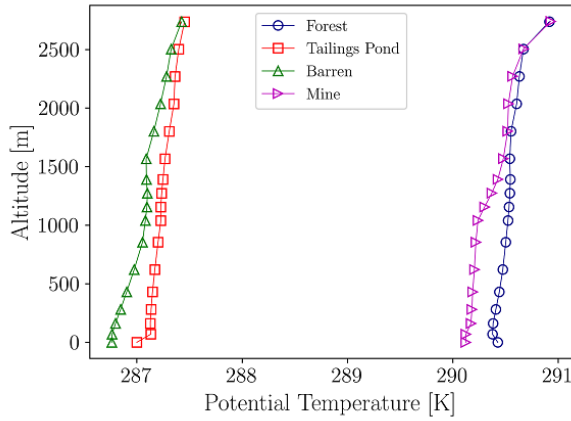
(b) Potential temperature at 0600 MST



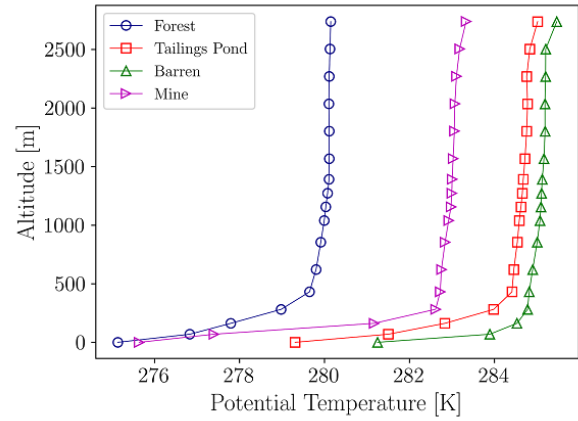
(c) Potential temperature at 1000 MST



(d) Potential temperature at 1400 MST



(e) Potential temperature at 1800 MST



(f) Potential temperature at 2200 MST

Figure 4.18: Potential temperature profiles based on pressure height at different locations on May 18, 2018; times in Mountain Standard Time (MST).

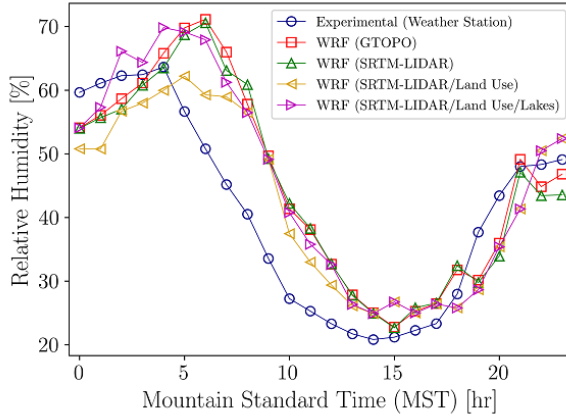
temperature according to the temperature of the nearest water body in its domains that causes lower temperatures over the pond. This is why there is no change of skin surface temperature and potential temperature when accounting for SRTM-LIDAR topography with land use but no lake model during the day and night times. The differential surface skin temperatures have profound effects on atmospheric dynamics over the mining facility. It can be noted that surface skin temperatures influence the surface-level atmospheric temperatures, particularly during the night and early morning hours, so that surface-level air temperatures are warmer over the pond compared to surrounding at 0200 MST. However, during the daytime, the forest surface level air temperatures are just as high as the mining and developed areas or at most marginally lower than the temperatures in the mining or developed areas.

Figure 4.18 shows the potential temperature profiles in domain 5 for the case accounting for SRTM-LIDAR topography, land use, and lake model at different locations on May 18, 2018. These potential temperature profiles demonstrate atmospheric stability variations as a function of diurnal time. In addition, and in support of surface-level contour plots of potential temperature, we can observe warmer columns of air over the pond compared to surrounding areas in the absence of solar heating (0200, 0600 and 2200 MST) and colder columns of air over the pond with solar heating (1000, 1400, and 1800 MST). This can also be attributed to the high heat capacity of water. The potential temperature profiles over the mine and forest also demonstrate unique complexities. Moreover, in the early morning and late night, the temperature gradient is positive at lower elevations while the surface temperature is lower than the higher elevations. In contrast, during daytime in most of the locations, the temperature gradient is negative which shows that the surface temperature is higher than the temperature at the nearest higher elevations [37].

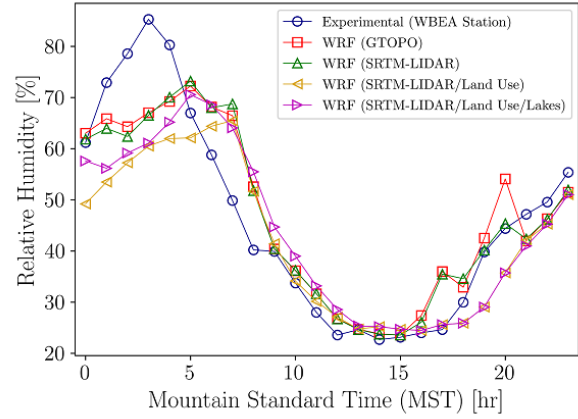
## Relative Humidity

Figure 4.19 shows the relative humidity at 2 m in domain 5 for different types of topography and land use. The difference between model and observation is larger during nighttime and early morning hours than the daytime.

Figures 4.20, 4.21 and B.13 to B.16 show the spatial distributions of relative humidity during stable and unstable conditions. There are no major differences between GTOPO and SRTM-LIDAR cases although accounting for land use modifications and lake model in WRF results in changes of relative humidity over the pond. When the lake model is enabled, the relative humidity over the pond area is lower than the surrounding areas at nighttime and early morning hours because the pond is warmer than the surrounding areas. However,



(a) Weather Station



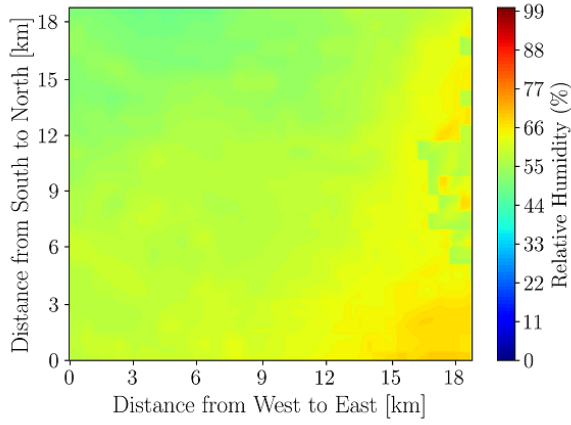
(b) WBEA Station

Figure 4.19: Effect of topography and land use changes on relative humidity at 2 m showing the median values of model and observation for each hour.

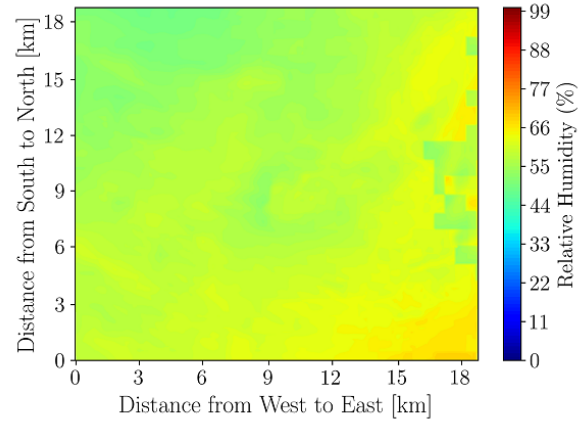
at daytime, the relative humidity over the pond area is higher than the surrounding areas because the pond is colder than the surrounding areas. The relative humidity is also inversely related to air temperature. For the same mixing ratio of water vapor in the air, the relative humidity at lower temperatures is higher because cold air can hold less water vapor before it is saturated. On the other hand, for the same mixing ratio of water vapor in the air, the relative humidity at higher temperature is lower because warm air can hold more water vapor before it is saturated.

### Sensible Vertical Turbulent Heat Flux

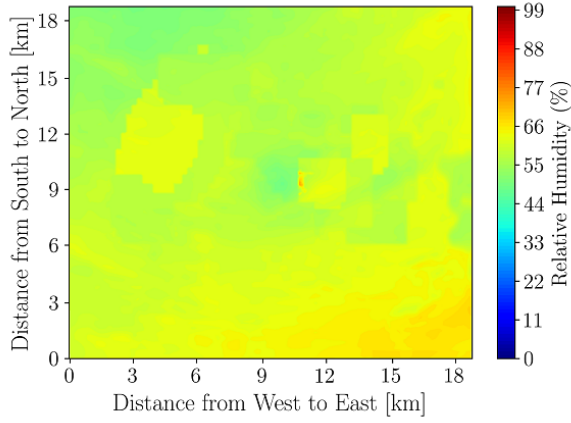
The effect of topography and land use on sensible heat flux at 10 m is shown in Figure 4.22. One of the major advantages of using land use modifications and lakes model is to predict the sensible vertical turbulent heat fluxes more accurately with errors reduced by 3.84 to 47.4  $\text{W m}^{-2}$  ( $\alpha = 0.1$ ). It is clearly identified that land use and lake model can predict the heat flux closer to the experimental measurements. As heat flux is mostly dependent on the land use classifications, WRF cases based on GTOPO and SRTM-LIDAR topographies without land use implementation and lake model exhibit large biases for heat flux with respect to the experimental measurements. WRF also shows less temporal variations of sensible vertical turbulent heat flux during the nighttime stable condition with a small difference between model and observation. On the other hand, WRF shows higher temporal variations



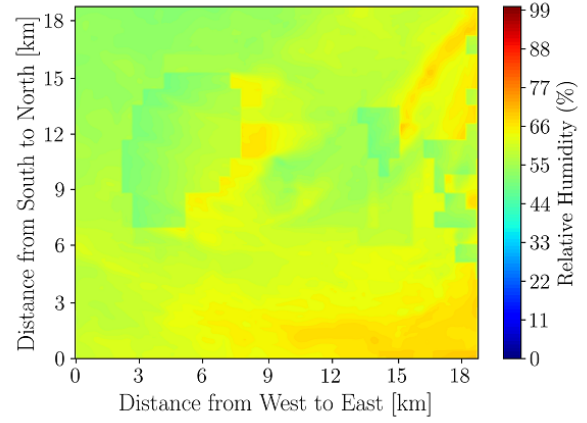
(a) GTOPO



(b) SRTM-LIDAR

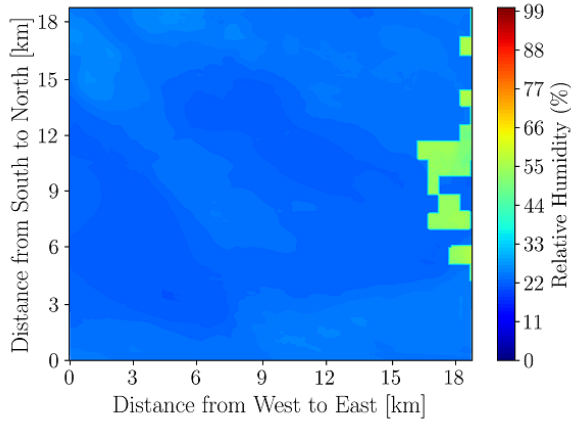


(c) SRTM-LIDAR with land use

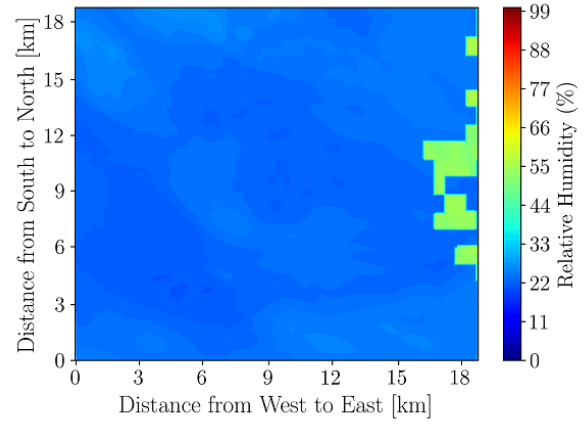


(d) SRTM-LIDAR with land use and lake model

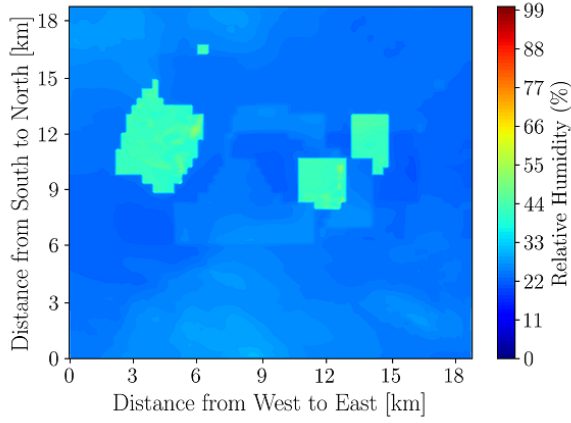
Figure 4.20: Effect of topography and land use changes on 2-m relative Humidity at 0200 MST on May 18, 2018.



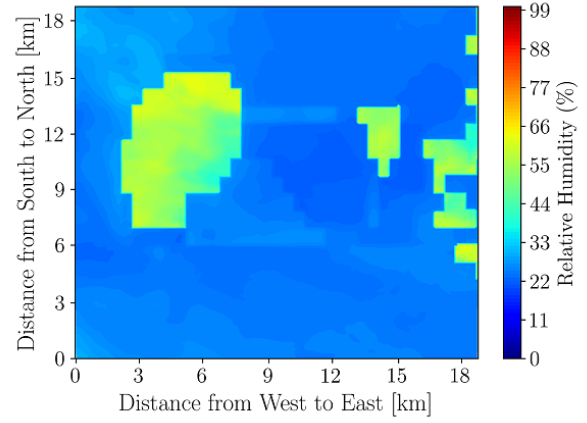
(a) GTOPO



(b) SRTM-LIDAR

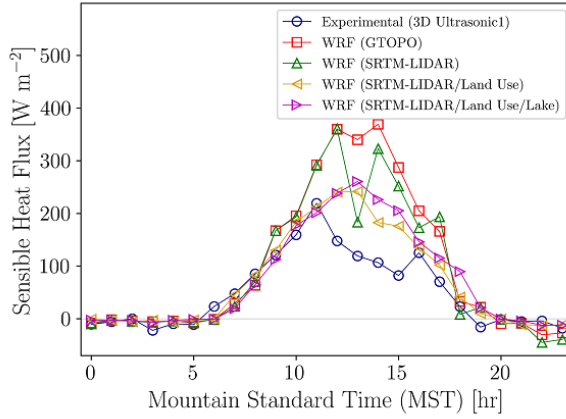


(c) SRTM-LIDAR with land use

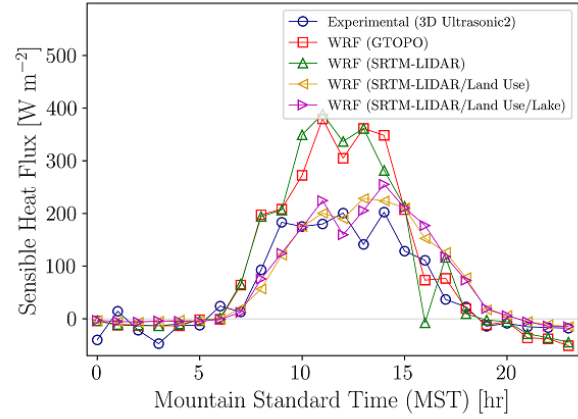


(d) SRTM-LIDAR with land use and lake model

Figure 4.21: Effect of topography and land use changes on 2-m relative Humidity at 1400 MST on May 18, 2018.



(a) 3D Ultrasonic 1



(b) 3D Ultrasonic 2

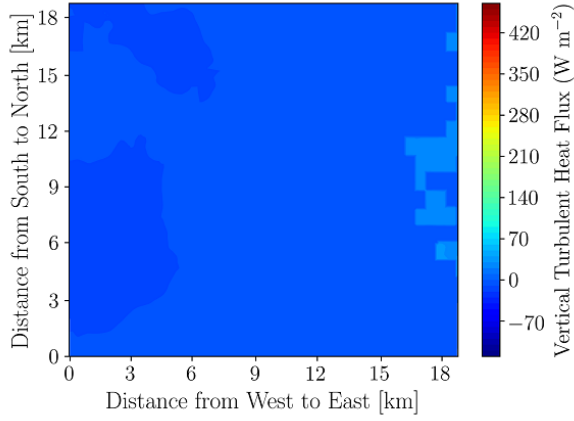
Figure 4.22: Effect of topography and land use changes on sensible vertical turbulent heat flux at 10 m.

of sensible vertical turbulent heat flux during the daytime unstable condition with a large difference between model and observation.

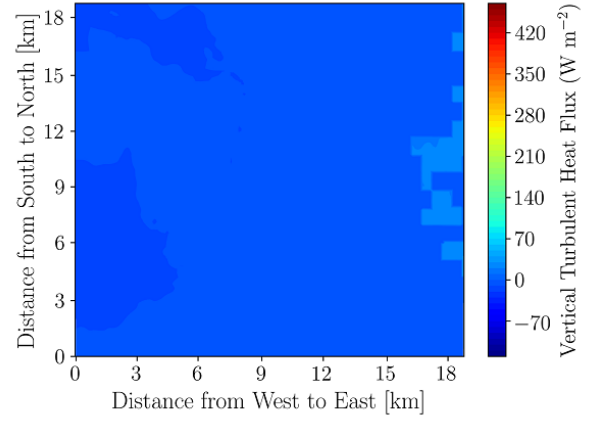
Figures 4.23, 4.24 and B.17 to B.20 show the spatial distributions of sensible vertical turbulent heat flux during the thermally stable and unstable conditions. There are no major differences between GTOPO and SRTM-LIDAR cases both at 0200 and 1400 MST. However, when the lake model is enabled, the sensible vertical turbulent heat flux over the pond area is higher than the surrounding areas at 0200 and 0600 MST as the water body is usually warmer than the surrounding areas that increases the heat flux over the pond. On the other hand, at daytime water bodies are colder compared to the surrounding areas resulting in lower heat flux over the pond compared to the surrounding areas.

### 4.3.2 Effects of Grid Resolution and Planetary Boundary Layer Parameterization

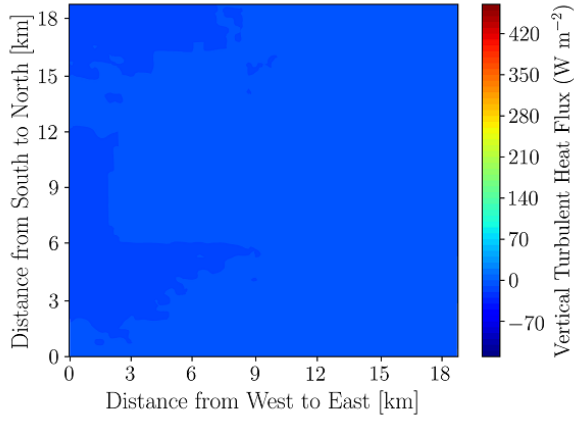
The effects of horizontal and vertical grid resolutions and PBL parameterization are discussed in this section. Based on the results in the previous section, only the case with SRTM-LIDAR topography, land use change, and lakes model are considered for this sensitivity study because overall lower errors were observed for these configurations. The bias and RMSE of different properties are shown in Table 4.5. The two-sample test was also employed in this case but



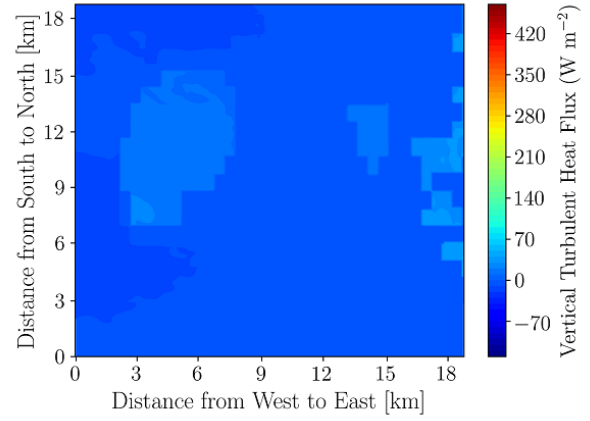
(a) GTOPO



(b) SRTM-LIDAR

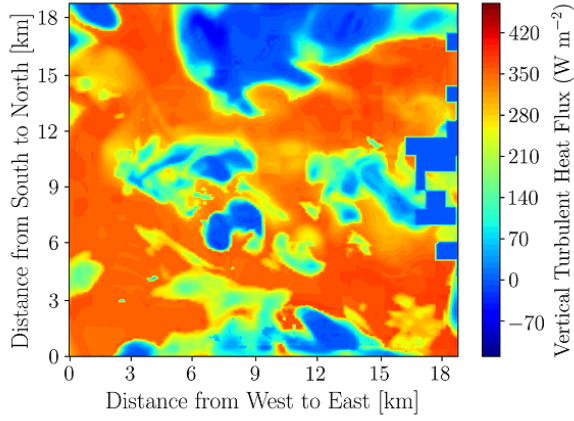


(c) SRTM-LIDAR with land use

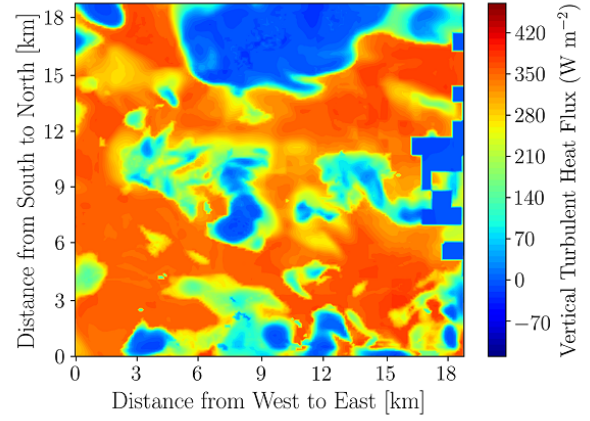


(d) SRTM-LIDAR with land use and lake model

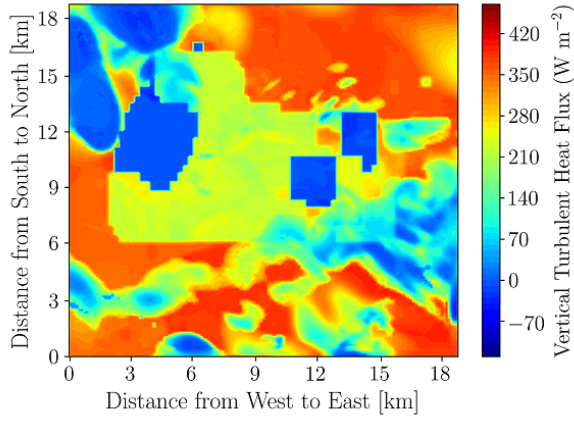
Figure 4.23: Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 0200 MST on May 18, 2018.



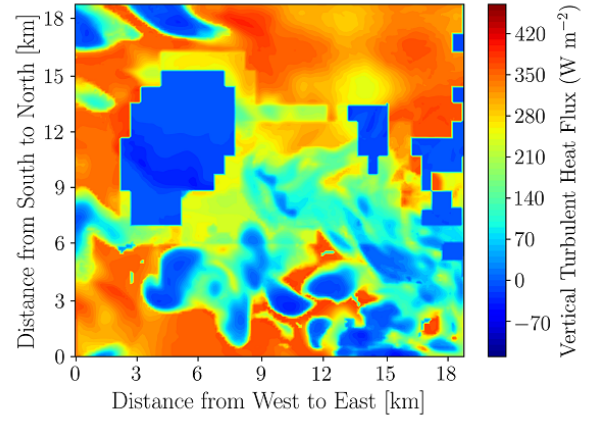
(a) GTOPO



(b) SRTM-LIDAR



(c) SRTM-LIDAR with land use



(d) SRTM-LIDAR with land use and lake model

Figure 4.24: Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 1400 MST on May 18, 2018.



a table is not provided for brevity.

## Wind Velocity

Figures 4.25 and 4.26 show the horizontal wind velocity in domain 5 near surface level for different types of grid resolutions and PBL parameterizations. For a given grid resolution, the PBL parameterization does not have a notable effect on the overall bias and RMSE of horizontal wind velocity. The high horizontal and mid/high vertical grid resolutions marginally improve the predictability of horizontal wind velocity at 10 m [24]. However, there is a major impact of grid resolutions in predicting horizontal wind velocity as measured by TANAB. Increasing the horizontal and vertical resolutions can reduce errors up to 1.52 to 3.40 m s<sup>-1</sup> ( $\alpha = 0.1$ ) in wind speed predictions near the surface up to 200 m altitude. Similar findings were reported in other literatures [30, 78, 107]. Hence, the horizontal grid resolutions should be at least 0.12 km in domain 5 with 90 vertical levels to obtain a lower bias of horizontal wind velocity as compared against TANAB.

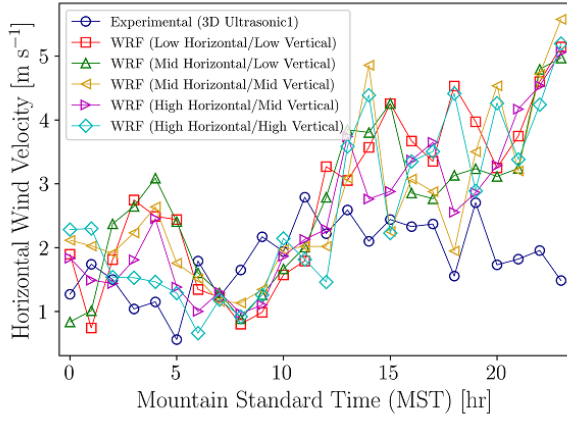
In the case of PBL parameterization, high horizontal and mid vertical grid resolution cases are chosen for plotting Figure 4.26 showing the hourly median values of wind speed for the selected three days of simulations. Both Yonsei University Scheme (YSU) and Large-Eddy Simulations (LES) have similar performance in predicting the diurnal variation of surface-level wind speed although YSU has better numerical convergence performance since LES cases occasionally diverge.

## Wind Direction

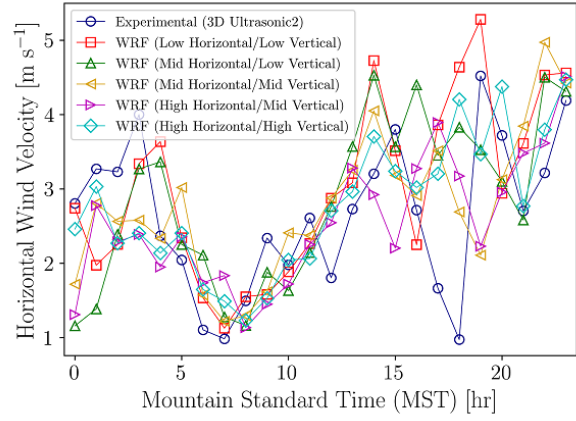
The effects of grid resolution and PBL parameterization on wind direction bias in domain 5 are shown in Figures 4.27 and 4.28. The changes in grid resolutions do not have a major impact on the overall bias and RMSE of wind direction. However, as a sonic instrument, SODAR has some limitations to measure the actual wind direction, as well as WRF may also struggle due to the low vertical resolution at higher altitudes. This study has also shown that the model-observation difference for wind direction is higher in thermally stable conditions than unstable conditions due to the inaccuracy of PBL parameterizations and difficulty to measure wind direction at low wind speeds [7].

Table 4.5: Effects of grid resolution and planetary boundary layer parameterization on bias and RMSE of different atmospheric properties for model-observation comparison.

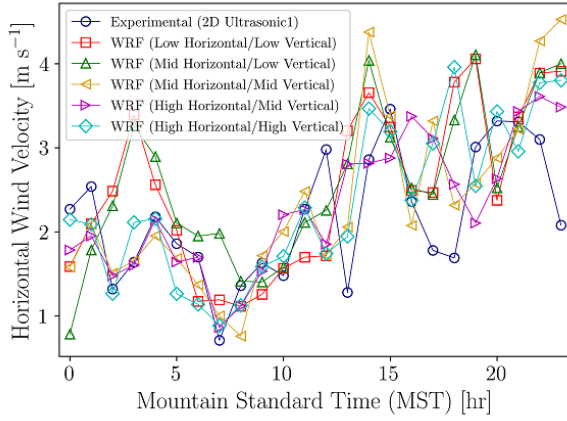
Parameter	Errors									
	Low horizontal		Mid horizontal		Mid horizontal		High horizontal		High horizontal	
	Low vertical	Low vertical	Low vertical	Low vertical	Mid vertical	Mid vertical	Mid vertical	Mid vertical	High vertical	High vertical
	YSU	LES	YSU	LES	YSU	LES	YSU	LES	YSU	LES
Horizontal wind velocity at 10 m ( $\text{m s}^{-1}$ )	BIAS	0.21	0.20	0.14	-0.01	0.02	0.01	-0.12	-0.05	-0.19
	RMSE	1.50	1.54	1.63	1.49	1.57	1.61	1.59	1.63	1.61
Wind direction at 10 m ( $^{\circ}$ )	BIAS	-5.04	-12.0	-0.78	-1.26	-3.39	-9.33	2.12	-4.69	-7.79
	RMSE	75.0	74.7	73.6	76.5	73.6	74.3	73.9	75.4	77.0
Horizontal wind velocity from TANAB ( $\text{m s}^{-1}$ )	BIAS	2.99	3.31	2.68	2.94	0.53	0.08	-0.21	1.47	1.46
	RMSE	4.48	4.62	4.36	4.51	1.87	1.68	1.64	3.58	3.62
Horizontal wind velocity from SODAR ( $\text{m s}^{-1}$ )	BIAS	0.67	0.80	0.63	0.52	0.55	0.50	0.01	0.49	-0.11
	RMSE	2.37	2.43	2.15	2.34	2.35	2.30	2.07	2.21	1.55
Wind direction from SODAR ( $^{\circ}$ )	BIAS	0.49	-2.50	1.67	-7.35	0.19	-9.46	-6.91	-5.75	-3.52
	RMSE	75.0	73.9	69.6	68.8	76.8	74.1	74.4	74.5	81.3
Temperature at 2 m ( $^{\circ}\text{C}$ )	BIAS	0.17	0.26	0.39	0.50	0.17	0.26	0.28	0.18	0.15
	RMSE	2.47	2.48	2.33	2.34	2.47	2.51	2.57	2.42	2.56
Potential temperature gradient from TANAB ( $\text{K m}^{-1}$ )	BIAS	0.015	0.015	0.015	0.015	0.015	0.016	0.017	0.015	0.015
	RMSE	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.030
Relative humidity at 2 m (%)	BIAS	-1.46	-1.01	-2.39	-2.16	-1.05	-0.66	-1.84	-1.29	-1.15
	RMSE	9.36	9.43	8.48	8.95	9.67	9.51	9.45	9.13	10.1
Sensible heat flux at 10 m ( $\text{W m}^{-2}$ )	BIAS	22.0	20.0	17.5	18.0	14.8	17.3	17.9	15.2	11.8
	RMSE	61.2	59.3	57.2	56.6	56.5	58.8	59.5	60.2	63.0



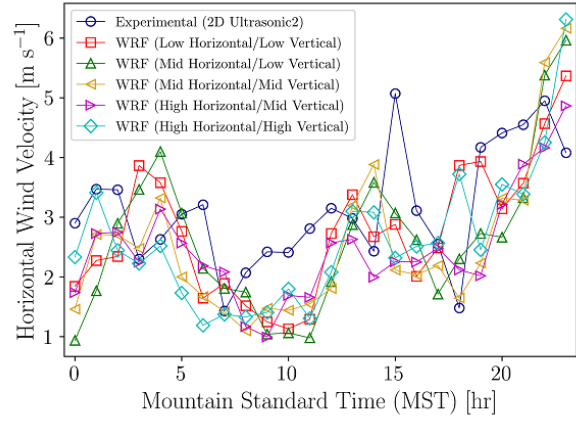
(a) 3D Ultrasonic 1



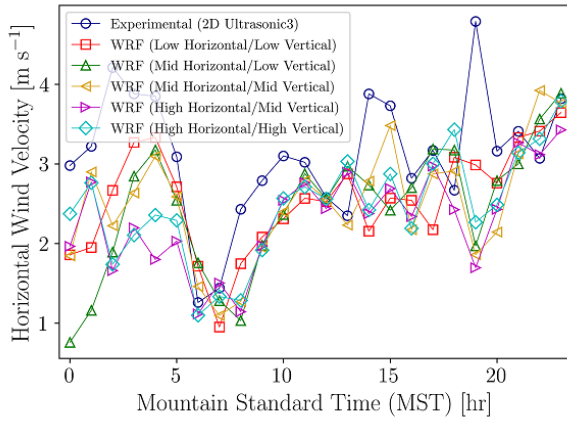
(b) 3D Ultrasonic 2



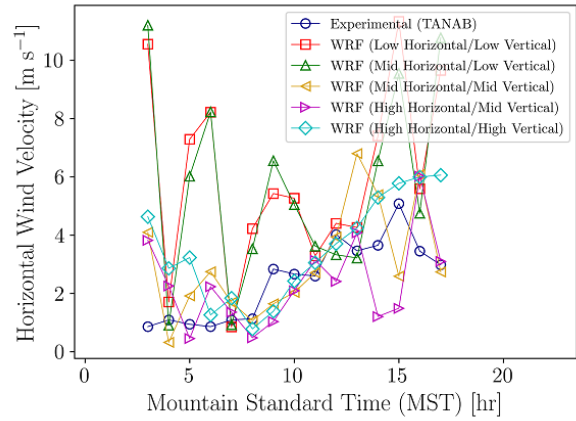
(c) 2D Ultrasonic 1



(d) 2D Ultrasonic 2

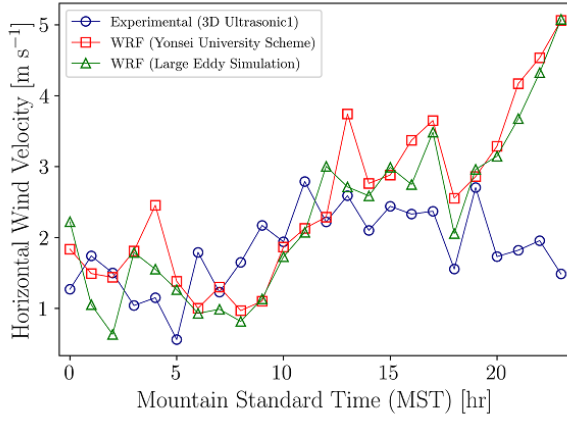


(e) 2D Ultrasonic 3

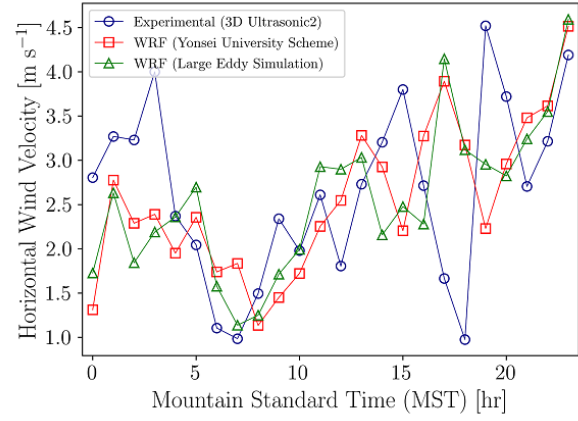


(f) TANAB

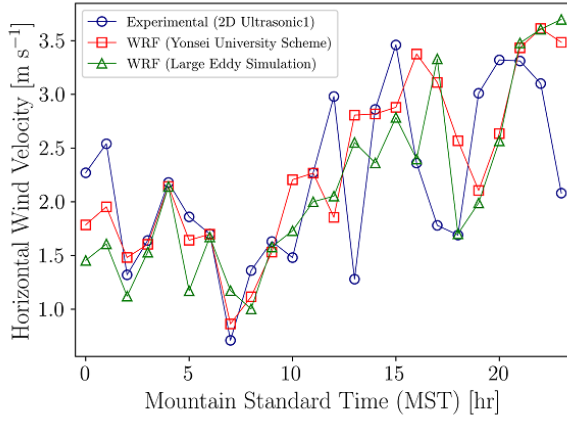
Figure 4.25: Effect of grid resolution on horizontal wind velocity near surface level showing the median values of model and observation for each hour.



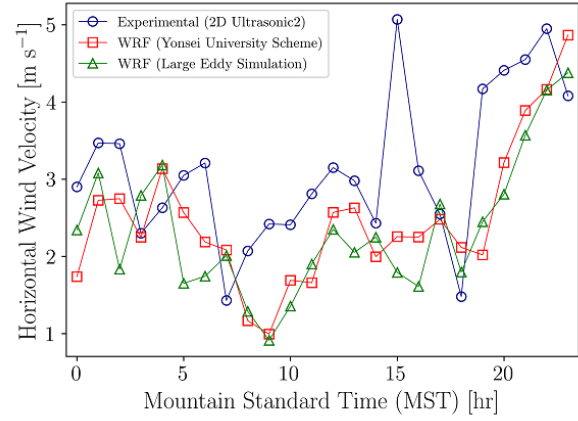
(a) 3D Ultrasonic 1



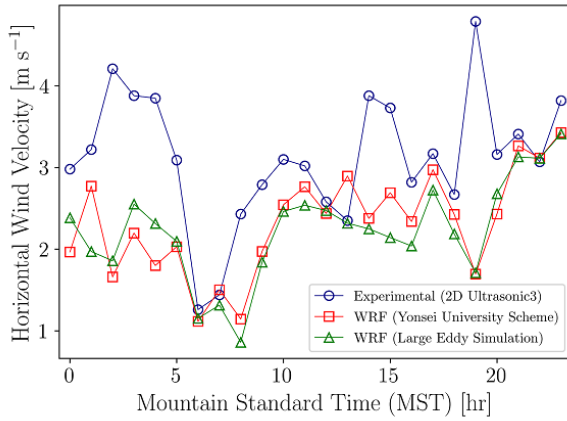
(b) 3D Ultrasonic 2



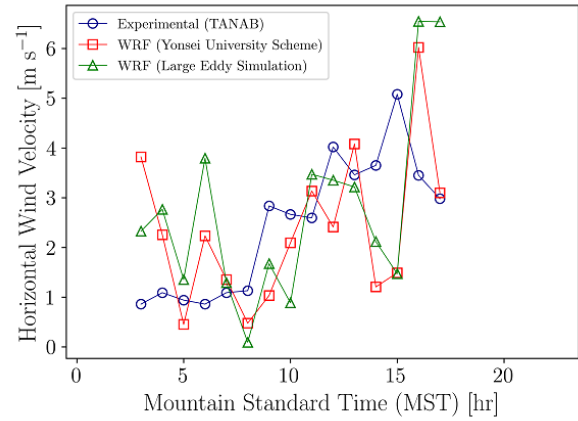
(c) 2D Ultrasonic 1



(d) 2D Ultrasonic 2

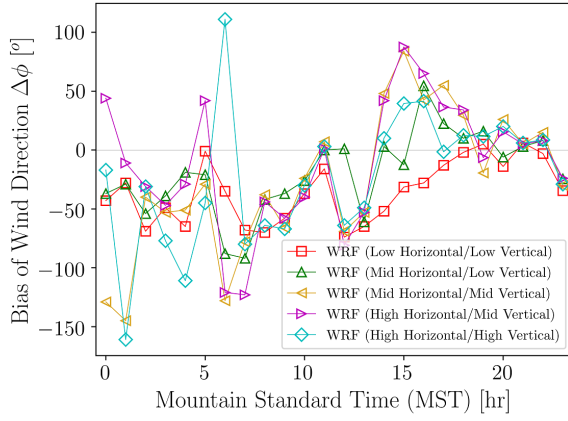


(e) 2D Ultrasonic 3

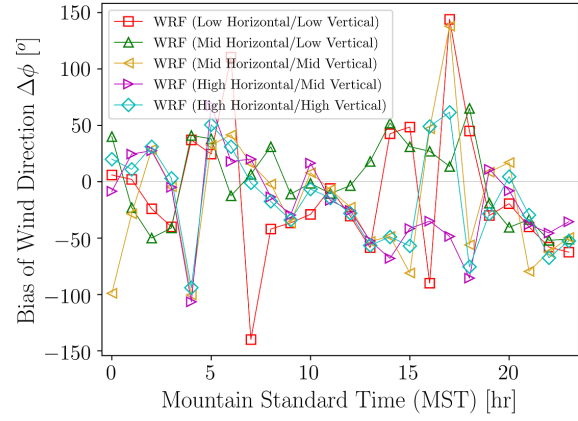


(f) TANAB

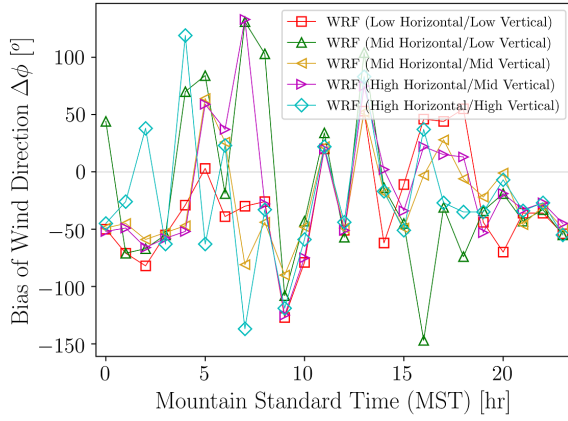
Figure 4.26: Effect of PBL parameterization on horizontal wind velocity near surface level showing the median values of model and observation for each hour.



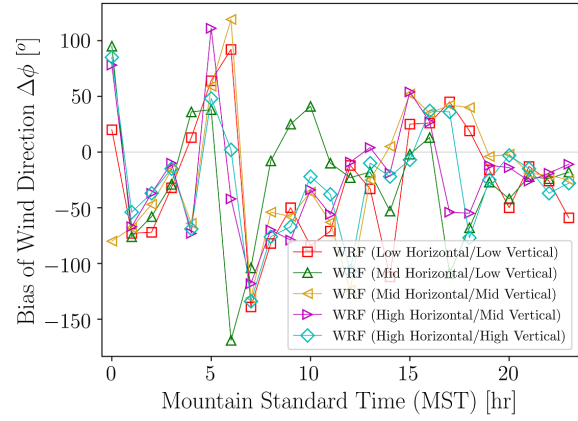
(a) 3D Ultrasonic 1



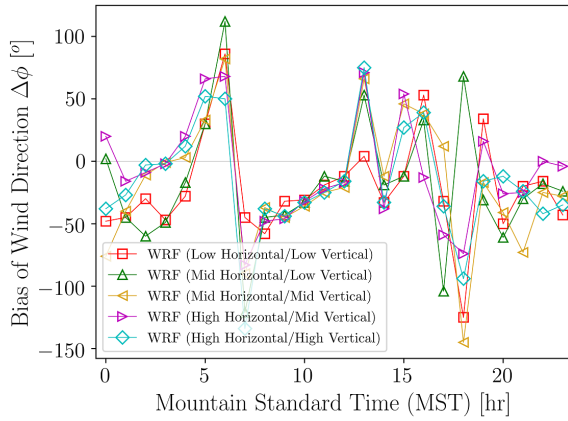
(b) 3D Ultrasonic 2



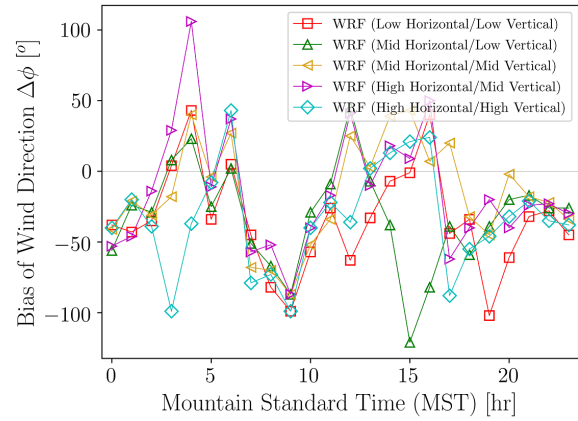
(c) 2D Ultrasonic 1



(d) 2D Ultrasonic 2

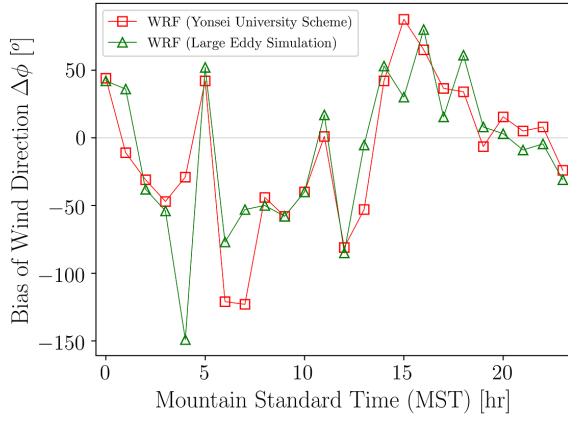


(e) 2D Ultrasonic 3

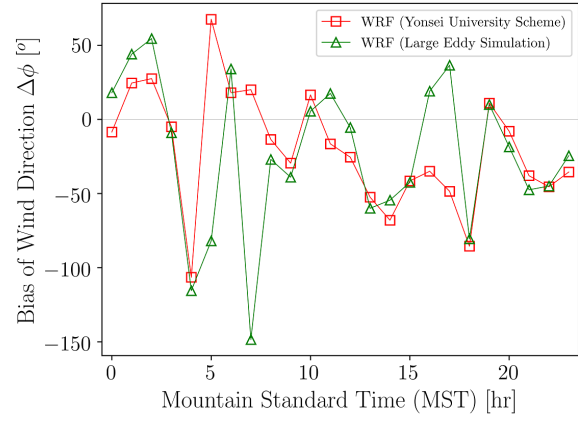


(f) WBEA Station

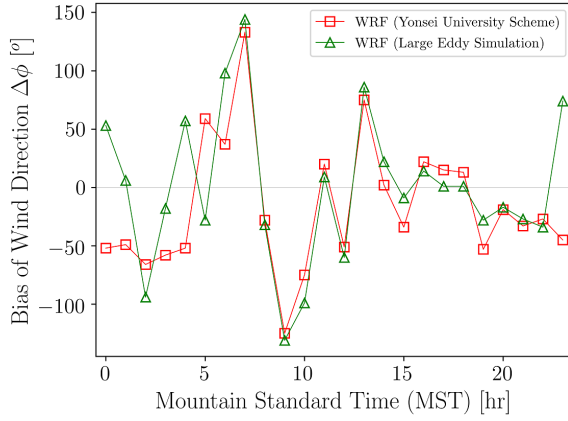
Figure 4.27: Effect of grid resolution on wind direction at 10 m showing the median values of wind direction bias for each hour.



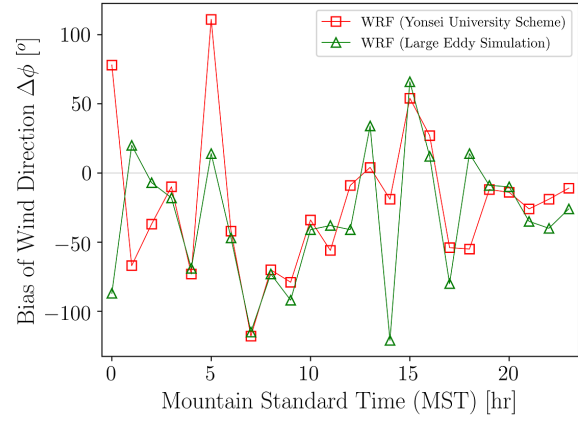
(a) 3D Ultrasonic 1



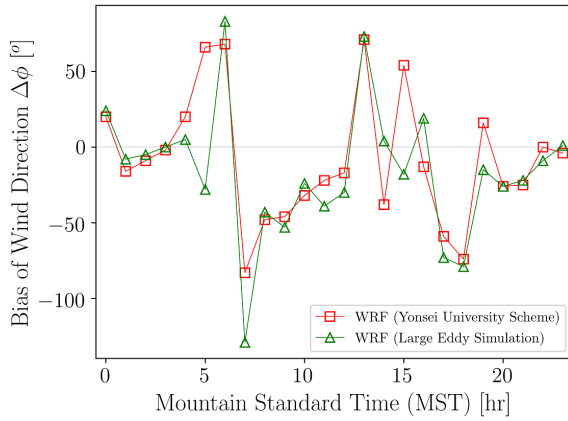
(b) 3D Ultrasonic 2



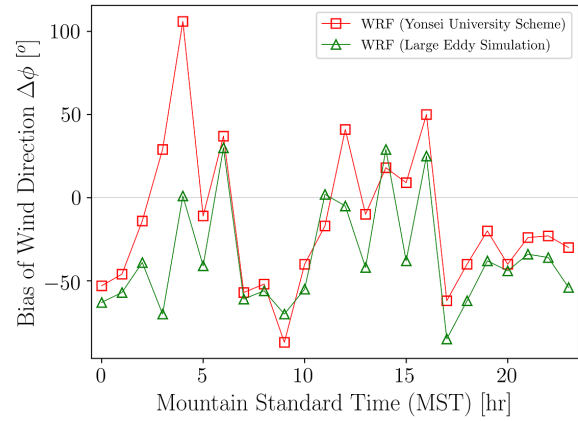
(c) 2D Ultrasonic 1



(d) 2D Ultrasonic 2



(e) 2D Ultrasonic 3



(f) WBEA Station

Figure 4.28: Effect of PBL parameterization on wind direction at 10 m showing the median values of wind direction bias for each hour.

## Temperature

Figures 4.29 and 4.30 show the temperature at 2 m in domain 5 for different types of grid resolutions and PBL parameterizations. From Table 4.5 and Figures 4.29 and 4.30 it is observed that grid resolution and PBL parameterization do not show any major impact on predicting temperature near the surface level. Moreover, the overall bias is also similar for different grid resolutions and PBL parameterizations.

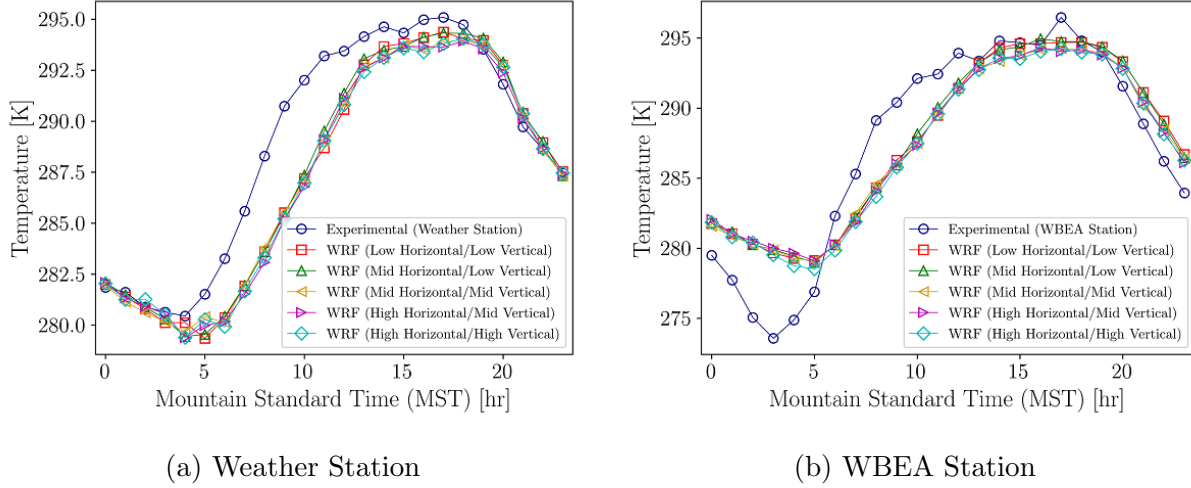


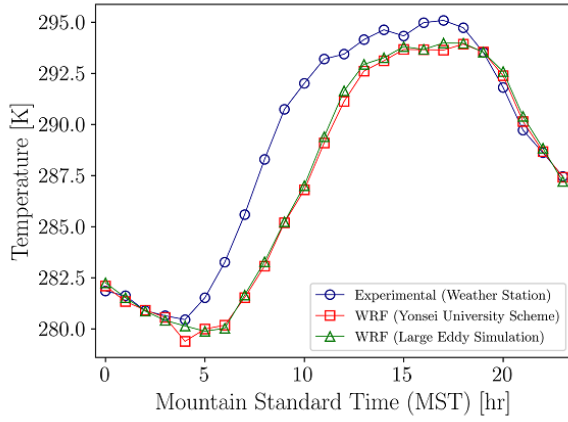
Figure 4.29: Effect of grid resolution on temperature at 2 m showing the median values of model and observation for each hour.

## Relative Humidity

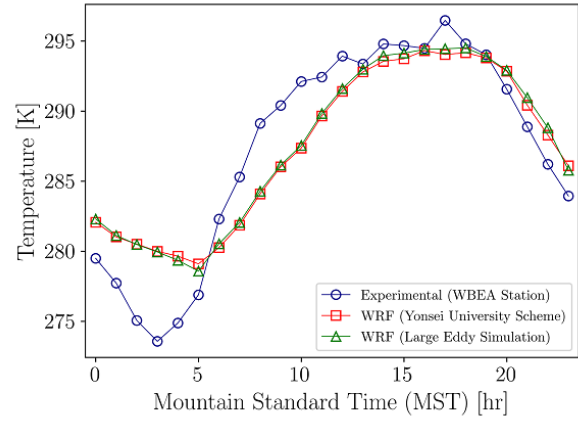
Figures 4.31 and 4.32 show the relative humidity at 2 m in domain 5 for different types of grid resolution and PBL parameterization. From Table 4.5 and these figures, it is observed that grid resolution and PBL parameterization do not show significant impact on predicting the relative humidity near the surface level. Moreover, the overall bias is also similar for different grid resolutions and PBL turbulence parameterizations.

## Sensible Vertical Turbulent Heat Flux

The effect of grid resolution and PBL parameterization on sensible heat flux at 10 m is shown in Figures 4.33 and 4.34. From this study, it is observed that there is an improvement of sensible vertical turbulent heat flux prediction by increasing resolutions beyond mid horizontal

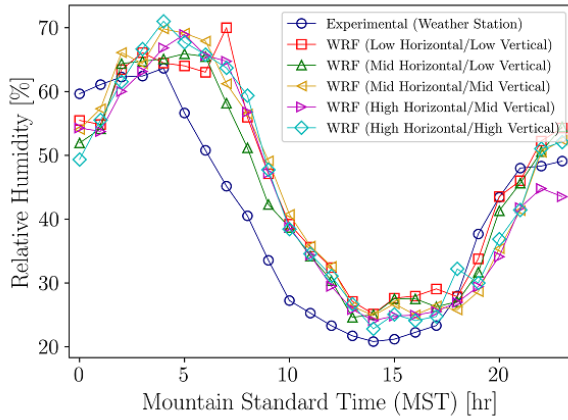


(a) Weather Station

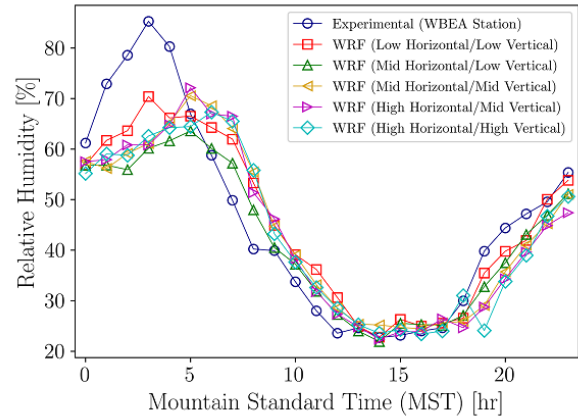


(b) WBEA Station

Figure 4.30: Effect of PBL parameterization on temperature at 2 m showing the median values of model and observation for each hour.



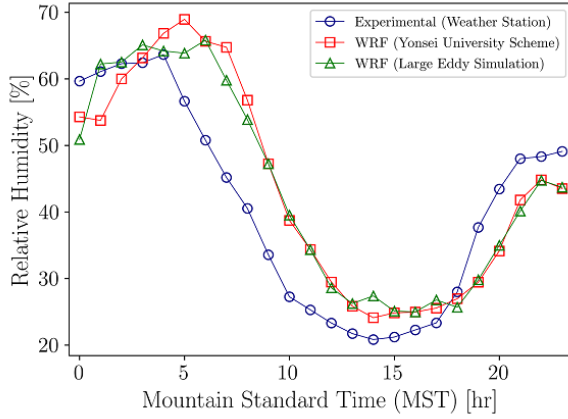
(a) Weather Station



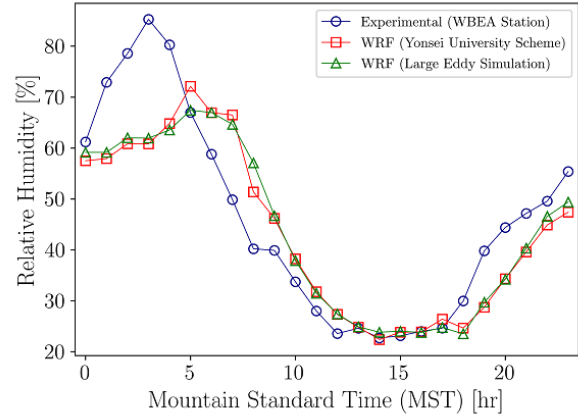
(b) WBEA Station

Figure 4.31: Effect of grid resolution on relative humidity at 2 m showing the median values of model and observation for each hour.





(a) Weather Station



(b) WBEA Station

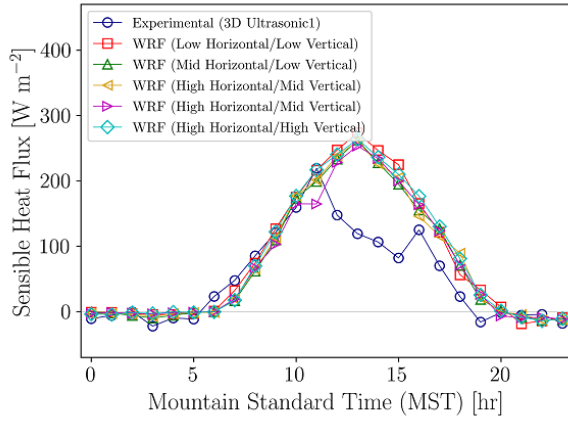
Figure 4.32: Effect of PBL parameterization on relative humidity at 2 m showing the median values of model and observation for each hour.

and mid vertical grid resolutions resulting  $-8.89$  to  $23.3 \text{ W m}^{-2}$  ( $\alpha = 0.1$ ) less bias. Similar findings were reported in other studies [128]. On the other hand, LES predicts the sensible vertical heat flux more accurately than YSU WRF simulations by  $-11.4$  to  $21.1 \text{ W m}^{-2}$  ( $\alpha = 0.1$ ) less bias. The time variations of heat flux at convective hours can also be identified for both YSU WRF and WRF-LES simulations. Hence, LES WRF simulations can be utilized to improve the predictions of sensible vertical turbulent heat flux both in thermally stable and unstable conditions.

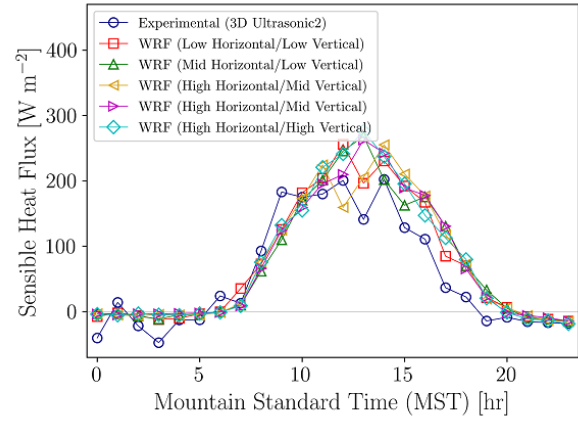
### 4.3.3 GFS Forecast versus Reanalysis Initial and Boundary Conditions

The Global Forecasting System (GFS) Forecast and Reanalysis datasets have been used in the WRF simulations to observe if they make any difference in the predictions.<sup>10</sup> GFS Forecast provides initial and time-varying boundary conditions to WRF at 3-hour time resolution and  $0.5^\circ$  spatial resolution with the global WRF model run into the future [39, 106, 126]. On the other hand, GFS Reanalysis datasets are based on model-observation interpolations and data assimilations to represent the atmospheric condition in a past state. In Reanalysis, statistical methods of the variability of measurements are performed to provide a diverse,

<sup>10</sup><https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>

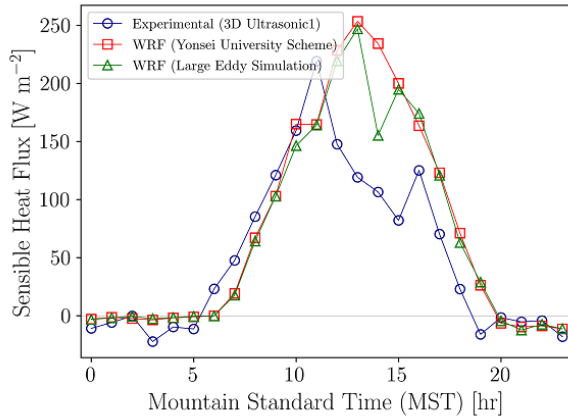


(a) 3D Ultrasonic 1

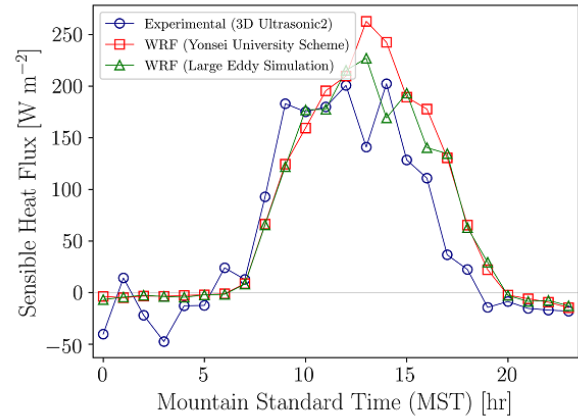


(b) 3D Ultrasonic 2

Figure 4.33: Effect of grid resolution on sensible vertical turbulent heat flux at 10 m showing the median values of model and observation for each hour.



(a) 3D Ultrasonic 1



(b) 3D Ultrasonic 2

Figure 4.34: Effect of PBL parameterization on sensible vertical turbulent heat flux at 10 m showing the median values of model and observation for each hour.

spatially complete, and comprehensible record of the meteorological properties at 6-hour time resolution and  $0.5^\circ$  spatial resolution [47, 88].

For this investigation, mid horizontal/vertical resolution WRF simulations with modified topography, land use, and lake model are run to minimize computational expense. Moreover, the Yonsei University (YSU) scheme is chosen for PBL parameterization because LES cases sometimes diverge during running the WRF simulations on the mid horizontal/vertical resolution. The comparison between GFS Forecast and Reanalysis for WRF simulations in domain 5 and the experimental observations on May 18, 24 and 30, 2018 is performed in this study. The bias and RMSE of different properties are shown in Table 4.6.

Table 4.6: Effects of GFS datasets on WRF simulations; table shows the bias and RMSE of different atmospheric properties between the WRF model and field observations.

Parameter	Errors	Forecast	Reanalysis
Horizontal wind velocity at 10 m ( $\text{m s}^{-1}$ )	BIAS	0.02	0.24
	RMSE	1.57	1.68
Wind direction at 10 m ( $^\circ$ )	BIAS	-3.39	-8.62
	RMSE	73.6	69.1
Horizontal wind velocity from TANAB ( $\text{m s}^{-1}$ )	BIAS	0.52	0.79
	RMSE	1.87	2.04
Horizontal wind velocity from SODAR ( $\text{m s}^{-1}$ )	BIAS	0.55	0.94
	RMSE	2.35	2.80
Wind direction from SODAR ( $^\circ$ )	BIAS	0.19	-6.37
	RMSE	76.8	78.1
Temperature at 2 m ( $^\circ\text{C}$ )	BIAS	0.17	-0.09
	RMSE	2.47	2.32
Potential temperature gradient from TANAB ( $\text{K m}^{-1}$ )	BIAS	0.015	0.017
	RMSE	0.029	0.030
Relative humidity at 2 m (%)	BIAS	-1.05	-0.43
	RMSE	9.67	9.22
Sensible heat flux at 10 m ( $\text{W m}^{-2}$ )	BIAS	14.8	17.5
	RMSE	56.5	59.4

Overall, GFS Forecast datasets have better performance than the GFS Reanalysis in predicting the meteorological properties both in surface level and higher elevations. For example, the bias in horizontal wind velocity with Forecast datasets near the surface level is  $0.02 \text{ m s}^{-1}$ , exhibiting -0.22 to  $0.67 \text{ m s}^{-1}$  ( $\alpha=0.1$ ) less bias than the Reanalysis cases. At higher elevations, the difference of bias between Forecast and Reanalysis cases is even more than the surface level comparison. Moreover, GFS Forecast results in -14.3 to  $24.8^\circ$

and  $-13.2$  to  $18.6 \text{ W m}^{-2}$  ( $\alpha=0.1$ ) less bias than Reanalysis cases in predicting the wind direction and sensible heat flux, respectively, near surface level. However, the prediction of relative humidity at 2 m improves by  $-1.97$  to  $3.2\%$  ( $\alpha=0.1$ ) when GFS Reanalysis datasets are implemented in WRF simulations. Based on these findings, the Forecast datasets are preferred over Reanalysis datasets. This recommendation is only valid for running WRF up to 36 hours, as longer run times may yield a different result suggesting and reanalysis datasets may be preferred.

#### 4.3.4 Meteorological Properties over the Mining Facility in Winter

The Winter variations of various atmospheric properties and the comparison between the WRF simulations and the experimental observations are evaluated in this discussion. The experimental observations were conducted from February 16 to March 24 in 2019. Like the Summer season, several meteorological instruments such as SODAR, TANAB, 3D ultrasonic anemometers, and weather station were used for field observations in the mining facility. Based on the availability of experimental datasets, three different days, March 16, 18, and 21 in 2019, were chosen to execute the WRF simulations. The bias and RMSE of different meteorological properties are shown in Table 4.7.

##### Horizontal Wind Velocity

Figure 4.35 shows the diurnal variations of horizontal wind velocity at 10 m elevation. From Table 4.7, the bias of wind speed is higher compared to Summer 2018 WRF simulations by  $0.39$  to  $1.95 \text{ m s}^{-1}$  ( $\alpha=0.1$ ) for both thermally stable and unstable conditions.

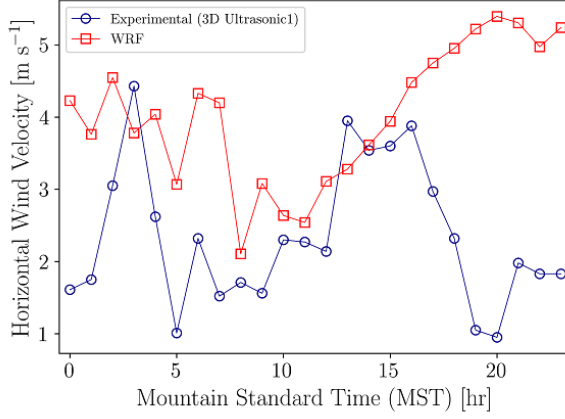
Figures 4.36 and C.1 illustrate the spatial distributions of horizontal wind velocity at different times in March 16, 2019. In the Winter season, occasionally the wind speed is more during early morning and late night hours compared the convective hours. This phenomenon is totally opposite to the Summer season. This is a feature of northern latitude climates such as the Arctic. However, unlike the Summer season, no circulation is observed in the mine because of the higher wind speeds during this season.

In support of the spatial distribution plots, the same phenomena are observed in the wind velocity profiles as shown in Figure 4.37. A high velocity of wind is observed in the early morning and late night at low elevations than the daytime. Although the mine was calmer in the Summer season during the early morning hours, in the Winter the wind velocity is

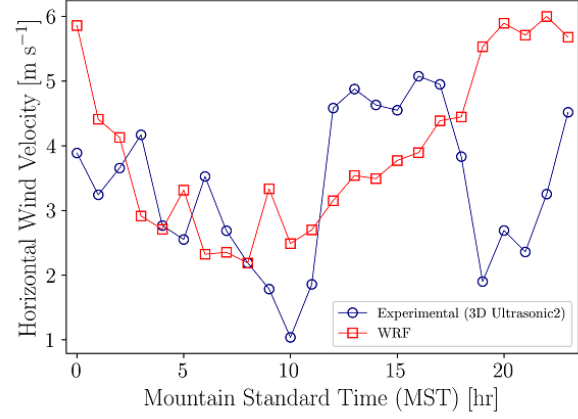
Table 4.7: Seasonal variations of various atmospheric properties in the mining facility; table shows the bias and RMSE between the WRF model and field observations.

Parameter	Errors	Summer 2018	Winter 2019
Horizontal wind velocity at 10 m ( $\text{m s}^{-1}$ )	BIAS	-0.12	-1.29
	RMSE	1.59	3.70
Wind direction at 10 m ( $^{\circ}$ )	BIAS	2.12	-17.4
	RMSE	73.9	51.9
Horizontal wind velocity from TANAB ( $\text{m s}^{-1}$ )	BIAS	-0.21	1.50
	RMSE	1.64	2.49
Horizontal wind velocity from SODAR ( $\text{m s}^{-1}$ )	BIAS	0.01	2.38
	RMSE	1.59	3.70
Wind direction from SODAR ( $^{\circ}$ )	BIAS	-6.91	-6.86
	RMSE	74.4	47.2
Temperature at 2 m ( $^{\circ}\text{C}$ )	BIAS	0.28	-5.86
	RMSE	2.57	6.78
Potential temperature gradient from TANAB ( $\text{K m}^{-1}$ )	BIAS	0.017	-0.014
	RMSE	0.029	0.033
Relative humidity at 2 m (%)	BIAS	-1.84	47.3
	RMSE	9.45	49.9
Sensible heat flux at 10 m ( $\text{W m}^{-2}$ )	BIAS	17.9	9.03
	RMSE	59.5	36.1

higher than the other locations in the facility during those hours. On the other hand, the mine becomes calmer during daytime wherein the wind velocity is higher in the barren area during convective hours.

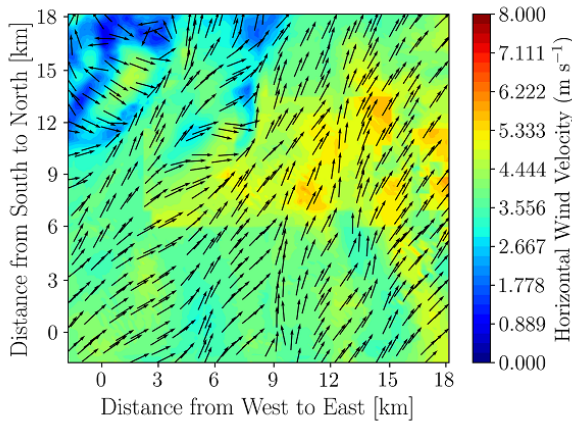


(a) 3D Ultrasonic 1

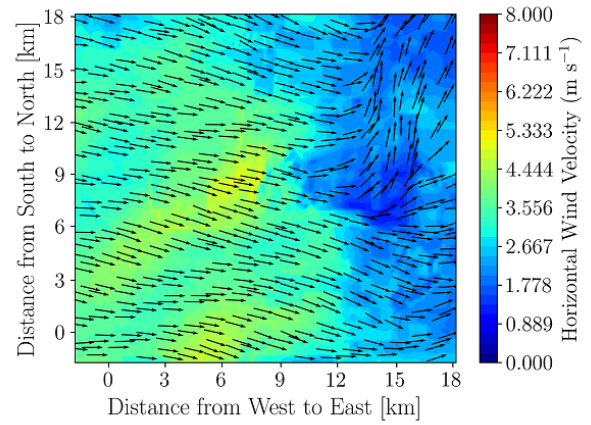


(b) 3D Ultrasonic 2

Figure 4.35: Horizontal wind velocity at 10 m elevation in Winter 2019 showing the median values of model and observation for each hour.

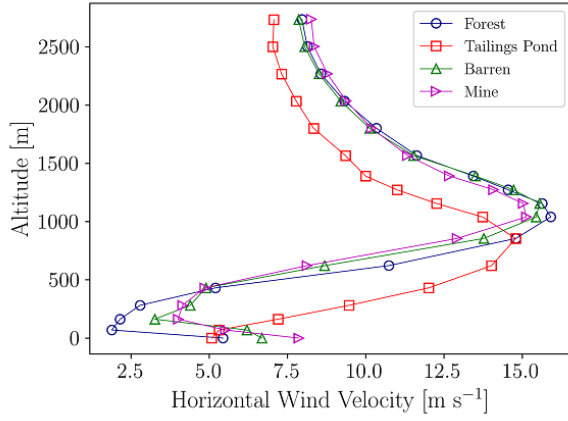


(a) Horizontal wind velocity at 0200 MST

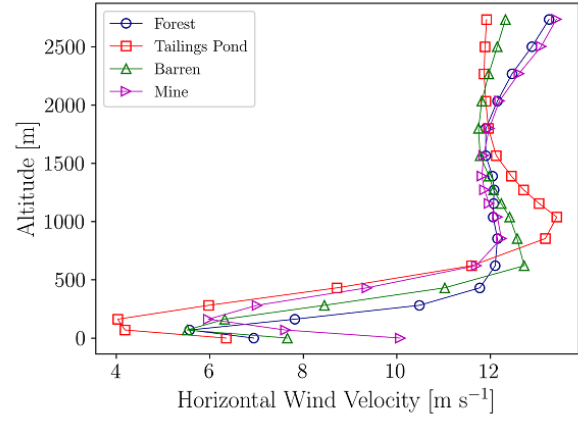


(b) Horizontal wind Velocity at 1400 MST

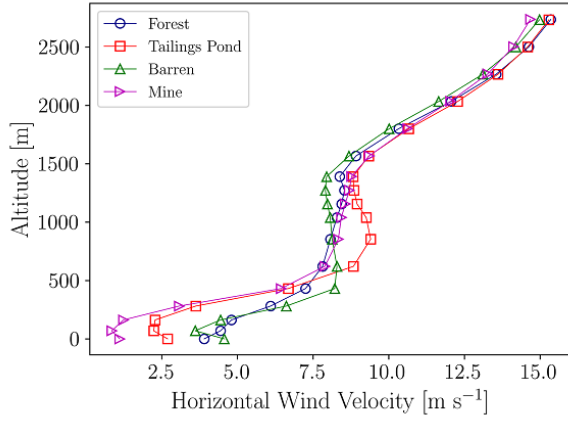
Figure 4.36: Horizontal wind velocity at 10 m in Winter on March 16, 2019; times in Mountain Standard Time (MST).



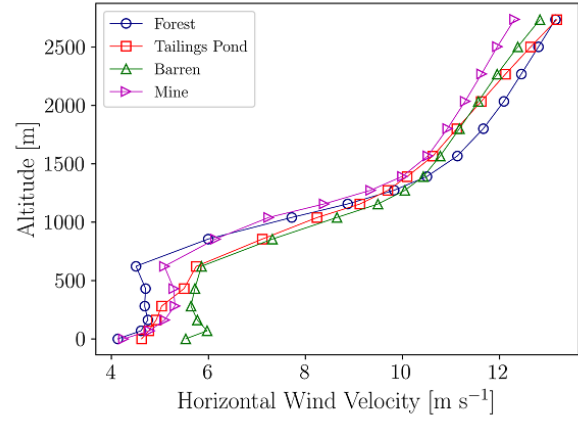
(a) Horizontal wind velocity at 0200 MST



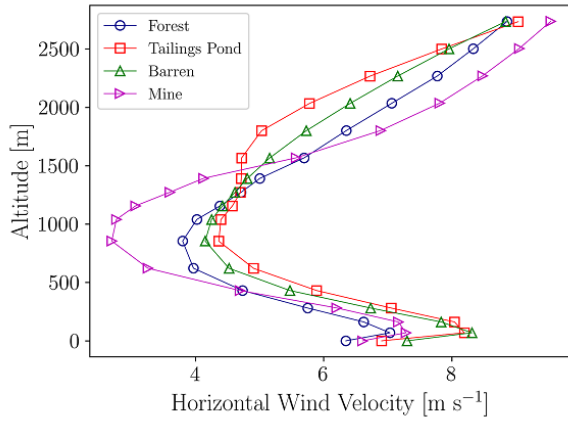
(b) Horizontal wind velocity at 0600 MST



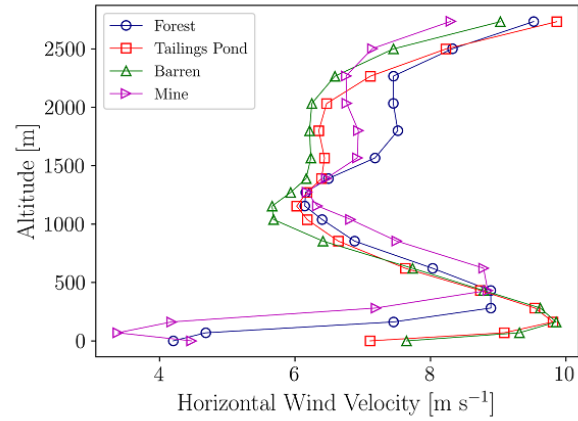
(c) Horizontal wind velocity at 1000 MST



(d) Horizontal wind velocity at 1400 MST



(e) Horizontal wind velocity at 1800 MST



(f) Horizontal wind velocity at 2200 MST

Figure 4.37: Horizontal wind velocity profiles based on pressure height at different locations on March 16, 2019; times in Mountain Standard Times (MST).

## Wind Direction

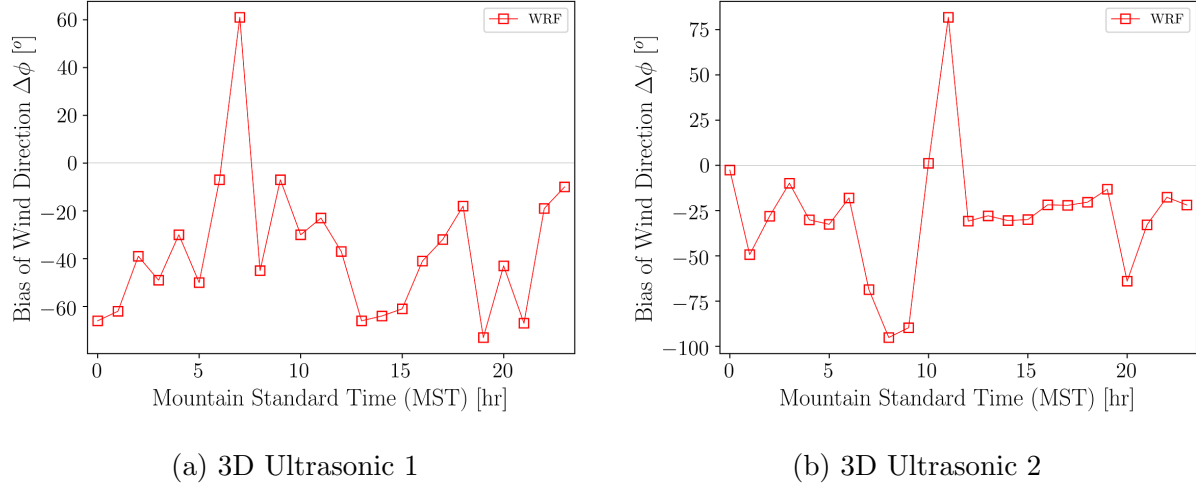


Figure 4.38: Bias of wind direction at 10 m elevation in Winter 2019 showing the median values of wind direction bias for each hour.

The bias of wind direction is almost consistent at surface level and higher elevations in both thermally stable and unstable conditions as shown in Figure 4.38. In Summer as the wind flows very slowly during early morning and late night hours, instruments have some limitations to measure the actual wind direction for low-velocity wind, and hence, there is a large bias between the observations and numerical results. However, in the winter season, because of the higher wind velocities, such difficulties are not observed in these meteorological instruments and that is why less bias of wind direction is observed in the Winter season. However, some large bias of wind direction is still present in the Winter WRF simulations, especially at thermally neutral conditions when the atmosphere changes its thermal stability status. Moreover, as shown in Figure C.2 the topography of the mining facility does not have a significant impact to vary the wind direction at different locations because of the higher wind velocities. Apart from this, the solar radiation energy budget of the surface is reduced and the snow covers the entire facility in the Winter, which decreases the rapid changes of surface temperatures.

## Temperature and Relative Humidity

The temporal variation of temperature and relative humidity at 2 m are shown in Figure 4.39. Although WRF can predict the temperature more accurately in the early morning,



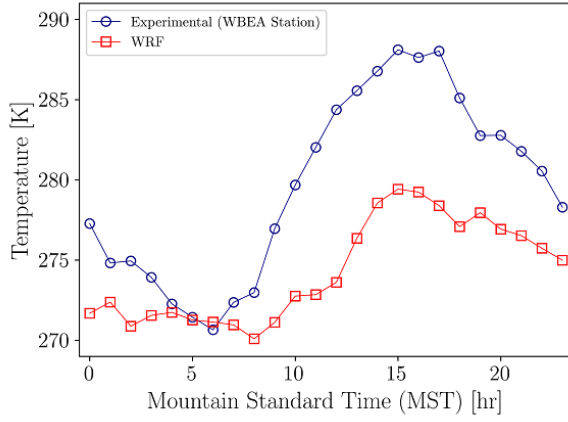
it struggles to predict the actual temperature during daytime and hence the overall bias of temperature is  $-5.86^{\circ}\text{C}$ , while in the Summer it is  $0.28^{\circ}\text{C}$ . However, WRF can still predict the diurnal trend of the temperature while under predicting it. On the other hand, WRF over predicts the relative humidity in the Winter as the consequence of the underprediction of temperature. There is a 47.3 % difference of relative humidity between the observations and WRF simulations in the Winter compared to only  $-1.84\%$  difference in the Summer. The overall predictability of surface-level temperature and relative humidity exhibit more bias in the Winter compared to the Summer by 4.73 to  $7.54^{\circ}\text{C}$  and 39 to 59 % ( $\alpha=0.1$ ) more bias, respectively.

Figures 4.40 to 4.42 and C.3 to C.5 show the spatial distribution of surface skin temperature, and 2-m temperature and relative humidity on March 16, 2019. As water bodies are warmer than the surrounding areas during nighttime, WRF predicts higher temperatures over the pond than the mining area. However, during the daytime, the mine is warmer than the pond although there is a significant increase in temperature outside the facility in the boreal forest. Moreover, there is no major variation of relative humidity in the early morning and late night, while during the daytime in the mining facility WRF predicts higher relative humidity than other regions in domain 5.

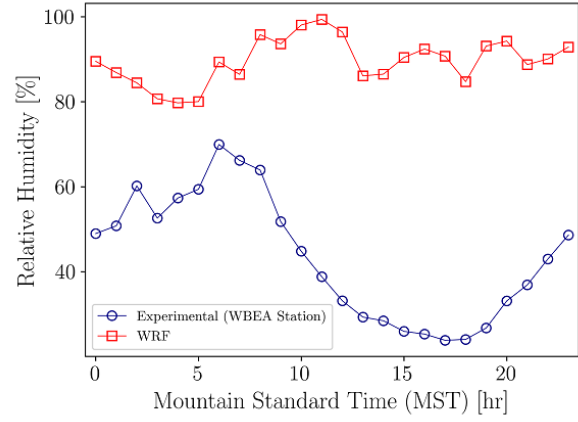
The temperature profiles at different locations on March 16, 2019 are shown in Figure 4.43 for both thermally stable and unstable conditions. It is observed that the pond is warmer during night time while it is cooler during convective hours. On the other hand, temperature over the mine is lower than the other regions in most of the times. Moreover, the atmosphere is mostly stable in the Winter season over the mining facility as there is a positive temperature gradient observed in most of the temperature profiles. However, at 1000 and 1400 MST some negative temperature gradients appear in the lower elevations when the atmosphere becomes unstable because of the higher surface temperature at convective hours.

### **Sensible Vertical Turbulent Heat Flux**

The sensible vertical turbulent heat flux at 10 m elevation is shown in Figure 4.44. In Winter, with reduced solar radiation, reduced temperature, and snow cover, there is a significantly negative heat flux during many hours. Similar to the Summer WRF simulations, the bias of heat flux is more during convective hours compared to the early morning and late night observations. However, a large variation of heat flux is observed in 3D Ultrasonic 2 during early morning and convective hours although it shows better agreement during the evening and late night comparisons. Overall, WRF simulations predict the trend of heat

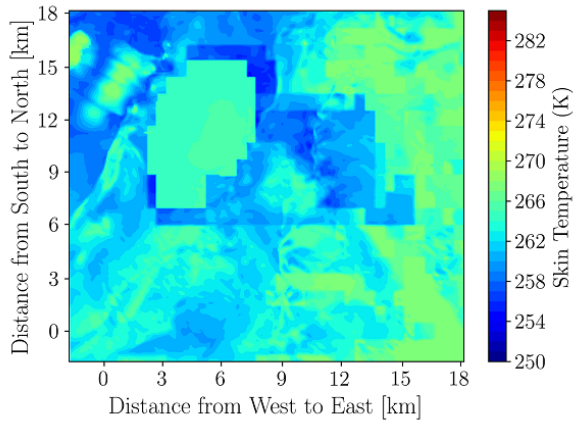


(a) Temperature

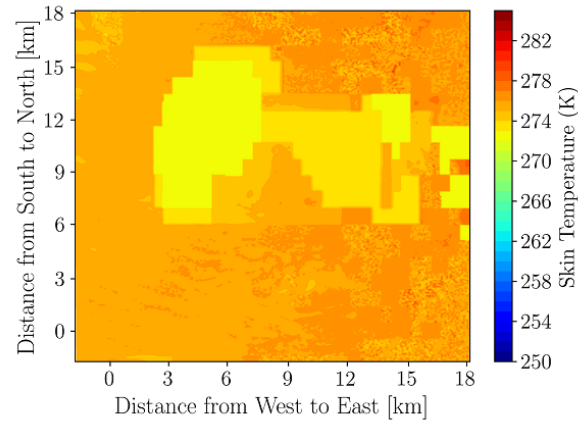


(b) Relative humidity

Figure 4.39: Temperature and relative humidity at 2 m in Winter 2019 showing the median values of model and observation for each hour.

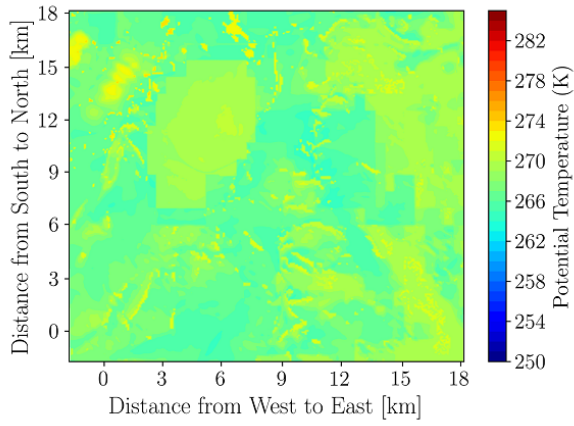


(a) Surface skin temperature at 0200 MST

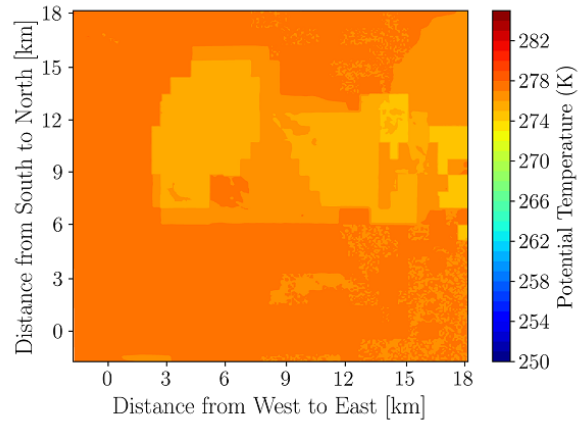


(b) Surface skin temperature at 1400 MST

Figure 4.40: Surface skin temperature on March 16, 2019.

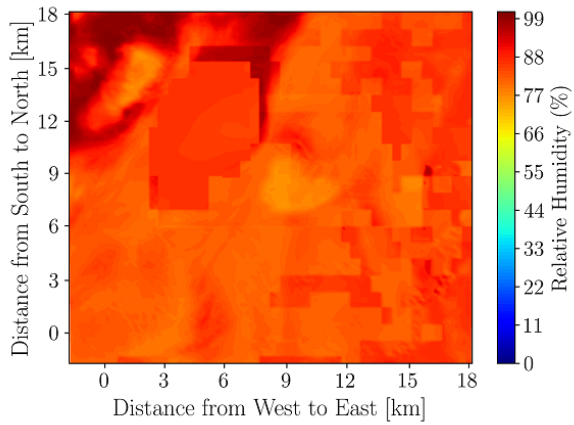


(a) Potential temperature at 0200 MST

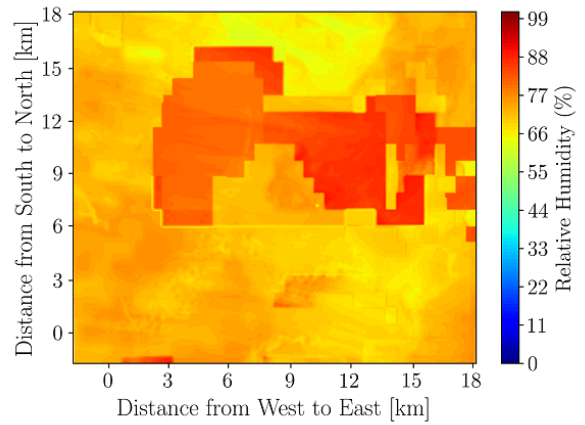


(b) Potential temperature at 1400 MST

Figure 4.41: Potential temperature at 2 m on March 16, 2019.

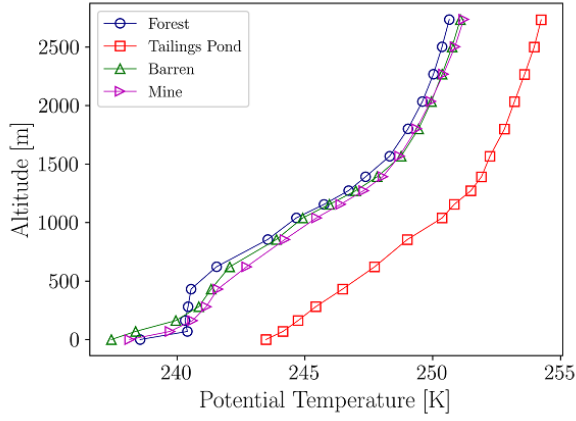


(a) Relative humidity at 0200 MST

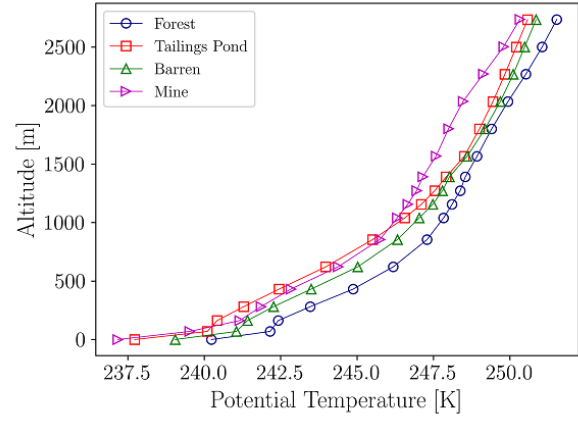


(b) Relative humidity at 1400 MST

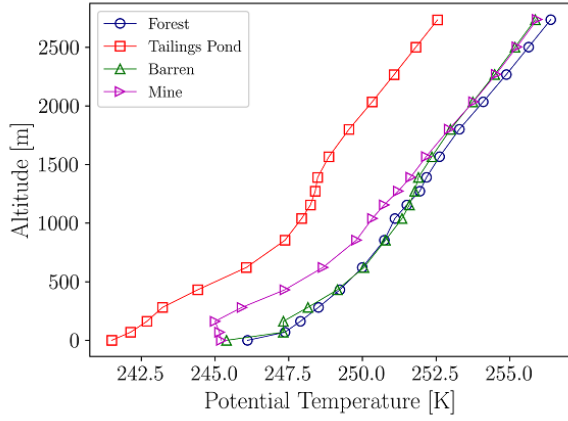
Figure 4.42: Relative Humidity at 2 m on March 16, 2019.



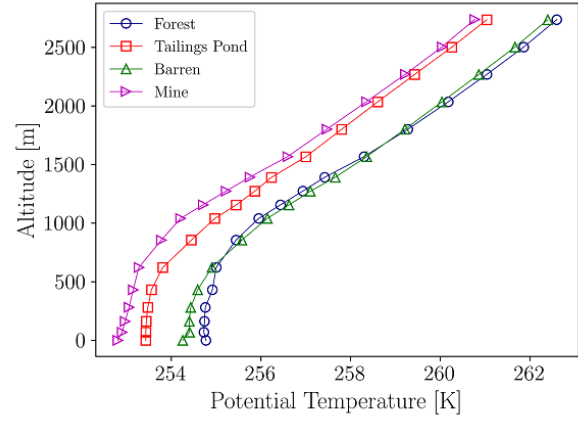
(a) Potential temperature at 0200 MST



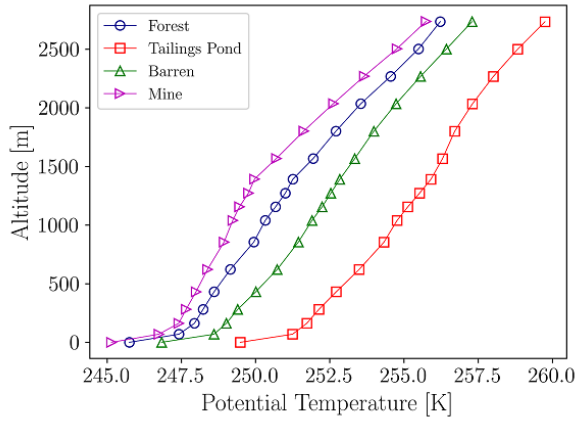
(b) Potential temperature at 0600 MST



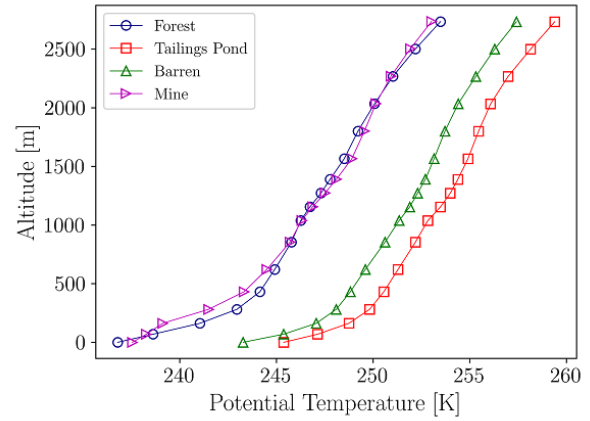
(c) Potential temperature at 1000 MST



(d) Potential temperature at 1400 MST



(e) Potential temperature at 1800 MST



(f) Potential temperature at 2200 MST

Figure 4.43: Potential temperature profiles based on pressure height at different locations on March 16, 2019; times in Mountain Standard Time (MST).

flux qualitatively during the Winter.

The spatial distributions of 10-m sensible heat flux are shown in Figures 4.45 and C.6 at different times. In support of the hourly plots, the heat flux is negative during thermally stable conditions while WRF predicts some positive heat flux during convective hours, most significantly from the boreal forest. However, although the heat flux is higher during day-times, in the mining facility it shows negative heat flux in thermally unstable conditions. This is because of the snow cover reflecting solar radiation, while it is not necessarily the case in the forest area. However, no significant variation of heat flux is observed between the pond and the mine area both during thermally stable and unstable conditions.

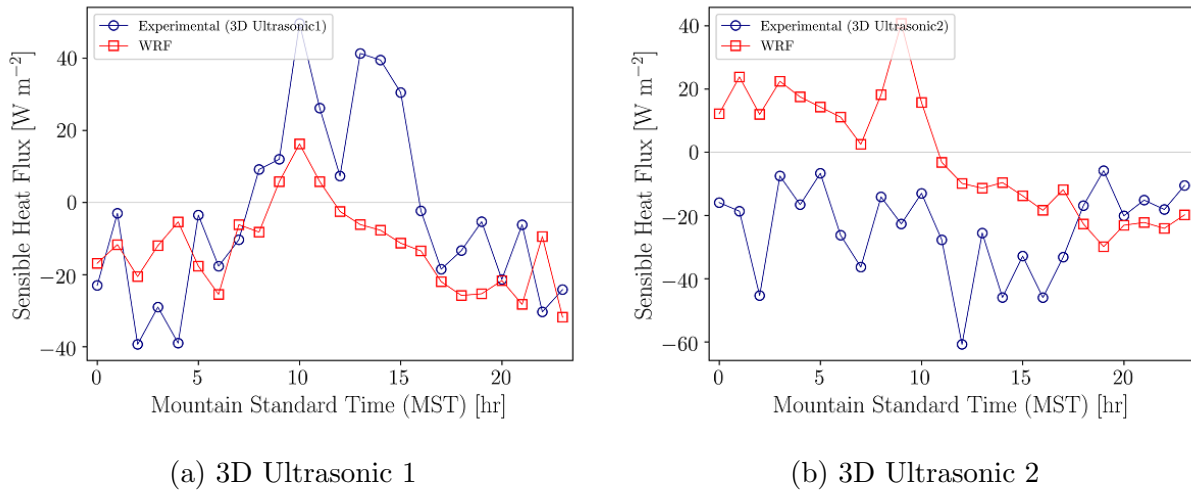


Figure 4.44: Sensible vertical turbulent heat flux at 10 m in Winter 2019 showing the median values of model and observation for each hour.

## 4.4 Conclusions

The Weather Research and Forecasting (WRF) model sensitivity tests are performed to observe distinct meteorological properties over complex topography of a remote open-pit mining facility in northern Canada. The sensitivity study varied the topography, land use, horizontal and vertical grid resolutions, Planetary Boundary-Layer (PBL) parameterization, initial datasets, and the season to arrive at the best configuration for running WRF while minimizing modelling errors in comparison to field observations. Comparisons of model results to observations are performed in both surface level and higher elevations using various

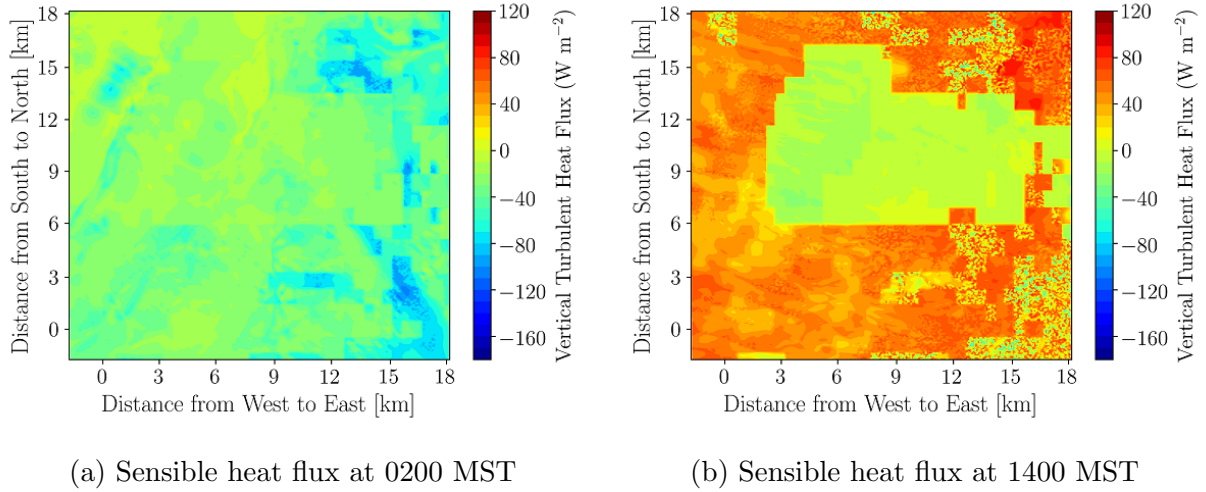


Figure 4.45: Sensible vertical turbulent heat flux at 10 m on March 16, 2019.

meteorological instruments and weather stations including 2D/3D ultrasonic anemometers, a Tethered And Navigated Air Blimp (TANAB), a SOnic Detection And Ranging (SODAR) instrument, cup anemometers, temperature, and relative humidity sensors. The default surface topography in the model was modified using the Shuttle Radar Topography Mission 1 Arc-Second (SRTM 1s) dataset in combination with local LIght Detection And Ranging (LIDAR) observations, while land use change was implemented from a MODerate Resolution Imaging Spectroradiometer (MODIS) 30s data product.

For Summertime simulations, it was observed that during thermally stable conditions the difference between model and observed wind speed and direction are more than those in thermally unstable conditions. This is likely due to the inaccuracies of experimental measurements and PBL parameterizations for thermally stable conditions. The model exhibits a better performance in predicting momentum and heat-related properties when surface topography, land use change, and lake model are accounted for. For instance, accounting for surface topography reduces the wind speed bias by 0.02 to 0.94  $\text{m s}^{-1}$  ( $\alpha = 0.1$ ) when compared to TANAB observations. While modifying land use and implementing a lake model in WRF do not have a significant effect on the prediction of momentum related properties, these changes undoubtedly provide better accuracy in predicting temperature gradient and sensible vertical turbulent heat flux. For instance, the sensible vertical turbulent heat flux can be predicted with 3.84 to 47.4  $\text{W m}^{-2}$  ( $\alpha = 0.1$ ) less error. Hence, modifying land use and implementing a lake model should be considered for more accurate WRF simulations.

As far as grid resolution is concerned, increasing horizontal or vertical resolutions does not significantly impact wind direction predictions. However, it has been found that resolutions greater than mid horizontal and mid vertical predict 1.52 to 3.40 m s<sup>-1</sup> ( $\alpha = 0.1$ ) less bias in wind speed and -8.89 to 23.3 W m<sup>-2</sup> ( $\alpha = 0.1$ ) less bias in heat flux compared to coarser resolutions. Investigation of the PBL parameterizations indicated that both Yonsei University (YSU) and Large-Eddy Simulation (LES) schemes provided similar performance, although LES diverges numerically for some simulations. On the other hand, LES predicts marginally more accurate results for sensible vertical turbulent heat flux than the YSU in most of the WRF simulations by reducing errors from -11.4 to 21.1 W m<sup>-2</sup> ( $\alpha = 0.1$ ).

Compared to Reanalysis datasets, Global Forecasting System (GFS) Forecast datasets have better accuracy in predicting most of the meteorological properties both near surface level and higher elevations. Overall, the Forecast datasets reduce biases with respect to Reanalysis datasets by -0.22 to 0.67 m s<sup>-1</sup> ( $\alpha=0.1$ ), -14.3 to 24.8°, -13.2 to 18.6 W m<sup>-2</sup>, and -1.97 to 3.2% ( $\alpha=0.1$ ) for surface level wind speed, surface-level wind direction, sensible heat flux, and relative humidity, respectively. This finding should be interpreted with caution because it is only valid for WRF runs up to 36 hours into the future. Long run times require further investigations.

The Winter simulations clearly indicate that WRF has less accuracy in predicting most of the meteorological properties during the Winter season. For example, wind velocity bias is 0.39 to 1.95 m s<sup>-1</sup> ( $\alpha=0.1$ ) more in the Winter than the Summer season. However, the RMSE of the wind direction is less during the Winter (51.9°) compared to the Summer (73.9°) because of the higher wind velocity as well as better accuracy of measurements. Moreover, the overall predictability of surface-level temperature and relative humidity exhibit more bias in the Winter compared to the Summer by 4.73 to 7.54 °C and 39 to 59 % ( $\alpha=0.1$ ), respectively. However, the predictability of heat flux is improved by -4.6 to 22.4 W m<sup>-2</sup> ( $\alpha=0.1$ ) in the Winter compared to the Summer. Nevertheless, WRF predicts a reasonable and qualitative behaviour of the Atmospheric Boundary Layer (ABL) in the Winter, which is characterized by higher wind speeds, more thermally stable conditions, and reduced sensible heat flux.

# Chapter 5

## Conclusions and Further Work

In this thesis, the sensitivity tests of the Weather Research and Forecasting (WRF) model are performed to observe distinct meteorological properties over complex terrains of an urban area and a remote mining facility in northern Canada. The sensitivity study in the urban area only involves varying the resolution of the surface topography, while for the mining facility, the sensitivity study is more rigorous involving variation of the topography, land use, horizontal and vertical grid resolutions, Planetary Boundary-Layer (PBL) parameterization, forcing data, and seasons. The objective was to arrive at the best configuration for running WRF while minimizing modelling errors in comparison to field observations. Comparisons of model results to observations are performed in both surface level and higher elevations using various meteorological instruments and weather stations including 2D/3D ultrasonic anemometers, a Tethered And Navigated Air Blimp (TANAB), two SONic Detection And Ranging (SODAR) instruments, cup anemometers, temperature, and relative humidity sensors. Key findings are summarized below.

### 5.1 WRF and Topography

The default surface topography in the model was modified using the Shuttle Radar Topography Mission 1 Arc-Second (SRTM 1s) dataset (for both the mine and urban simulations) in combination with local LIght Detection And Ranging (LIDAR) observations (for the mine simulation only). In the mining facility, there is no significant improvement in surface-level wind velocity and wind direction as a result of increasing topographical resolutions. However, accounting for high-resolution surface topography reduces the wind speed bias by 0.02 to 0.94 m s<sup>-1</sup> ( $\alpha = 0.1$ ) when compared to TANAB observations probing the surface layer



up to 200 m in altitude. On the other hand, modifying surface topography does not have a major impact on heat-related properties.

In the urban area, low topographic resolution WRF simulations show reasonable agreement with observed relative humidity, while high topographic resolution simulations predict the potential temperature more accurately at surface level. For example, the Normalized Mean Square Error (NMSE) of relative humidity is 0.00029 % with the low-topographic resolution, compared to the NMSE of the relative humidity of 0.00089 % for high topographic WRF simulations. On the other hand, the NMSE of temperature for low topographic WRF simulations is 0.00014 %, which is marginally higher than the high topographic WRF simulations of having NMSE of about 0.00011 %. The error of horizontal wind velocity is less with high topographic resolution WRF simulations while low topographic resolution WRF simulations deliver much better results in terms of predicting the vertical wind velocity. For instance, the NMSE of vertical velocity is only 0.005 % for low topographic WRF simulation, while it is 0.086 % for WRF simulations with higher topographic resolution. It is also found that predictions of surface-level wind direction are less reliable in the urban area, especially at thermally stable conditions, compared to wind directions at higher altitudes. However, as the lower atmosphere is calm during thermally stable hours, the wind moves very slowly, which may also cause inaccurate measurements of wind directions using the anemometers and result in large biases of wind direction in thermally stable condition.

## 5.2 WRF and Land Use Classification

Land use change was implemented from a MODerate Resolution Imaging Spectroradiometer (MODIS) 30s data product. Only the mine simulations are investigated for this sensitivity test. The model exhibits a better performance in predicting heat-related properties when land use change and lake modelling are accounted for. While modifying land use and implementing a lake model in WRF do not have a significant effect on the prediction of momentum related properties, these changes undoubtedly provide better accuracy in predicting temperature gradient, and sensible vertical turbulent heat flux predictions. For instance, the sensible vertical turbulent heat flux can be predicted with 3.84 to 47.4 W m<sup>-2</sup> ( $\alpha = 0.1$ ) less bias which encourages the implementation of the actual land use and lakes physics for WRF simulations. Hence, modifying land use and implementing a lake model should be considered to analyze the atmospheric conditions more accurately.

### 5.3 WRF and Grid Resolution

As far as grid resolution is concerned, only the mine simulations are investigated for this sensitivity test. Increasing horizontal or vertical resolutions does not significantly impact wind direction predictions. However, increasing horizontal and vertical resolutions can reduce bias up to 1.52 to 3.40 m s<sup>-1</sup> ( $\alpha = 0.1$ ) in wind speed predictions near the surface up to 200 m altitude. The effect of grid resolutions on relative humidity and sensible vertical heat flux is marginal although mid horizontal and mid vertical or any grid resolutions higher than this improve the sensible heat flux by predicting -8.89 to 23.3 W m<sup>-2</sup> ( $\alpha = 0.1$ ) less bias compared to the other grid resolutions. Overall, the resolution having a horizontal grid size of 0.12 km or 0.09 km in the smallest domain and implementation of 90 vertical levels should be sufficient to provide accurate predictions in comparison to experimental measurements of the mining facility.

### 5.4 WRF and Planetary Boundary-layer Parameterization

Investigation of the PBL parameterizations is only considered for the mine case. It was indicated that both Yonsei University (YSU) and Large-Eddy Simulation (LES) schemes provided similar performance, although LES diverges numerically for some simulations. On the other hand, LES predicts marginally more accurate results for sensible vertical turbulent heat flux than the YSU in most of the WRF simulations by reducing errors from -11.4 to 21.1 W m<sup>-2</sup> ( $\alpha = 0.1$ ). Overall, either YSU or LES can be chosen based on the atmospheric properties of interest, bearing in mind that each provides unique advantages.

### 5.5 WRF and Global Forecasting System Initial and Boundary Datasets

Global Forecasting System (GFS) Forecast datasets have better accuracy than the GFS Reanalysis in predicting the meteorological properties both at surface level and higher elevations. Only the mine case was considered for this sensitivity test. Overall, the Forecast datasets reduce biases with respect to Reanalysis datasets by -0.22 to 0.67 m s<sup>-1</sup> ( $\alpha=0.1$ ), -14.3 to 24.8°, -13.2 to 18.6 W m<sup>-2</sup>, and -1.97 to 3.2% ( $\alpha=0.1$ ) for surface level wind speed,

surface-level wind direction, sensible heat flux, and relative humidity, respectively. Overall, GFS Forecast datasets are recommended. It must be noted that this recommendation is only valid when running WRF up to 36 hours into the future. Longer run times require a separate analysis and may suggest reanalysis datasets being preferred.

## 5.6 WRF and Seasonal Variations of Meteorological Properties

Based on the findings of this study, WRF shows better agreements with observations in the Summer season compared to the Winter season. Only the mine case is considered for this sensitivity test. For instance, in the Winter, WRF exhibits 0.39 to 1.95 m s<sup>-1</sup> ( $\alpha=0.1$ ) more bias in wind velocity compared to the Summer. However, the RMSE of the wind direction is less during the Winter (51.9°) compared to the Summer (73.9°) because of the higher wind velocity as well as better accuracy of measurements. As in the Winter season snow covers the mining facility and solar radiation is weak, the biases of heat-related properties such as temperature and relative humidity are increased significantly compared to the Summer season. For example, the bias of temperature is 4.73 to 7.54 °C ( $\alpha=0.1$ ) more in the Winter than Summer, while the relative humidity bias is 39 to 59% ( $\alpha=0.1$ ) more in the Winter than the Summer. However, the predictability of heat flux is improved by -4.6 to 22.4 W m<sup>-2</sup> ( $\alpha=0.1$ ) in the Winter compared to the Summer. Overall, WRF predicts the dynamics and thermodynamics of ABL qualitatively well in the Winter, characterized by higher wind speeds, increased thermal stability, and reduced turbulent heat flux.

## 5.7 Future Work

The limitations of this study include lack of more rigorous observations and area-distributed measurements with larger spatial coverage. There is also a limitation of measuring temperature profiles to evaluate the performance of WRF at higher elevations. Future research should investigate seasonal variations and applications of WRF to other complex topographies to confirm the findings of this thesis. More fundamental contributions should also be made to reduce WRF errors in such applications, particularly for the northern climate. Future research should also account for the anthropogenic release of heat at a region to better account for the surface energy budget toward more accurate meteorological predictions. One

possibility is to employ WRF-Chem to model plumes of area fugitive green-house gas emissions to understand meteorological implications over a complex topography with land use alterations in area fugitive emissions. The current WRF results can also be potentially used as boundary and initial conditions for use with high-resolution computational fluid dynamics simulations.

# Bibliography

- [1] ABUALKISHIK, A. Z. A comparative study on the software architecture of WRF and other numerical weather prediction models. *Journal of Theoretical and Applied Information Technology* 96, 24 (2018).
- [2] ALBERT, D. G., DECATO, S. N., AND CARBEE, D. L. Snow cover effects on acoustic sensors. *Cold Reg. Sci. Technol.* 52, 2 (2008), 132–145.
- [3] ALEXANDRU, A., DE ELIA, R., AND LAPRISE, R. Internal variability in regional climate downscaling at the seasonal scale. *Mon. Weather Rev.* 135, 9 (2007), 3221–3238.
- [4] ALIABADI, A. A., KRAYENHOFF, E. S., NAZARIAN, N., CHEW, L. W., ARMSTRONG, P. R., AFSHARI, A., AND NORFORD, L. K. Effects of roof-edge roughness on air temperature and pollutant concentration in urban canyons. *Bound.-Lay. Meteorol.* 164, 2 (2017), 249–279.
- [5] ALIABADI, A. A., MORADI, M., CLEMENT, D., LUBITZ, W., AND GHARABAGHI, B. Flow and temperature dynamics in an urban canyon under a comprehensive set of wind directions, wind speeds, and thermal stability conditions. *Environ. Fluid Mech.* 19, 1 (2019), 81–109.
- [6] ALIABADI, A. A., STAEBLER, R., DE GRANDPRÉ, J., ZADRA, A., AND VAILLANCOURT, P. Comparison of estimated atmospheric boundary layer mixing height in the Arctic and Southern Great Plains under statically stable conditions: Experimental and numerical aspects. *Atmos. Ocean* 54, 1 (2016), 60–74.
- [7] ALIABADI, A. A., STAEBLER, R. M., LIU, M., AND HERBER, A. Characterization and parametrization of Reynolds stress and turbulent heat flux in the stably-stratified lower Arctic troposphere using aircraft measurements. *Bound.-Lay. Meteorol.* 161, 1 (2016), 99–126.
- [8] ALIABADI, A. A., THOMAS, J. L., HERBER, A. B., STAEBLER, R. M., LEITCH, W. R., SCHULZ, H., LAW, K. S., MARELLE, L., BURKART, J., WILLIS, M. D., ET AL. Ship emissions measurement in the Arctic by plume intercepts of the Canadian Coast Guard icebreaker Amundsen from the Polar 6 aircraft platform. *Atmos. Chem. Phys.* 16, 12 (2016), 7899–7916.

- [9] ALIGO, E. A., GALLUS JR, W. A., AND SEGAL, M. On the impact of WRF model vertical grid resolution on Midwest summer rainfall forecasts. *Weather Forecast.* 24, 2 (2009), 575–594.
- [10] ANDERSON, P. S., LADKIN, R. S., AND RENFREW, I. A. An autonomous Doppler SODAR wind profiling system. *J. Atmos. Ocean. Tech.* 22, 9 (2005), 1309–1325.
- [11] ANTONIOU, I., COURTNEY, M., JØRGENSEN, H. E., MIKKELSEN, T., VON HUNERBEIN, S., BRADLEY, S., PIPER, B., HARRIS, M., MARTI, I., ARISTU, M., ET AL. Remote sensing the wind using Lidars and Sodars. In *Proceedings EWEC* (2007).
- [12] ANTONIOU, I., JØRGENSEN, H. E., BRADLEY, S. G., HUNERBEIN, S., CUTLER, N., KINDLER, D., NOORD, M., AND WARMBIER, G. The profiler intercomparison experiment (PIE). In *European Wind Energy Conference and Exhibition, London (GB)* (2004), pp. 1–7.
- [13] ARNOLD, D., MORTON, D., SCHICKER, I., SEIBERT, P., ROTACH, M., HORVATH, K., DUDHIA, J., SATOMURA, T., MÜLLER, M., ZÄNGL, G., ET AL. *High resolution modelling in complex terrain: Report on the HiRCoT 2012 Workshop, Vienna, 21-23 February 2012*. Institut für Meteorologie, Department Wasser-Atmosphäre-Umwelt, Univ. f. Bodenkultur, 2012.
- [14] ARTHUR, R. S., LUNDQUIST, K. A., MIROCHA, J. D., AND CHOW, F. K. Topographic effects on radiation in the WRF model with the immersed boundary method: Implementation, validation, and application to complex terrain. *Mon. Weather Rev.* 146, 10 (2018), 3277–3292.
- [15] AWAN, N. K., TRUHETZ, H., AND GOBIET, A. Parameterization-induced error characteristics of MM5 and WRF operated in climate mode over the Alpine region: an ensemble-based analysis. *J. Climate* 24, 12 (2011), 3107–3123.
- [16] BAKLANOV, A., SCHLÜNZEN, K., SUPPAN, P., BALDASANO, J., BRUNNER, D., AKSOYOGLU, S., CARMICHAEL, G., DOUROS, J., FLEMMING, J., FORKEL, R., ET AL. Online coupled regional meteorology chemistry models in Europe: Current status and prospects. *Atmos. Chem. Phys.* 14, 1 (2014), 317–398.
- [17] BAXTER, R. Development of a universal acoustic pulse transponding system for performance auditing of Sodars. In *Proceedings from the 7th International Symposium on Acoustic Remote Sensing and Associated Techniques of the Atmosphere and Oceans* (1994), pp. 3–35.
- [18] BEDOYA, C., ISAZA, C., DAZA, J. M., AND LÓPEZ, J. D. Automatic identification of rainfall in acoustic recordings. *Ecol. Indic.* 75 (2017), 95–100.

- [19] BEHRENS, P., OSULLIVAN, J., ARCHER, R., AND BRADLEY, S. Underestimation of monostatic SODAR measurements in complex terrain. *Bound.-Lay. Meteorol.* 143, 1 (2012), 97–106.
- [20] BERG, L. K., AND ZHONG, S. Sensitivity of MM5-simulated boundary layer characteristics to turbulence parameterizations. *J. Appl. Meteorol.* 44, 9 (2005), 1467–1483.
- [21] BEYRICH, F. Mixing height estimation from SODAR data - A critical discussion. *Atmos. Environ.* 31, 23 (1997), 3941–3953.
- [22] BRADLEY, S. *Atmospheric acoustic remote sensing: principles and applications*. CRC press, 2007.
- [23] CARPENTER, R., SHAW, B. L., MARGULIS, M., BARR, K. S., BAYNARD, T., YATES, D., AND SHARP, J. Short-term numerical forecasts using WindTracer LIDAR data. In *Fourth Conference on Weather, Climate, and the New Energy Economy*, American Meteorological Society (Austin, TX, 2013).
- [24] CARVALHO, D., ROCHA, A., GÓMEZ-GESTEIRA, M., AND SANTOS, C. A sensitivity study of the WRF model in wind simulation for an area of high wind energy. *Environ. Modell. Softw.* 33 (2012), 23–34.
- [25] CERLINI, P. B., EMANUEL, K. A., AND TODINI, E. Orographic effects on convective precipitation and space-time rainfall variability: preliminary results. *Hydrol. Earth Syst. Sc.* 9, 4 (2005), 285–299.
- [26] CHALLA, V. S., INDRACANTI, J., RABARISON, M. K., YOUNG, J., PATRICK, C., BAHAM, J. M., HUGHES, R., AND YERRAMILLI, A. Numerical experiments on the sensitivity of WRF-CMAQ simulations of air quality in the Mississippi Gulf coastal region to PBL and Land surface models. In *The 6th Annual CMAQ Conference, Chapel Hill, NC, October 1-3* (2007).
- [27] CHAN, P. Measurement of turbulence intensity profile by a mini-SODAR. *Meteorol. Appl.* 15, 2 (2008), 249–258.
- [28] CHENG, F.-Y., HSU, Y.-C., LIN, P.-L., AND LIN, T.-H. Investigation of the effects of different land use and land cover patterns on mesoscale meteorological simulations in the Taiwan area. *J. Appl. Meteorol. Clim.* 52, 3 (2013), 570–587.
- [29] CHENG, W. Y., AND STEENBURGH, W. J. Evaluation of surface sensible weather forecasts by the WRF and the Eta models over the Western United States. *Weather Forecast.* 20, 5 (2005), 812–821.
- [30] CHIN, H., GLASCOE, L., LUNDQUIST, J., AND WHARTON, S. Impact of WRF physics and grid resolution on low-level wind prediction: towards the assessment of climate change impact on future wind power. Tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA, United States, February 2010.

- [31] CHU, X., XUE, L., GEERTS, B., RASMUSSEN, R., AND BREED, D. A case study of Radar observations and WRF LES simulations of the impact of ground-based glaciogenic seeding on orographic clouds and precipitation. Part I: Observations and model validations. *J. Appl. Meteorol. Clim.* 53, 10 (2014), 2264–2286.
- [32] CIVEROLO, K., HOGREFE, C., LYNN, B., ROSENTHAL, J., KU, J.-Y., SOLECKI, W., COX, J., SMALL, C., ROSENZWEIG, C., GOLDBERG, R., ET AL. Estimating the effects of increased urbanization on surface meteorology and ozone concentrations in the New York City metropolitan region. *Atmos. Environ.* 41, 9 (2007), 1803–1818.
- [33] COSTANTINO, L., HEINRICH, P., MZÉ, N., AND HAUCHECORNE, A. Convective gravity wave propagation and breaking in the stratosphere: comparison between WRF model simulations and LIDAR data. In *Ann. Geophys.* (2015), vol. 33, pp. 1155–1171.
- [34] COULTER, R., AND KALLISTRATOVA, M. Two decades of progress in SODAR techniques: a review of 11 ISARS proceedings. *Meteorol. Atmos. Phys.* 85, 1-3 (2004), 3–19.
- [35] CRESCENTI, G. H. The degradation of Doppler SODAR performance due to noise: A review. *Atmos. Environ.* 32, 9 (1998), 1499–1509.
- [36] CUXART, J. When can a high-resolution simulation over complex terrain be called LES? *Front. Earth Sci.* 3 (2015), 87.
- [37] DE MEIJ, A., BOSSIOLI, E., PENARD, C., VINUESA, J., AND PRICE, I. The effect of SRTM and Corine Land Cover data on calculated gas and PM10 concentrations in WRF-Chem. *Atmos. Environ.* 101 (2015), 177–193.
- [38] DE MEIJ, A., AND VINUESA, J. Impact of SRTM and Corine Land Cover data on meteorological parameters using WRF. *Atmos. Res.* 143 (2014), 351–370.
- [39] DIAGNE, M., DAVID, M., BOLAND, J., SCHMUTZ, N., AND LAURET, P. Post-processing of solar irradiance forecasts from WRF model at Reunion Island. *Sol. Energy* 105 (2014), 99–108.
- [40] DRAXL, C., AND MAYR, G. J. Meteorological wind energy potential in the Alps using ERA40 and wind measurement sites in the Tyrolean Alps. *Wind Energy* 14, 4 (2011), 471–489.
- [41] DUDHIA, J. Numerical study of convection observed during the winter monsoon experiment using a mesoscale two-dimensional model. *J. Atmos. Sci.* 46, 20 (1989), 3077–3107.
- [42] EMEIS, S., AND TÜRK, M. Frequency distributions of the mixing height over an urban area from SODAR data. *Meteorol. Z.* 13, 5 (2004), 361–367.



- [43] FORSTER, P., RAMASWAMY, V., ARTAXO, P., BERNTSEN, T., BETTS, R., FAHEY, D. W., HAYWOOD, J., LEAN, J., LOWE, D. C., MYHRE, G., ET AL. Changes in atmospheric constituents and in radiative forcing. Chapter 2. In *Climate Change 2007. The Physical Science Basis*. 2007.
- [44] FRIEDL, M. A., SULLA-MENASHE, D., TAN, B., SCHNEIDER, A., RAMANKUTTY, N., SIBLEY, A., AND HUANG, X. MODIS Collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sens. Environ.* *114*, 1 (2010), 168–182.
- [45] GARRATT, J. R. The atmospheric boundary layer. *Earth-Sci. Rev.* *37*, 1-2 (1994), 89–134.
- [46] GILLIAM, R. C., AND PLEIM, J. E. Performance assessment of new land surface and planetary boundary layer physics in the WRF-ARW. *J. Appl. Meteorol. Clim.* *49*, 4 (2010), 760–774.
- [47] GIOVANNINI, L., ANTONACCI, G., ZARDI, D., LAITI, L., AND PANZIERA, L. Sensitivity of simulated wind speed to spatial resolution over complex terrain. *Energy Proced.* *59* (2014), 323–329.
- [48] GU, H., JIN, J., WU, Y., EK, M. B., AND SUBIN, Z. M. Calibration and validation of lake surface temperature simulations with the coupled WRF-lake model. *Climatic Change* *129*, 3 (2015), 471–483.
- [49] HALTINER, G. J., AND WILLIAMS, R. T. Numerical prediction and dynamic meteorology. Tech. rep., 1980.
- [50] HAN, Z., SAKURAI, T., UEDA, H., CARMICHAEL, G., STREETS, D., HAYAMI, H., WANG, Z., HOLLOWAY, T., ENGARDT, M., HOZUMI, Y., ET AL. MICS-Asia II: Model intercomparison and evaluation of ozone and relevant species. *Atmos. Environ.* *42* (2008), 3491–3509.
- [51] HE, J., YU, Y., YU, L., LIU, N., AND ZHAO, S. Impacts of uncertainty in land surface information on simulated surface temperature and precipitation over China. *Int. J. Climatol.* *37* (2017), 829–847.
- [52] HIRABAYASHI, S., KROLL, C. N., AND NOWAK, D. J. Component-based development and sensitivity analyses of an air pollutant dry deposition model. *Environ. Modell. Softw.* *26*, 6 (2011), 804–816.
- [53] HONG, S.-Y. A new stable boundary-layer mixing scheme and its impact on the simulated East Asian summer monsoon. *Q. J. Roy. Meteor. Soc.* *136*, 651 (2010), 1481–1496.
- [54] HOUZE, R. A. Orographic effects on precipitating clouds. *Rev. Geophys.* *50*, 1 (2012).

- [55] HU, X.-M., NIELSEN-GAMMON, J. W., AND ZHANG, F. Evaluation of three planetary boundary layer schemes in the WRF model. *J. Appl. Meteorol. Climatol.* 49, 9 (2010), 1831–1844.
- [56] JANJIC, Z. A nonhydrostatic model based on a new approach. *Meteorol. Atmos. Phys.* 82, 1-4 (2003), 271–285.
- [57] JANJIC, Z. I., GERRITY JR, J., AND NICKOVIC, S. An alternative approach to nonhydrostatic modeling. *Mon. Weather Rev.* 129, 5 (2001), 1164–1178.
- [58] JIMÉNEZ, P. A., AND DUDHIA, J. Improving the representation of resolved and unresolved topographic effects on surface wind in the WRF model. *J. Appl. Meteorol. Clim.* 51, 2 (2012), 300–316.
- [59] JIMÉNEZ, P. A., AND DUDHIA, J. On the ability of the WRF model to reproduce the surface wind direction over complex terrain. *J. Appl. Meteorol. Clim.* 52, 7 (2013), 1610–1617.
- [60] JIMÉNEZ-ESTEVE, B., UDINA, M., SOLER, M. R., PEPIN, N., AND MIRÓ, J. R. Land use and topography influence in a complex terrain area: A high resolution mesoscale modelling study over the Eastern Pyrenees using the WRF model. *Atmos. Res.* 202 (2018), 49–62.
- [61] JUNG, S.-H., IM, E.-S., AND HAN, S.-O. The effect of topography and sea surface temperature on heavy snowfall in the Yeongdong region: A case study with high resolution WRF simulation. *Asia-Pac. J. Atmos. Sci.* 48, 3 (2012), 259–273.
- [62] KATURJI, M., ZAWAR-REZA, P., AND ZHONG, S. Surface layer response to topographic solar shading in Antarctica’s dry valleys. *J. Geophys. Res. Atmos* 118, 22 (2013), 12–332.
- [63] KUMAR, P., BHATTACHARYA, B. K., AND PAL, P. Impact of vegetation fraction from Indian geostationary satellite on short-range weather forecast. *Agr. Forest Meteorol.* 168 (2013), 82–92.
- [64] KUMAR, P., KISHTAWAL, C., AND PAL, P. Impact of ECMWF, NCEP, and NCM-RWF global model analysis on the WRF model forecast over Indian Region. *Theor. Appl. Climatol.* 127, 1-2 (2017), 143–151.
- [65] KUMAR, R. *Weather Research and Forecasting model with chemistry (WRF-CHEM) over South Asia*. PhD thesis, University of Hamburg, Hamburg, Germany, 2012.
- [66] KURKOWSKI, N. P., STENSRUD, D. J., AND BALDWIN, M. E. Assessment of implementing satellite-derived land cover data in the Eta model. *Weather Forecast.* 18, 3 (2003), 404–416.

- [67] LAPRISE, R. The Euler equations of motion with hydrostatic pressure as an independent variable. *Mon. Weather Rev.* 120, 1 (1992), 197–207.
- [68] LEMONE, M. A., TEWARI, M., CHEN, F., AND DUDHIA, J. Objectively determined fair-weather CBL depths in the ARW-WRF model and their comparison to CASES-97 observations. *Mon. Weather Rev.* 141, 1 (2013), 30–54.
- [69] LIAN, J., WU, L., BRÉON, F.-M., BROQUET, G., VAUTARD, R., ZACCHEO, T. S., DOBLER, J., AND CIAIS, P. Evaluation of the WRF-UCM mesoscale model and ECMWF global operational forecasts over the Paris region in the prospect of tracer atmospheric transport modeling. *Elem. Sci. Anth.* 6, 1 (2018).
- [70] LIN, Y.-L., FARLEY, R. D., AND ORVILLE, H. D. Bulk parameterization of the snow field in a cloud model. *J. Clim. Appl. Meteorol.* 22, 6 (1983), 1065–1092.
- [71] LORENTE-PLAZAS, R., JIMÉNEZ, P. A., DUDHIA, J., AND MONTÁVEZ, J. P. Evaluating and improving the impact of the atmospheric stability and orography on surface winds in the WRF model. *Mon. Weather Rev.* 144, 7 (2016), 2685–2693.
- [72] LUPAȘCU, A., IRIZA, A., AND DUMITRACHE, R. Using a high resolution topographic data set and analysis of the impact on the forecast of meteorological parameters. *Romanian Reports in Physics* 67, 2 (2015), 653–664.
- [73] MASS, C. F., OVENS, D., WESTRICK, K., AND COLLE, B. A. Does increasing horizontal resolution produce more skillful forecasts? The results of two years of real-time numerical weather prediction over the Pacific Northwest. *B. Am. Meteorol. Soc.* 83, 3 (2002), 407–430.
- [74] MCCARTHY, M., HARPHAM, C., GOODESS, C., AND JONES, P. Simulating climate change in UK cities using a regional climate model, HadRM3. *Int. J. Climatol.* 32, 12 (2012), 1875–1888.
- [75] MIAO, Y., LIU, S., ZHENG, Y., WANG, S., AND CHEN, B. Numerical study of the effects of topography and urbanization on the local atmospheric circulations over the Beijing-Tianjin-Hebei, China. *Adv. Meteorol.* 2015 (2015).
- [76] MIGLIETTA, M. M., THUNIS, P., GEORGIEVA, E., PEDERZOLI, A., BESSAGNET, B., TERRENOIRE, E., AND COLETTE, A. Evaluation of WRF model performance in different European regions with the DELTA-FAIRMODE evaluation tool. *Int. J. Environ. Pollut.* 50, 1 (2012), 83.
- [77] MIHALIKOVA, M., KIRKWOOD, S., ARNAULT, J., AND MIKHAYLOVA, D. Observation of a tropopause fold by MARA VHF wind-profiler radar and ozonesonde at Wasa, Antarctica: comparison with ECMWF analysis and a WRF model simulation. In *Ann. Geophys.* (2012), vol. 30, pp. 1411–1421.

- [78] MIROCHA, J. D., SIMPSON, M. D., FAST, J. D., BERG, L. K., AND BASKETT, R. Investigation of boundary-layer wind predictions during nocturnal low-level jet events using the Weather Research and Forecasting model. *Wind Energy* 19, 4 (2016), 739–762.
- [79] MOENG, C., DUDHIA, J., KLEMP, J., AND SULLIVAN, P. Examining two-way grid nesting for large eddy simulation of the PBL using the WRF model. *Mon. Weather Rev.* 135, 6 (2007), 2295–2311.
- [80] MOORE, J. T., AND GRAVES, C. E. COMETs influence on meteorological education and research within universities: The Saint Louis University experience. In *21st Conference on Weather Analysis and Forecasting/17th Conference on Numerical Weather Prediction* (2005).
- [81] MUGHAL, M. O., LYNCH, M., YU, F., MCGANN, B., JEANNERET, F., AND SUTTON, J. Wind modelling, validation and sensitivity study using Weather Research and Forecasting model in complex terrain. *Environ. Modell. Softw.* 90 (2017), 107–125.
- [82] NAHIAN, M. R., NAZEM, A., MORADI, M., MAHMUD, S., LUBITZ, W. D., AND ALIABADI, A. A. WRF simulations of urban climate: Do we need low or high topographic resolution? In *CSME-CFDSC Congress 2019* (London, Ontario, 2019).
- [83] NOLAN, D. S., ZHANG, J. A., AND STERN, D. P. Evaluation of planetary boundary layer parameterizations in tropical cyclones by comparison of in situ observations and high-resolution simulations of Hurricane Isabel (2003). Part I: Initialization, maximum winds, and the outer-core boundary layer. *Mon. Weather Rev.* 137, 11 (2009), 3651–3674.
- [84] NOSSENT, J., ELSSEN, P., AND BAUWENS, W. Sobol’ sensitivity analysis of a complex environmental model. *Environ. Modell. Softw.* 26, 12 (2011), 1515–1525.
- [85] NTWALI, D., OGWANG, B. A., AND ONGOMA, V. The impacts of topography on spatial and temporal rainfall distribution over Rwanda based on WRF model. *Atmospheric and Climate Sciences* 6, 02 (2016), 145.
- [86] O’BRIEN, J. J. Alternative solutions to the classical vertical velocity problem. *J. Appl. Meteorol.* 9, 2 (1970), 197–203.
- [87] OYAMA, K. V. A thermodynamic foundation for modeling the moist atmosphere. *J. Atmos. Sci.* 47, 21 (1990), 2580–2593.
- [88] PAGÈS, M., PEPIN, N., AND MIRÓ, J. Measurement and modelling of temperature cold pools in the Cerdanya valley (Pyrenees), Spain. *Meteorol. Appl.* 24, 2 (2017), 290–302.

- [89] PAL, J. S., GIORGI, F., BI, X., ELGUINDI, N., SOLMON, F., GAO, X., RAUSCHER, S. A., FRANCISCO, R., ZAKY, A., WINTER, J., ET AL. Regional climate modeling for the developing world: the ICTP RegCM3 and RegCNET. *B. Am. Meteorol. Soc.* 88, 9 (2007), 1395–1410.
- [90] PALMÉN, E. On the mechanism of the vertical heat flux and generation of kinetic energy in the atmosphere. *Tellus* 18, 4 (1966), 838–845.
- [91] PARK, S.-H., KLEMP, J. B., AND KIM, J.-H. Hybrid mass coordinate in WRF-ARW and its impact on upper-level turbulence forecasting. *Mon. Weather Rev.*, 2019 (2019).
- [92] PATZ, J. A., CAMPBELL-LENDRUM, D., HOLLOWAY, T., AND FOLEY, J. A. Impact of regional climate change on human health. *Nature* 438, 7066 (2005), 310.
- [93] PINEDA, N., JORBA, O., JORGE, J., AND BALDASANO, J. Using NOAA AVHRR and SPOT VGT data to estimate surface parameters: Application to a mesoscale meteorological model. *Int. J. Remote Sens.* 25, 1 (2004), 129–143.
- [94] PIPER, B., BRADLEY, S., AND VON HUNERBEIN, S. Calibration method principles for monostatic sodars. *UPWIND Project Report* (2007).
- [95] PLEIM, J. E. A combined local and nonlocal closure model for the atmospheric boundary layer. Part I: Model description and testing. *J. Appl. Meteorol. Climatol.* 46, 9 (2007), 1383–1395.
- [96] PONTOPPIDAN, M., REUDER, J., MAYER, S., AND KOLSTAD, E. W. Downscaling an intense precipitation event in complex terrain: The importance of high grid resolution. *Tellus A: Dynamic Meteorology and Oceanography* 69, 1 (2017), 1271561.
- [97] POWERS, J. G., KLEMP, J. B., SKAMAROCK, W. C., DAVIS, C. A., DUDHIA, J., GILL, D. O., COEN, J. L., GOCHIS, D. J., AHMADOV, R., PECKHAM, S. E., ET AL. The Weather Research and Forecasting model: Overview, system efforts, and future directions. *B. Am. Meteorol. Soc.* 98, 8 (2017), 1717–1737.
- [98] QIAN, Y., GHAN, S. J., AND LEUNG, L. R. Downscaling hydroclimatic changes over the Western US based on CAM subgrid scheme and WRF regional climate simulations. *Int. J. Climatol.* 30, 5 (2010), 675–693.
- [99] ROGERS, A., WALLS, E., HENSON, W., AND MANWELL, J. Addressing ground clutter corruption of SODAR measurements. In *45th AIAA Aerospace Sciences Meeting and Exhibit* (2007), p. 1226.
- [100] ROUGIER, J., SEXTON, D. M., MURPHY, J. M., AND STAINFORTH, D. Analyzing the climate sensitivity of the HadSM3 climate model using ensembles from different but related experiments. *J. Climate* 22, 13 (2009), 3540–3557.

- [101] ROUX, G., LIU, Y., MONACHE, L., SHEU, R.-S., AND WARNER, T. T. Verification of high resolution WRF-RTFDFA surface forecasts over mountains and plains. In *10th WRF users' workshop* (2009), Citeseer, pp. 20–23.
- [102] SAIDE, P. E., CARMICHAEL, G. R., SPAK, S. N., GALLARDO, L., OSSES, A. E., MENA-CARRASCO, M. A., AND PAGOWSKI, M. Forecasting urban PM10 and PM2.5 pollution episodes in very stable nocturnal conditions and complex terrain using WRF-Chem CO tracer model. *Atmos. Environ.* *45*, 16 (2011), 2769–2780.
- [103] SANTOS-ALAMILLOS, F., POZO-VÁZQUEZ, D., RUIZ-ARIAS, J., AND TOVAR-PESCADOR, J. Influence of land-use misrepresentation on the accuracy of WRF wind estimates: Evaluation of GLCC and CORINE land-use maps in Southern Spain. *Atmos. Res.* *157* (2015), 17–28.
- [104] SCHWEIGER, E. W., BOLGRIEN, D. W., ANGRADI, T. R., AND KELLY, J. R. Environmental monitoring and assessment of a great river ecosystem: the upper Missouri River pilot. *Environ. Monit. Assess.* *103*, 1-3 (2005), 21–40.
- [105] SERTEL, E., ROBOCK, A., AND ORMECI, C. Impacts of land cover data quality on regional climate simulations. *Int. J. Climatol.* *30*, 13 (2010), 1942–1953.
- [106] SKAMAROCK, W. C., KLEMP, J. B., DUDHIA, J., GILL, D. O., BARKER, D. M., HUANG, X. Y., WANG, W., AND POWERS, J. G. A description of the advanced research WRF version 3. Tech. rep., National Center For Atmospheric Research Boulder Co Mesoscale and Microscale Div., 2008.
- [107] SMITH, E. N., GIBBS, J. A., FEDOROVICH, E., AND KLEIN, P. M. Wrf Model study of the Great Plains low-level jet: Effects of grid spacing and boundary layer parameterization. *J. Appl. Meteorol. Climatol.* *57*, 10 (2018), 2375–2397.
- [108] STAHL, K., TALLAKSEN, L. M., GUDMUNDSSON, L., AND CHRISTENSEN, J. H. Streamflow data from small basins: A challenging test to high-resolution regional climate modeling. *J. Hydrometeorol.* *12*, 5 (2011), 900–912.
- [109] STORM, B., DUDHIA, J., BASU, S., SWIFT, A., AND GIAMMANCO, I. Evaluation of the Weather Research and Forecasting model on forecasting low-level jets: Implications for wind energy. *Wind Energy* *12*, 1 (2009), 81–90.
- [110] STULL, R. B. *An introduction to boundary layer meteorology*, vol. 13. Springer Science & Business Media, 2012.
- [111] SUBIN, Z. M., RILEY, W. J., AND MIRONOV, D. An improved lake model for climate simulations: Model structure, evaluation, and sensitivity analyses in CESM1. *J. Adv. Model. Earth Sy.* *4*, 1 (2012), M02001.

- [112] TALBOT, C., BOU-ZEID, E., AND SMITH, J. Nested mesoscale large-eddy simulations with WRF: performance in real test cases. *J. Hydrometeorol.* *13*, 5 (2012), 1421–1441.
- [113] TAO, Z., SANTANELLO, J., CHIN, M., ZHOU, S., TAN, Q., KEMP, E., AND PETERS-LIDARD, C. Effect of land cover on atmospheric processes and air quality over the continental United States - A NASA unified WRF (NU-WRF) model study. *Atmos. Chem. Phys.* *13*, 2 (2013).
- [114] TAYLOR, C. M., LAMBIN, E. F., STEPHENNE, N., HARDING, R. J., AND ESSERY, R. L. The influence of land use change on climate in the Sahel. *J. Climate* *15*, 24 (2002), 3615–3629.
- [115] UDINA, M., SOLER, M. R., AND SOL, O. A modeling study of a trapped lee-wave event over the Pyrénées. *Mon. Weather Rev.* *145*, 1 (2017), 75–96.
- [116] VOGT, S., AND THOMAS, P. Test of a phased array sodar by intercomparison with tower data. *J. Atmos. Ocean. Tech.* *11*, 1 (1994), 94–102.
- [117] WALKER, H., LEONE JR, J., AND KIM, J. The effects of elevation data representation on mesoscale atmospheric model simulations. Tech. rep., Lawrence Livermore National Lab., CA (United States), 1996.
- [118] WALPOLE, R. E., MYERS, R. E., MYERS, S. L., AND YE, K. *Probability & Statistics For Engineers & Scientists*, 7 ed. Prentice Hall, Upper Saddle River, NJ 07458, 2002.
- [119] WANG, H., SKAMAROCK, W. C., AND FEINGOLD, G. Evaluation of scalar advection schemes in the Advanced Research WRF model using large-eddy simulations of aerosol-cloud interactions. *Mon. Weather Rev.* *137*, 8 (2009), 2547–2558.
- [120] WANG, W., BARKER, D., BRAY, J., BRUYERE, C., DUDA, M., DUDHIA, J., GILL, D., AND MICHALAKES, J. Users guide for advanced research WRF (ARW) modeling system version 3. Tech. rep., 2007.
- [121] WANG, Z., DUAN, A., AND WU, G. Impacts of boundary layer parameterization schemes and air-sea coupling on WRF simulation of the East Asian summer monsoon. *Science China Earth Sciences* *57*, 7 (2014), 1480–1493.
- [122] WYNGAARD, J. C. *Turbulence in the Atmosphere*. Cambridge University Press, 2010.
- [123] XIE, B., FUNG, J. C., CHAN, A., AND LAU, A. Evaluation of nonlocal and local planetary boundary layer schemes in the WRF model. *J. Geophys. Res.- Atmos.* *117*, D12 (2012).
- [124] XU, L., LIU, H., DU, Q., AND WANG, L. Evaluation of the WRF-lake model over a highland freshwater lake in Southwest China. *J. Geophys. Res.-Atmos.* *121*, 23 (2016).

- [125] XUE, Y., JANJIC, Z., DUDHIA, J., VASIC, R., AND DE SALES, F. A review on regional dynamical downscaling in intraseasonal to seasonal simulation/prediction and major factors that affect downscaling ability. *Atmos. Res.* 147 (2014), 68–85.
- [126] YAN, H., AND GALLUS JR, W. A. An evaluation of QPF from the WRF, NAM, and GFS models using multiple verification methods over a small domain. *Weather Forecast.* 31, 4 (2016), 1363–1379.
- [127] YÁÑEZ-MORRONI, G., GIRONÁS, J., CANEO, M., DELGADO, R., AND GARREAUD, R. Using the Weather Research and Forecasting (WRF) model for precipitation forecasting in an Andean region with complex topography. *Atmosphere* 9, 8 (2018), 304.
- [128] YANG, Z.-L. Modeling land surface processes in short-term weather and climate studies. In *Observation, Theory And Modeling Of Atmospheric Variability*. World Scientific, 2004, pp. 288–313.
- [129] YU, Z. Regional Climate Modeling over Ontario using the WRF model. Master’s thesis, York University, Toronto, Ontario, August 2013.
- [130] ZÄNGL, G., GANTNER, L., HARTJENSTEIN, G., AND NOPPEL, H. Numerical errors above steep topography: A model intercomparison. *Meteor. Z.* 13, 2 (2004), 69–76.
- [131] ZHANG, C., LIN, H., CHEN, M., AND YANG, L. Scale matching of multiscale digital elevation model (DEM) data and the Weather Research and Forecasting (WRF) model: a case study of meteorological simulation in Hong Kong. *Arab. J. of Geosci.* 7, 6 (2014), 2215–2223.
- [132] ZHANG, D.-L., AND ZHENG, W.-Z. Diurnal cycles of surface winds and temperatures as simulated by five boundary layer parameterizations. *J. Appl. Meteorol.* 43, 1 (2004), 157–169.
- [133] ZHENG, Y., ALAPATY, K., HERWEHE, J. A., DEL GENIO, A. D., AND NIYOGI, D. Improving high-resolution weather forecasts using the Weather Research and Forecasting (WRF) Model with an updated Kain-Fritsch scheme. *Mon. Weather Rev.* 144, 3 (2016), 833–860.
- [134] ZHOU, B., SIMON, J. S., AND CHOW, F. K. The convective boundary layer in the terra incognita. *J. Atmos. Sci.* 71, 7 (2014), 2545–2563.
- [135] ZHU, R., AND XU, D. Multi-scale turbulent planetary boundary layer parameterization in mesoscale numerical simulation. *Quarterly Journal of Applied Meteorology* 15, 5 (2004), 543–555.



# Appendix A

## How to Operate the PA-5 SODAR

### A.1 Turning on the SODAR

- The hardware key should be inserted in the SODAR laptop.
- The power switch of the laptop should be pressed to turn on the laptop.
- When the boot-up process is completed, a terminal window called **Start Sodar** is automatically opened.
- Finally after the calibration cycle the SODAR will start beeping automatically.

### A.2 Setting the Date and Time

- After opening the terminal, type **su -**.
- Then the password should be inserted.
- At the prompt type **date MMddhhmmYYYY**. For example, to set date and time as July 30 2019 12 hour 39 minute, it should be written as **date 073012392019**.
- Finally type **exit**.

### A.3 Changing the Averaging Time

- The **Sodter** program should be opened either from the bottom of the **Start Sodar** screen or from the desktop.

- Then type **LIST** and press the Enter key to get a list of current parameters.
- To change the averaging time type **SMATIM**. The parameter will appear with current values by a message asking for a new value.
- To set the averaging time to 30 minutes type **0:30:0** and then press the Enter key.

## A.4 Saving the SODAR Data on a USB Flash Memory

- After opening the terminal, type **cd /usr/src/sodar** and press Enter key to go to the SODAR directory.
- Then the flash memory card should be inserted in the available USB port.
- To mount the flash memory type **mk**.
- To save a file, for example, the 20190730.dat file type **cp 20190730.dat /mnt/usb1/20190730.dat** and then press the Enter key.
- Before unplugging the flash memory to unmount it type **uk** and then press the Enter key.
- After getting the prompt back, the USB flash memory can be unplugged.

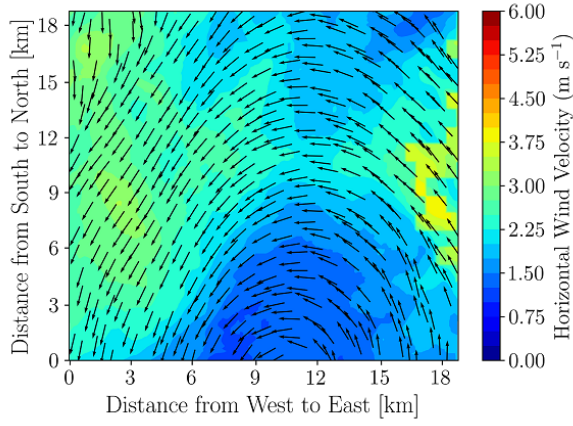
## A.5 Turning off the SODAR

- The SODAR must be paused before turning off the SODAR.
- The **Sodter** program should be opened either from the bottom of the **Start Sodar** screen or from the desktop.
- When the **Start Sodter** screen is popped up, type **pause**.
- It will ask to insert the number of minutes. For example, to pause the SODAR for 20 minutes, it should be written as **20** and then the enter key should be pressed.
- To restart the SODAR in any case type **GO**.
- After pausing the SODAR, the **Stop computer** icon should be clicked either from the bottom of the **Start Sodter** screen or from the desktop to turn off the SODAR system.

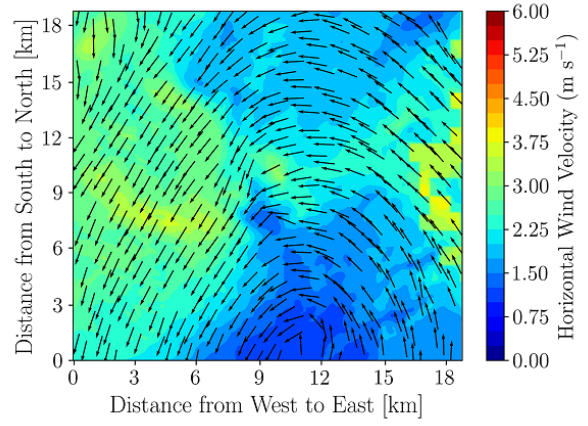
## Appendix B

# Spatial Distributions of Various Atmopsheric Properties over a Mining Facility in Summer 2018

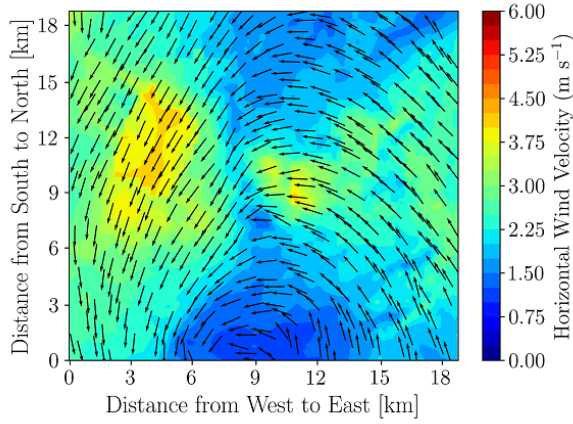
## B.1 Summer: Horizontal Wind Velocity at 10 m



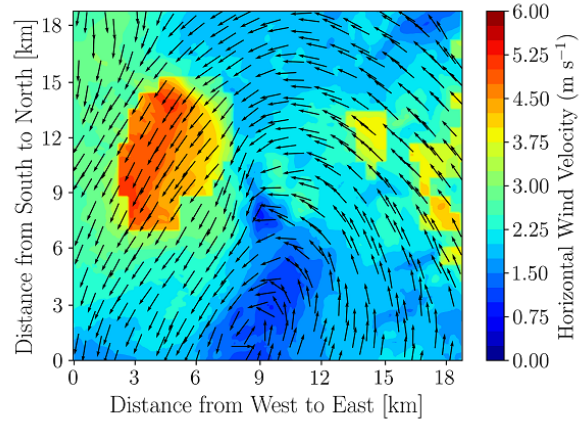
(a) GTOPO



(b) SRTM-LIDAR

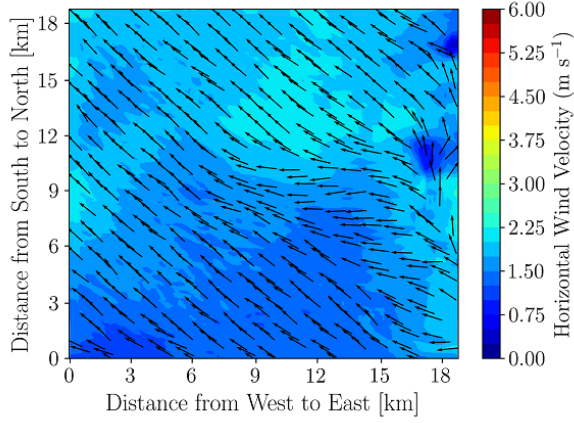


(c) SRTM-LIDAR with land use

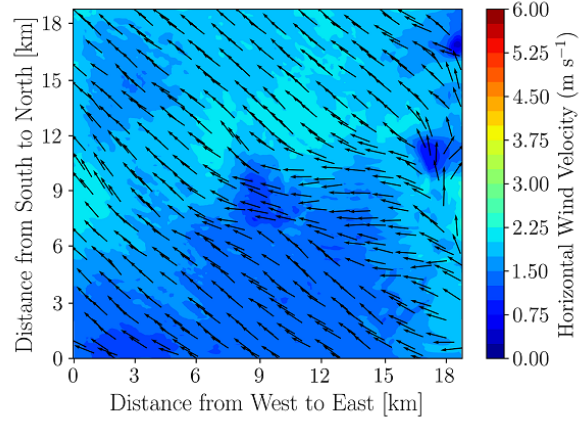


(d) SRTM-LIDAR with land use and lake model

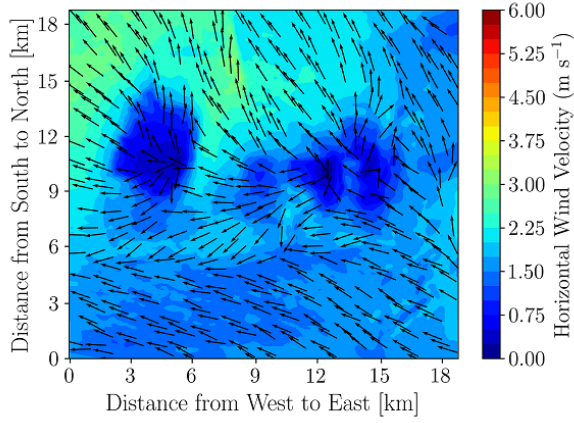
Figure B.1: Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 0600 MST in domain 5.



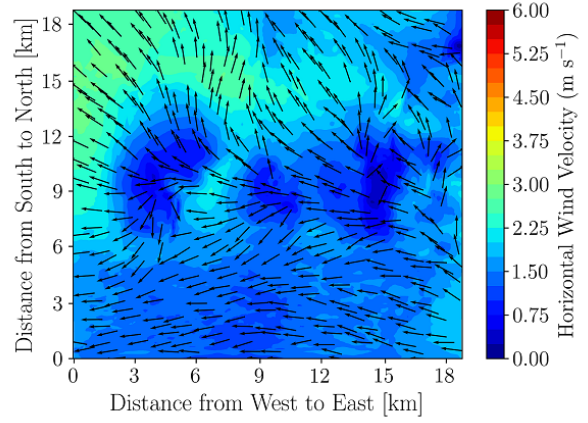
(a) GTOPO



(b) SRTM-LIDAR

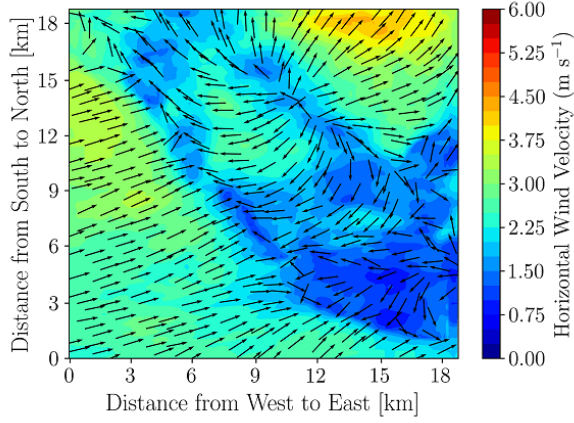


(c) SRTM-LIDAR with land use

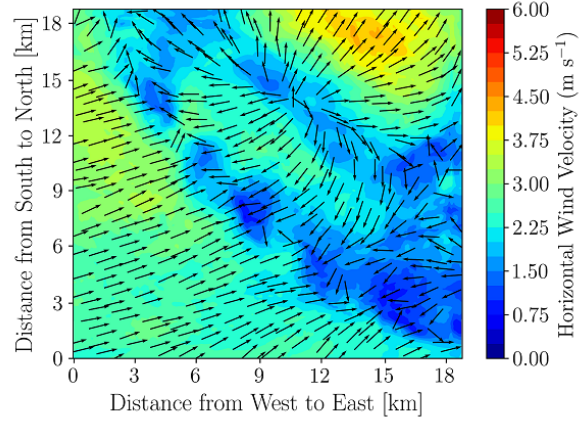


(d) SRTM-LIDAR with land use and lake model

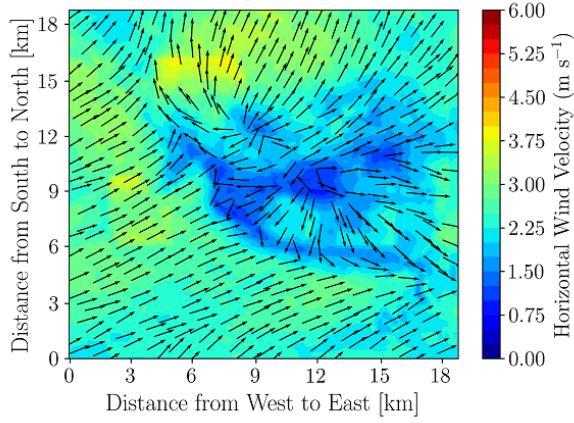
Figure B.2: Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 1000 MST in domain 5.



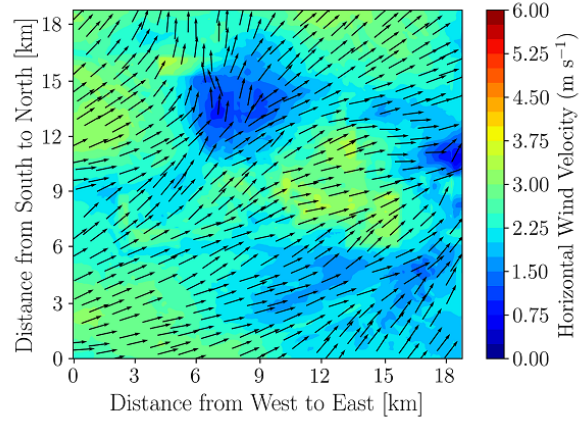
(a) GTOPO



(b) SRTM-LIDAR



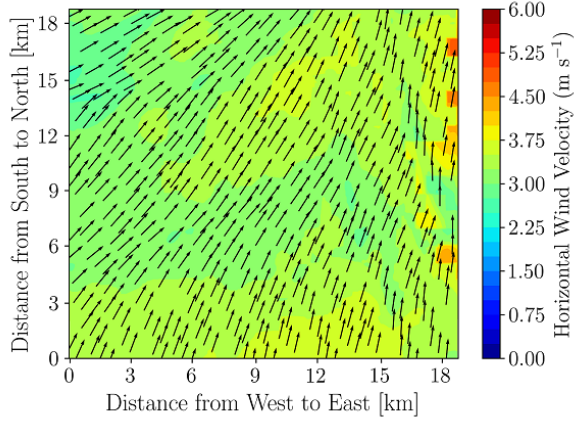
(c) SRTM-LIDAR with land use



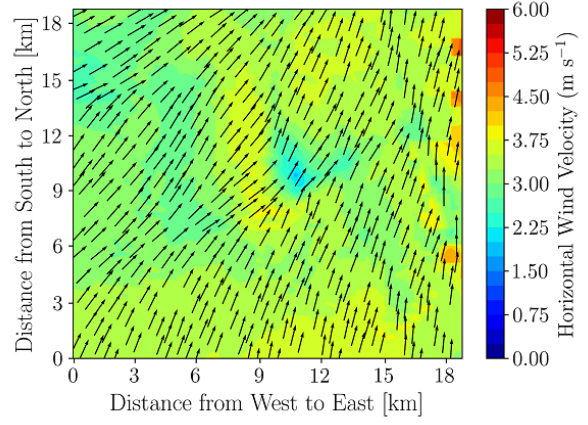
(d) SRTM-LIDAR with land use and lake model

Figure B.3: Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 1800 MST in domain 5.

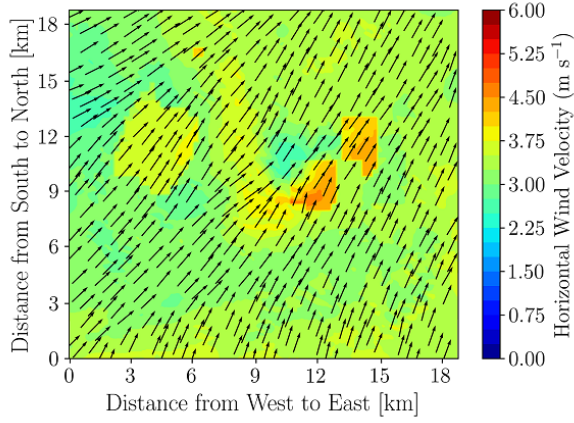




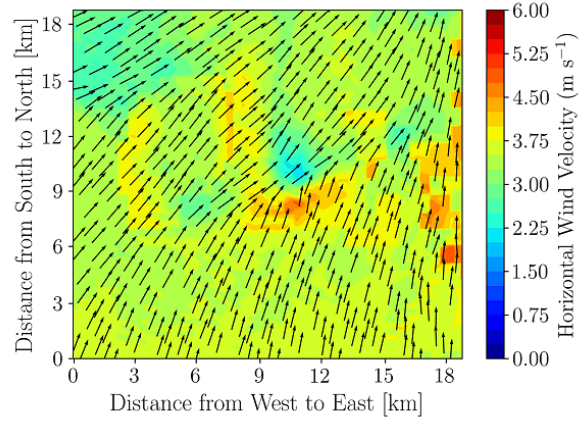
(a) GTOPO



(b) SRTM-LIDAR



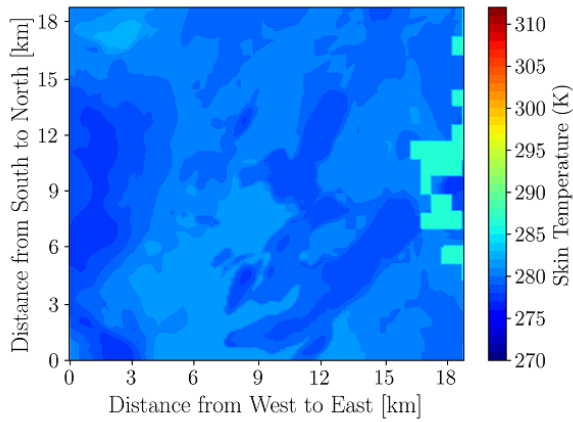
(c) SRTM-LIDAR with land use



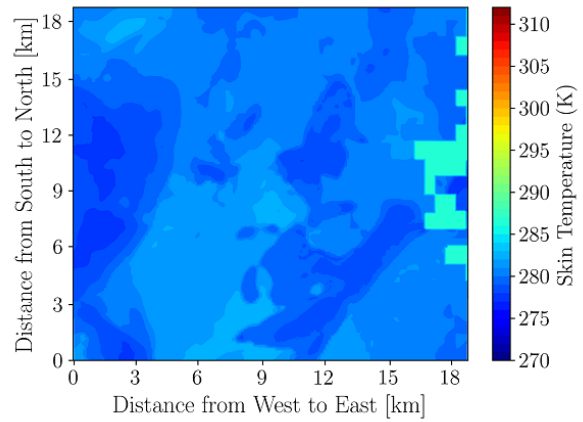
(d) SRTM-LIDAR with land use and lake model

Figure B.4: Effect of topography and land use changes on 10-m horizontal wind velocity magnitude and direction at 2200 MST in domain 5.

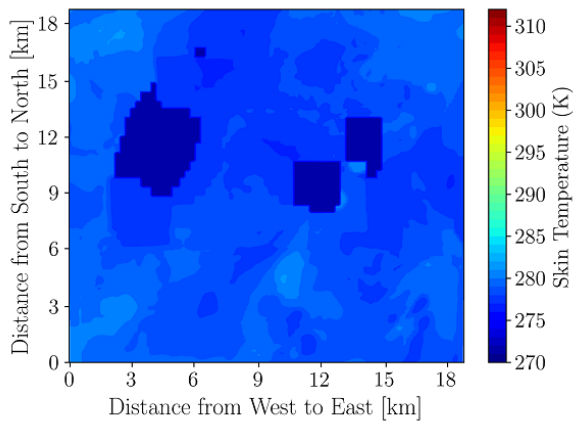
## B.2 Summer: Surface Skin Temperature



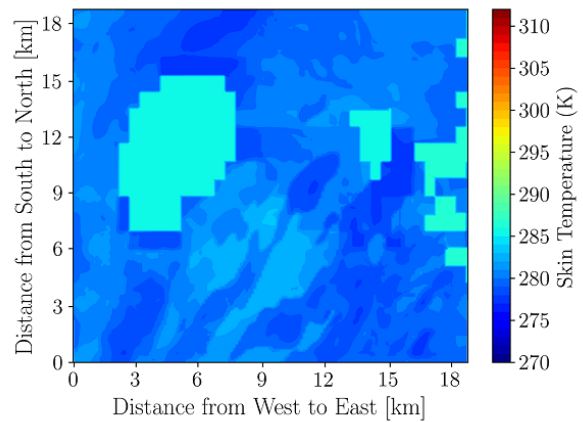
(a) GTOPO



(b) SRTM-LIDAR



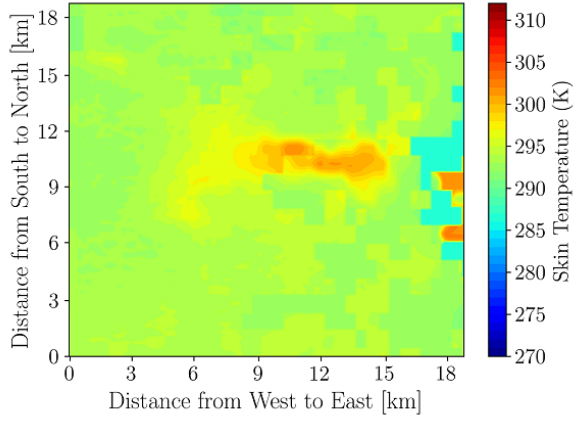
(c) SRTM-LIDAR with land use



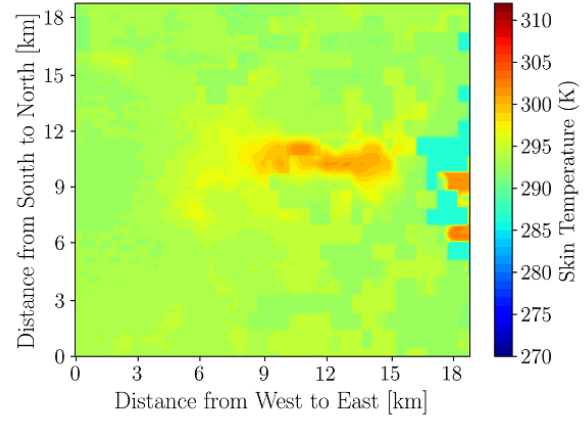
(d) SRTM-LIDAR with land use and lake model

Figure B.5: Effect of topography and land use changes on surface skin temperature at 0600 MST in domain 5 on May 18, 2018.

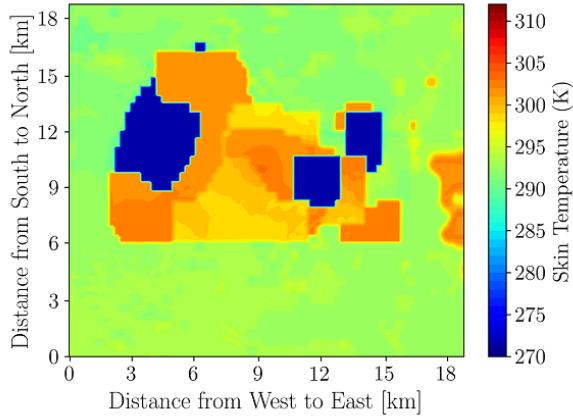




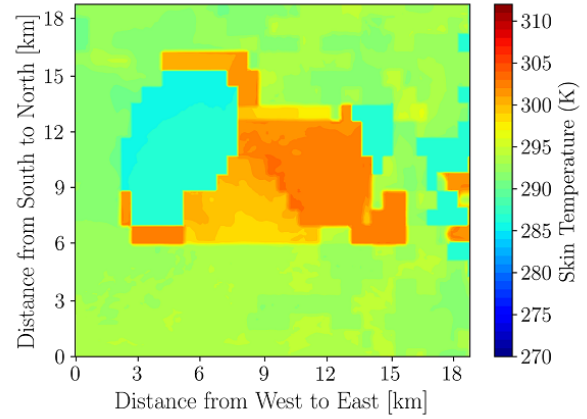
(a) GTOPO



(b) SRTM-LIDAR

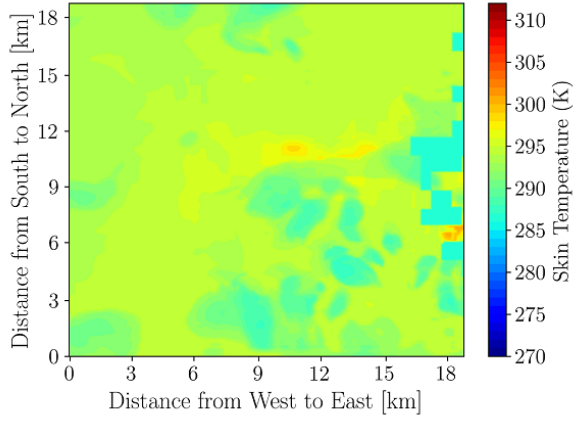


(c) SRTM-LIDAR with land use

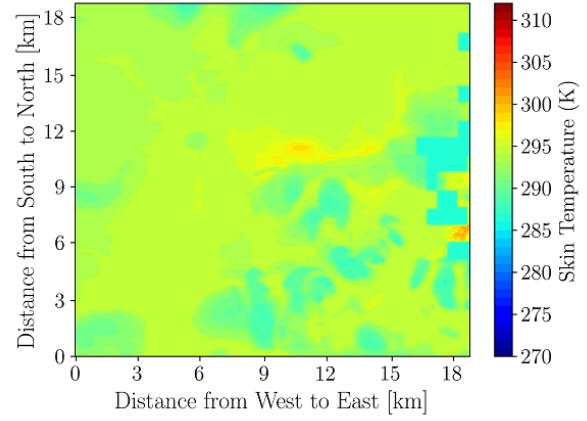


(d) SRTM-LIDAR with land use and lake model

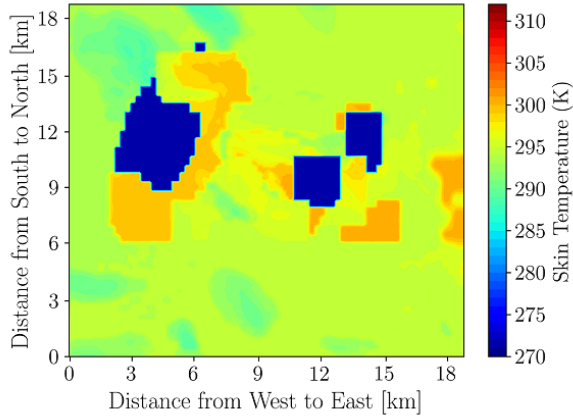
Figure B.6: Effect of topography and land use changes on surface skin temperature at 1000 MST in domain 5 on May 18, 2018.



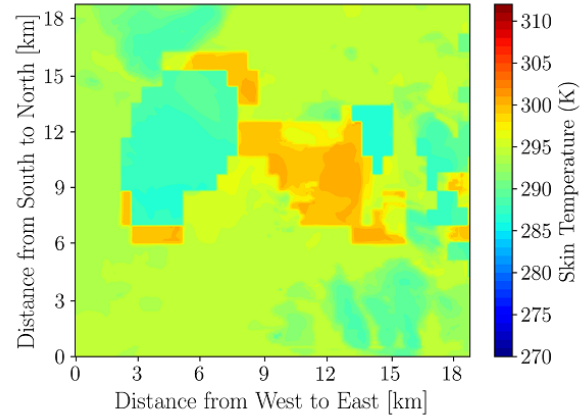
(a) GTOPO



(b) SRTM-LIDAR

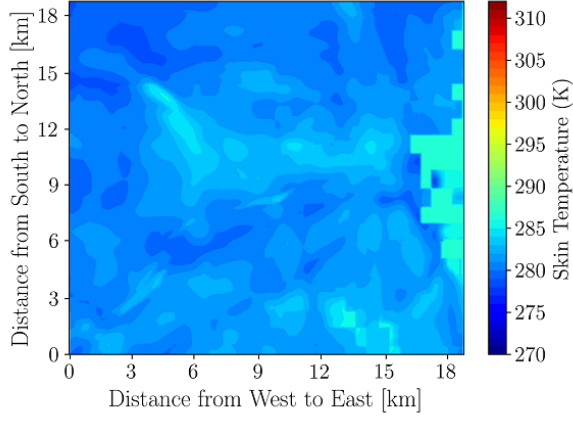


(c) SRTM-LIDAR with land use

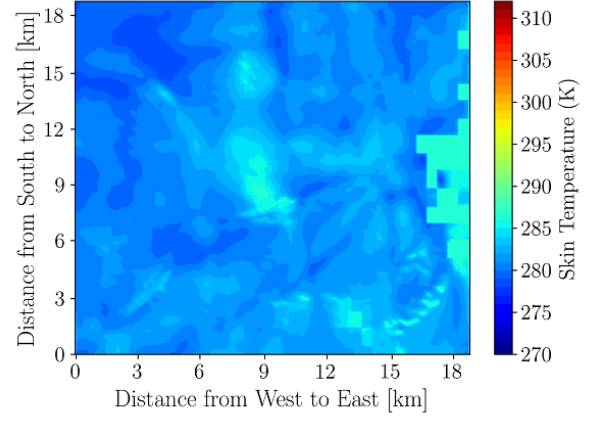


(d) SRTM-LIDAR with land use and lake model

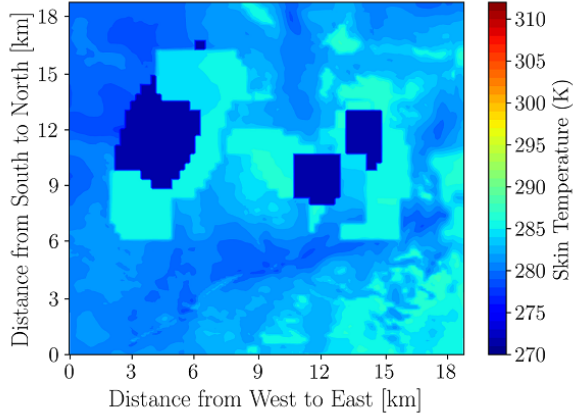
Figure B.7: Effect of topography and land use changes on surface skin temperature at 1800 MST in domain 5 on May 18, 2018.



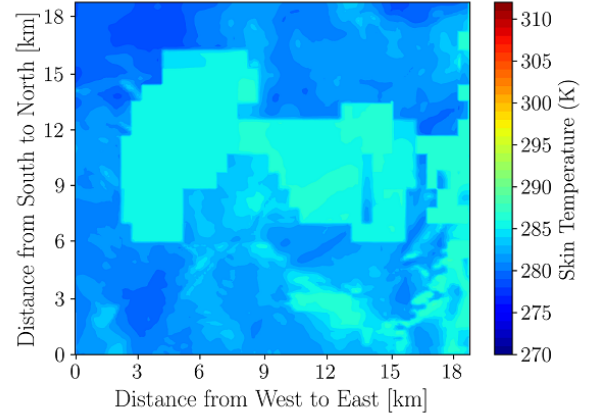
(a) GTOPO



(b) SRTM-LIDAR



(c) SRTM-LIDAR with land use



(d) SRTM-LIDAR with land use and lake model

Figure B.8: Effect of topography and land use changes on surface skin temperature at 2200 MST in domain 5 on May 18, 2018.

### B.3 Summer: Potential Temperature at 2 m

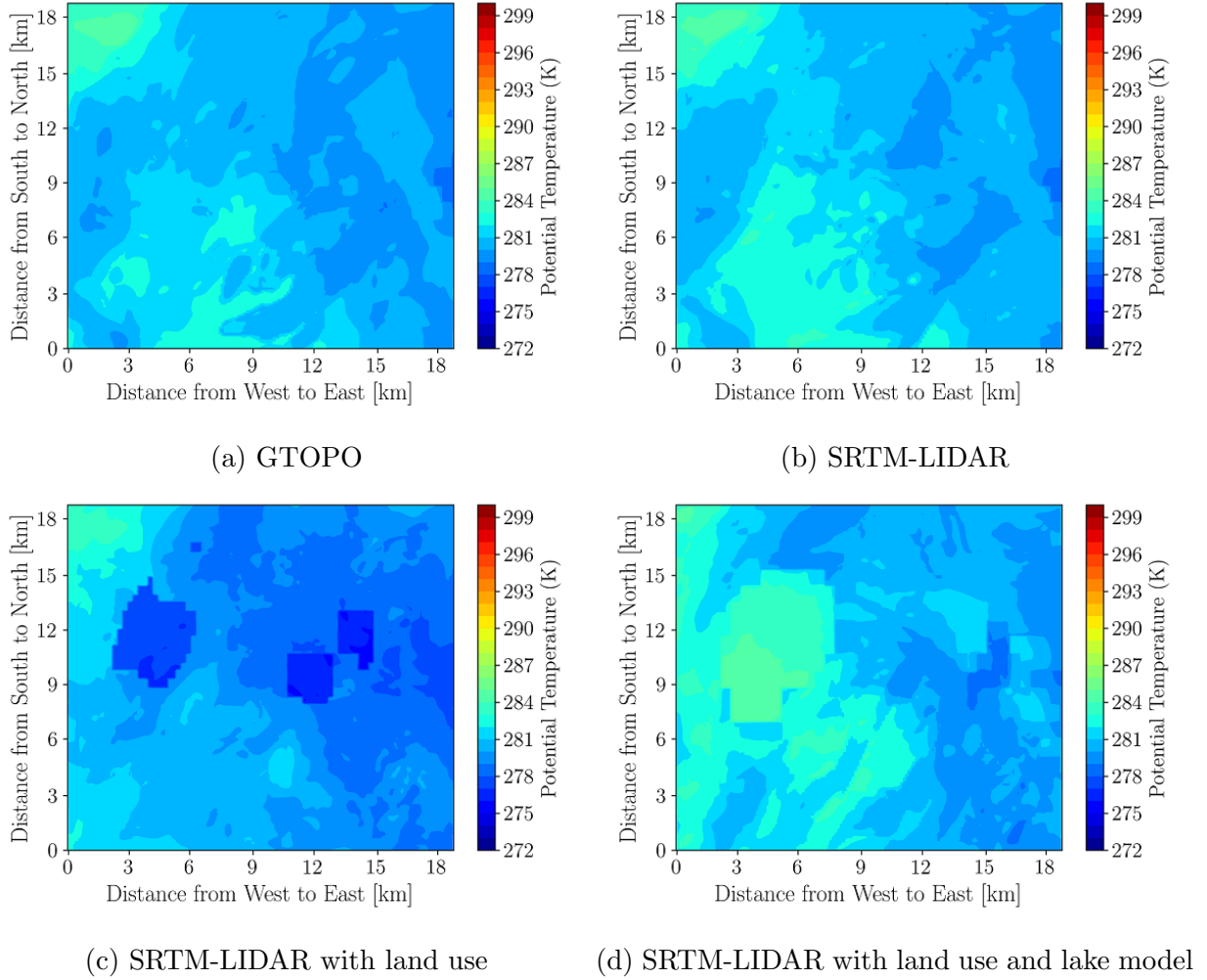
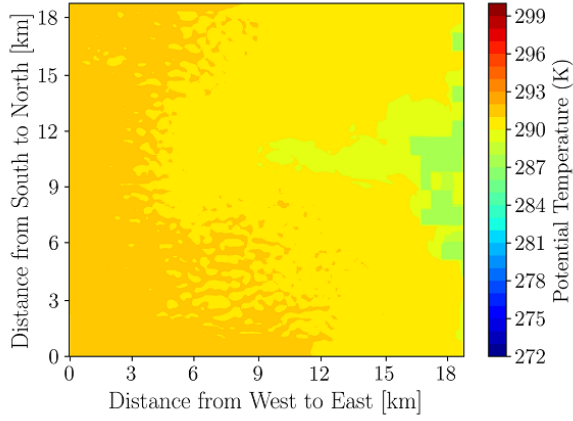
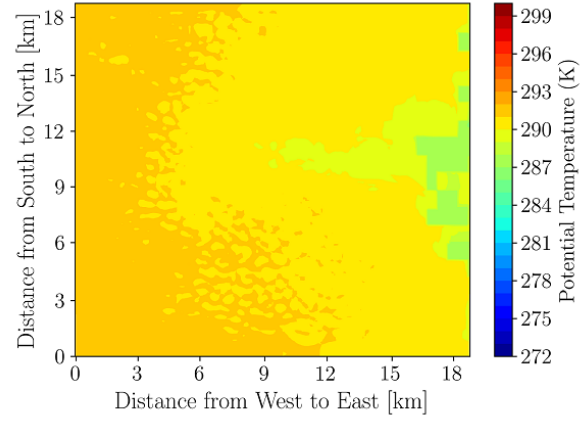


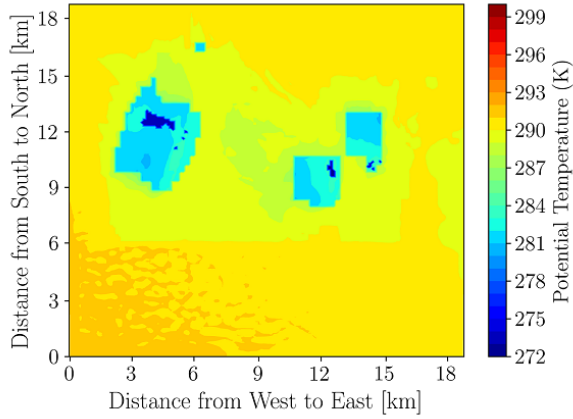
Figure B.9: Effect of topography and land use changes on 2-m potential temperature at 0600 MST in domain 5 on May 18, 2018.



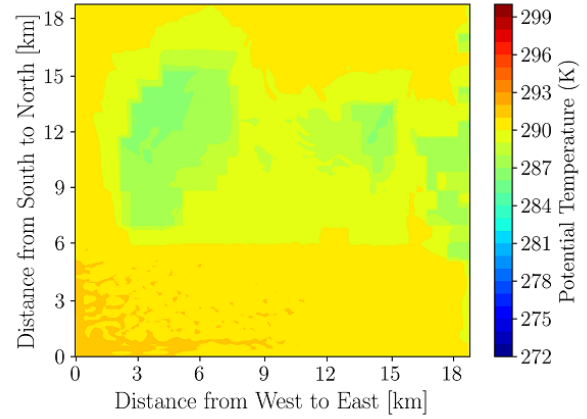
(a) GTOPO



(b) SRTM-LIDAR

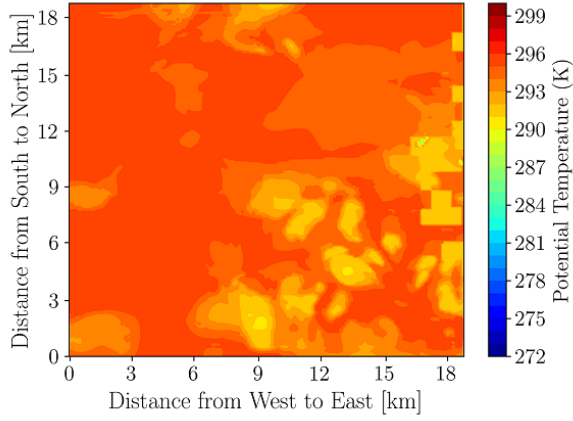


(c) SRTM-LIDAR with land use

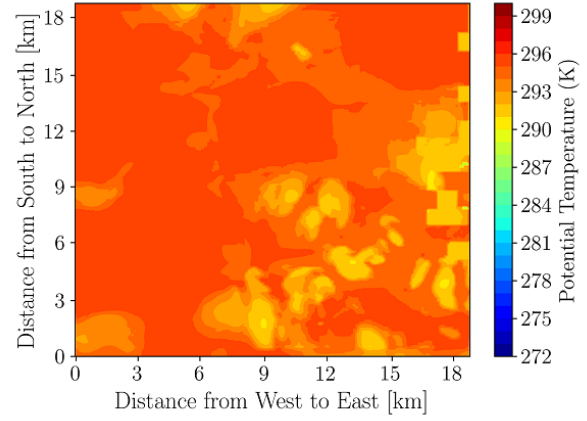


(d) SRTM-LIDAR with land use and lake model

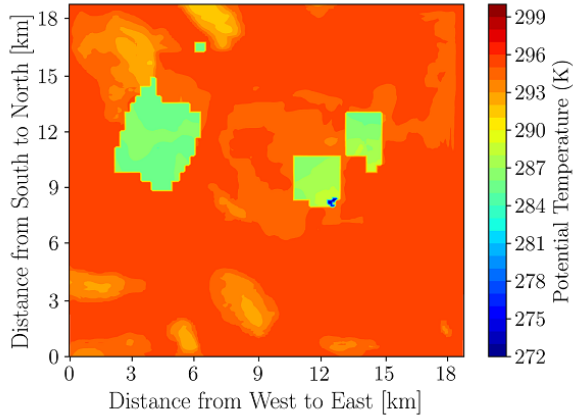
Figure B.10: Effect of topography and land use changes on 2-m potential temperature at 1000 MST in domain 5 on May 18, 2018.



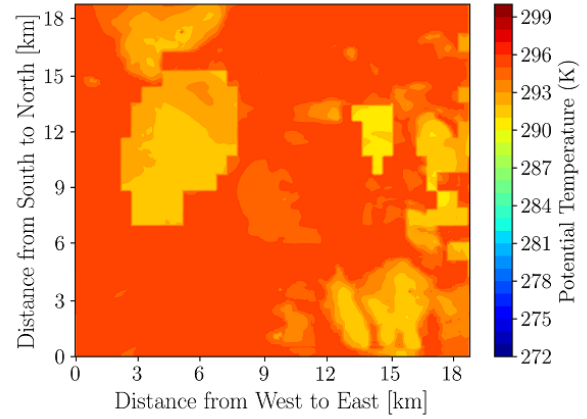
(a) GTOPO



(b) SRTM-LIDAR

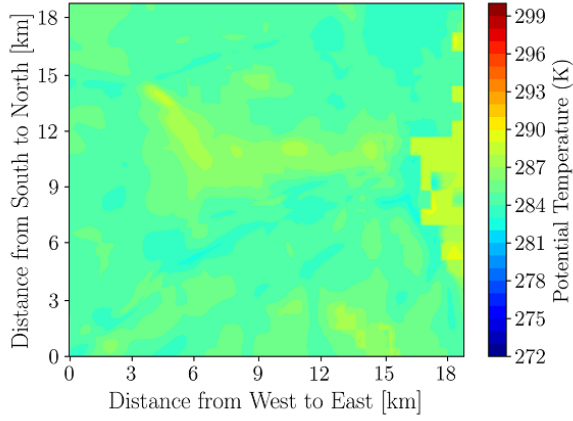


(c) SRTM-LIDAR with land use

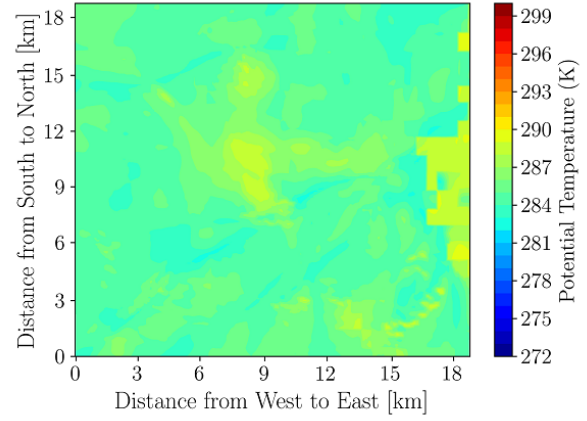


(d) SRTM-LIDAR with land use and lake model

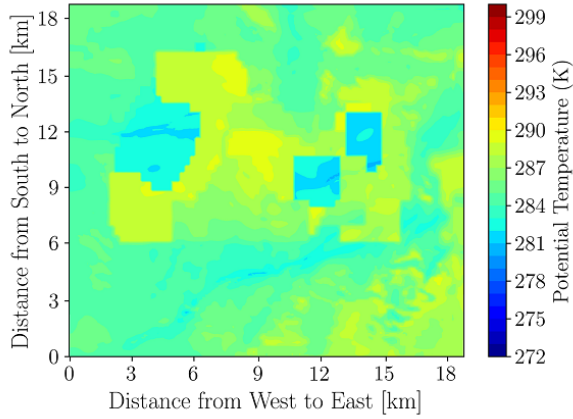
Figure B.11: Effect of topography and land use changes on 2-m potential temperature at 1800 MST in domain 5 on May 18, 2018.



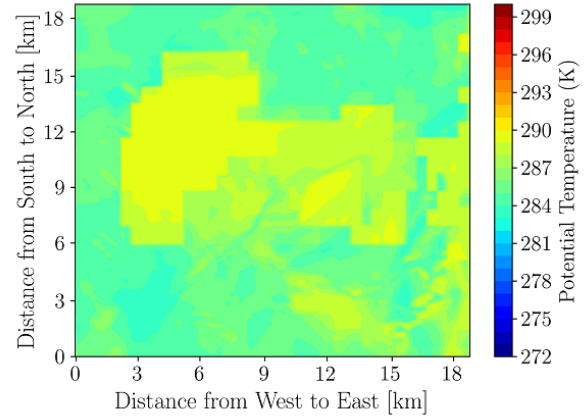
(a) GTOPO



(b) SRTM-LIDAR



(c) SRTM-LIDAR with land use



(d) SRTM-LIDAR with land use and lake model

Figure B.12: Effect of topography and land use changes on 2-m potential temperature at 2200 MST in domain 5 on May 18, 2018.

## B.4 Summer: Relative Humidity at 2 m

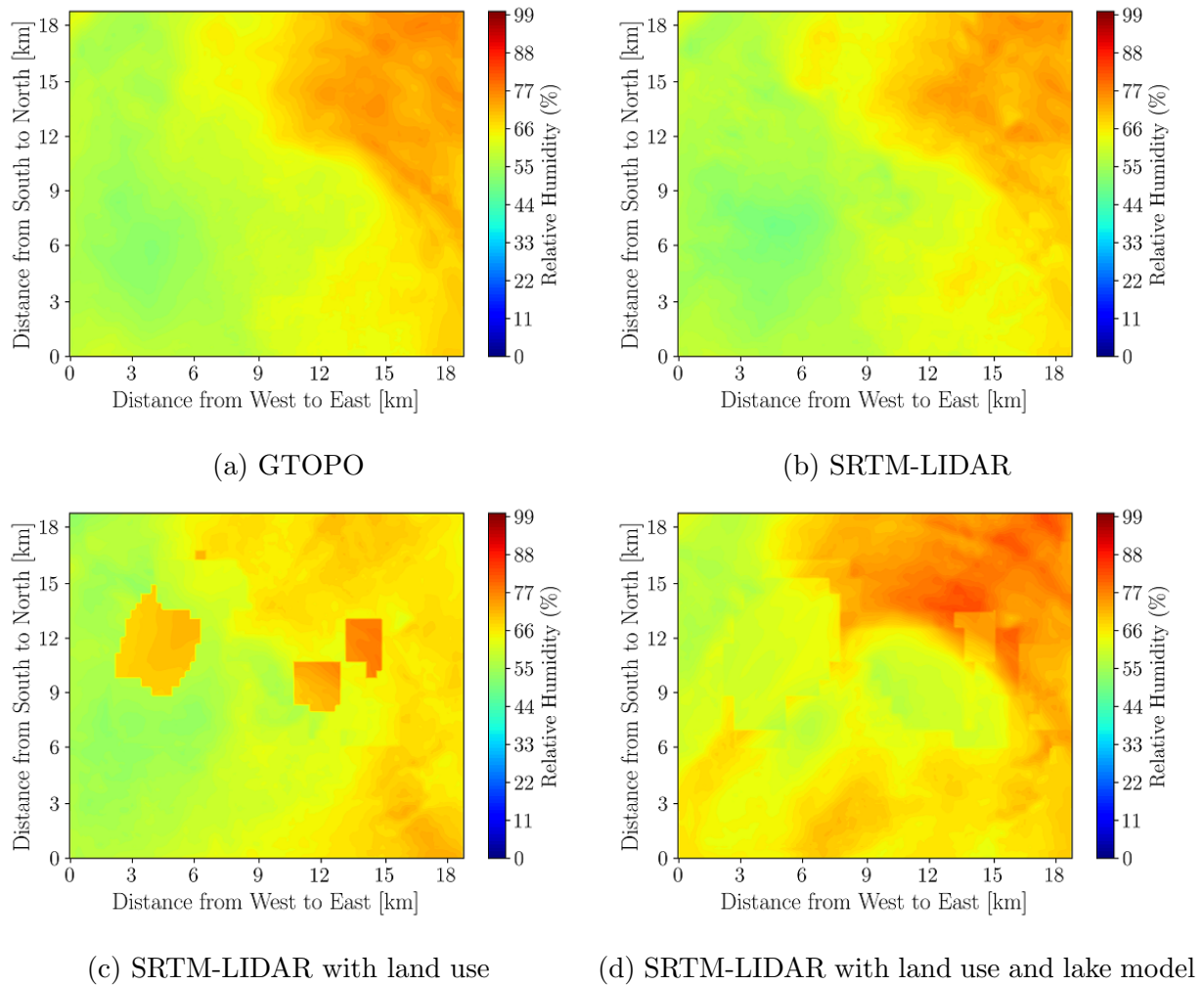
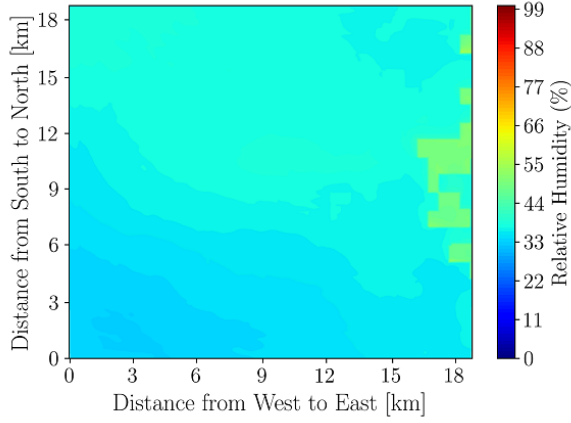
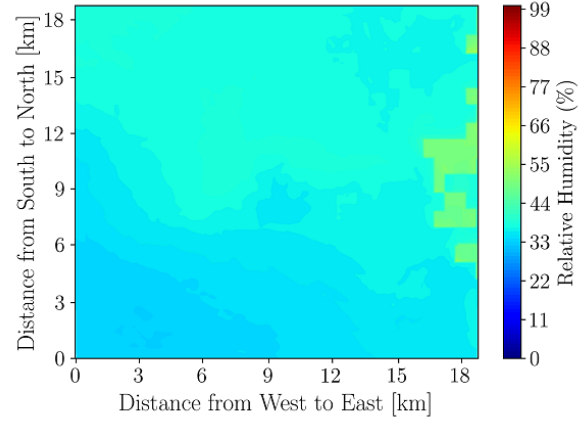


Figure B.13: Effect of topography and land use changes on 2-m relative Humidity at 0600 MST in domain 5 on May 18, 2018.

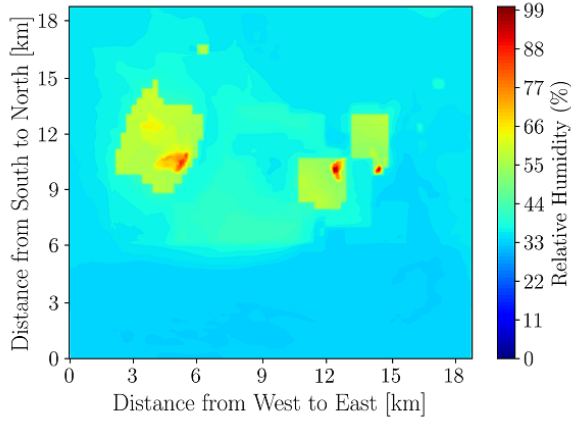




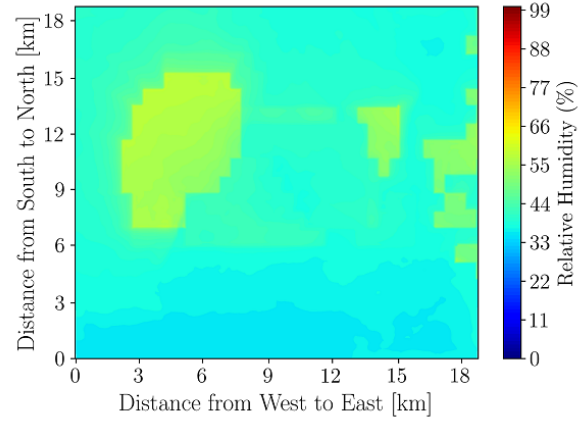
(a) GTOPO



(b) SRTM-LIDAR

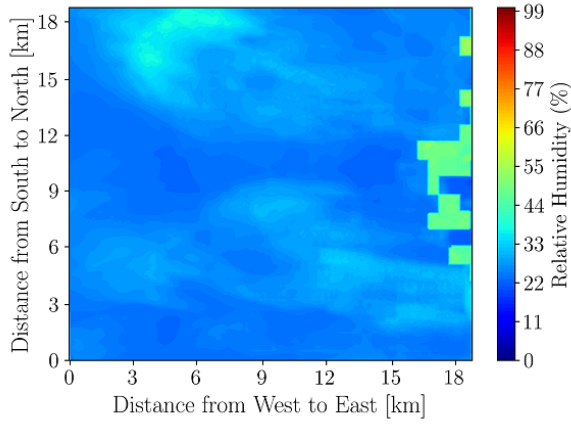


(c) SRTM-LIDAR with land use

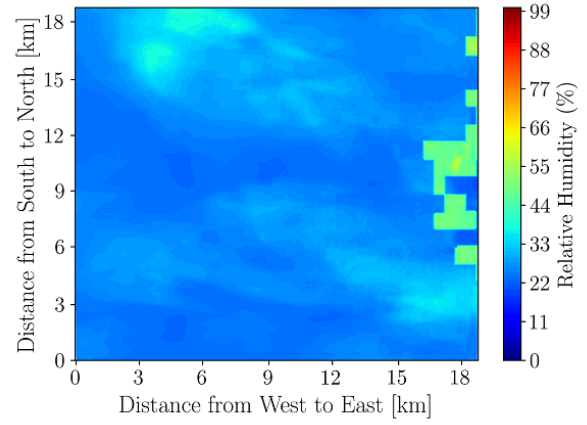


(d) SRTM-LIDAR with land use and lake model

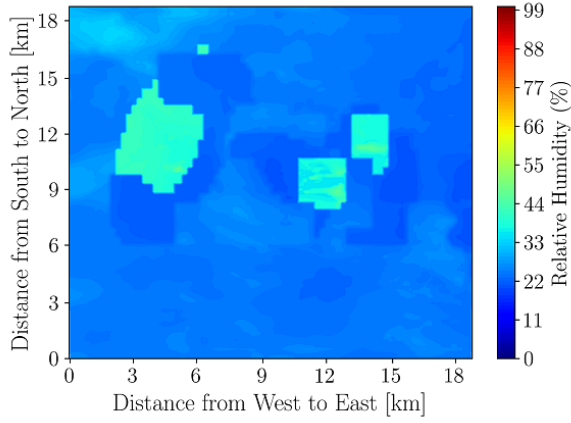
Figure B.14: Effect of topography and land use changes on 2-m relative Humidity at 1000 MST in domain 5 on May 18, 2018.



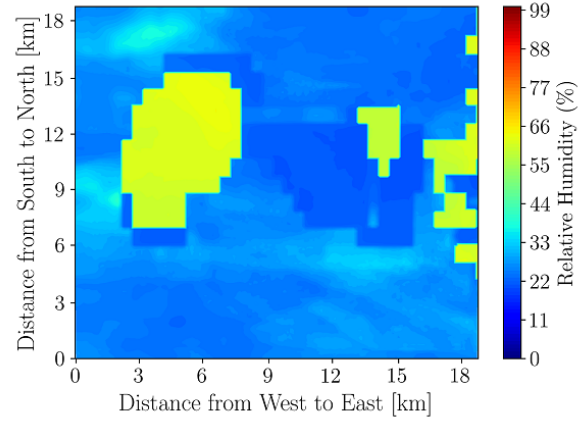
(a) GTOPO



(b) SRTM-LIDAR

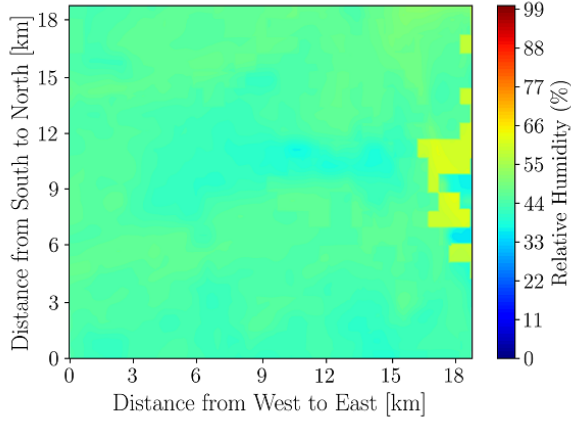


(c) SRTM-LIDAR with land use

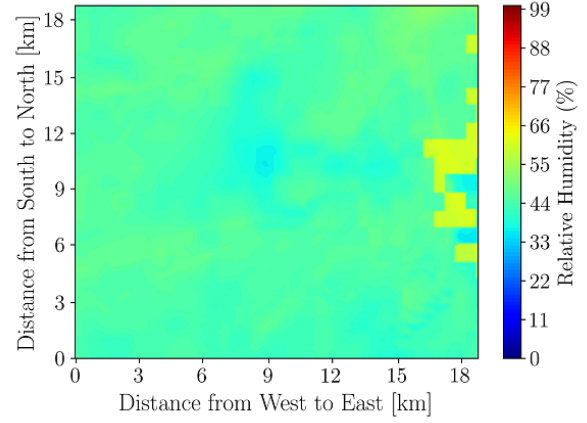


(d) SRTM-LIDAR with land use and lake model

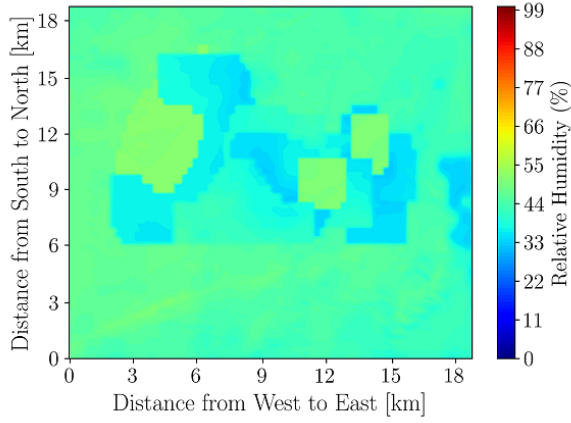
Figure B.15: Effect of topography and land use changes on 2-m relative Humidity at 1800 MST in domain 5 on May 18, 2018.



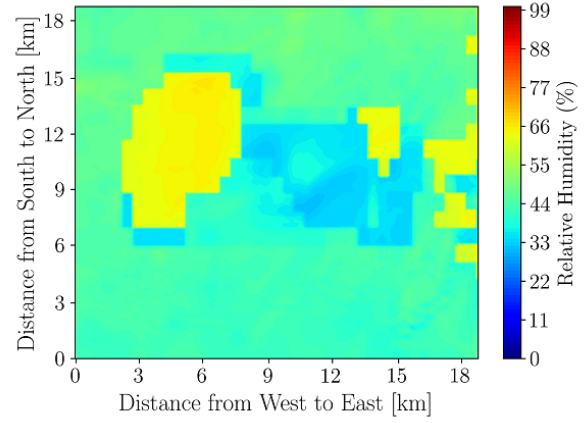
(a) GTOPO



(b) SRTM-LIDAR



(c) SRTM-LIDAR with land use



(d) SRTM-LIDAR with land use and lake model

Figure B.16: Effect of topography and land use changes on 2-m relative Humidity at 2200 MST in domain 5 on May 18, 2018.

## B.5 Summer: Sensible Vertical Turbulent Heat Flux at 10 m

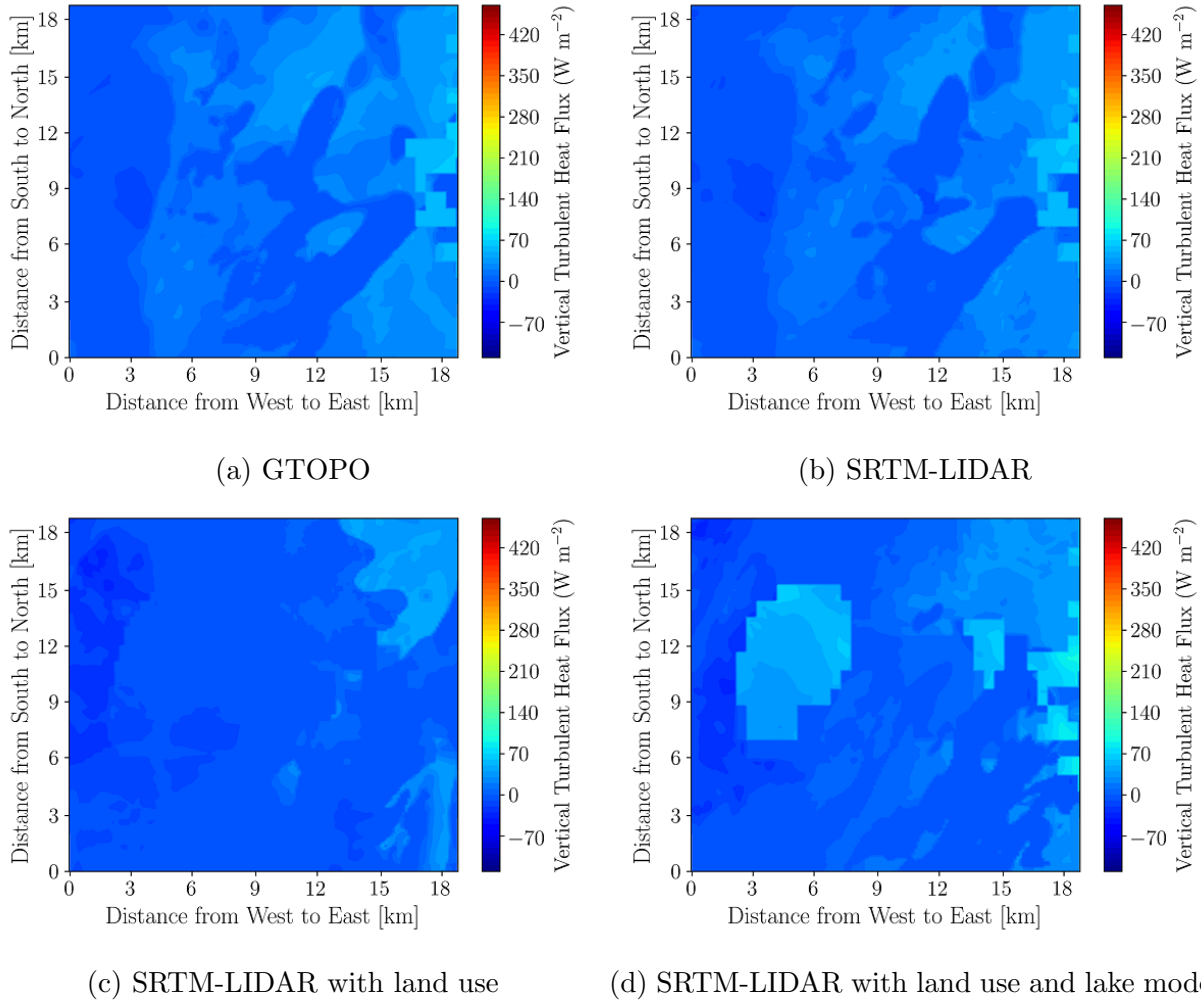
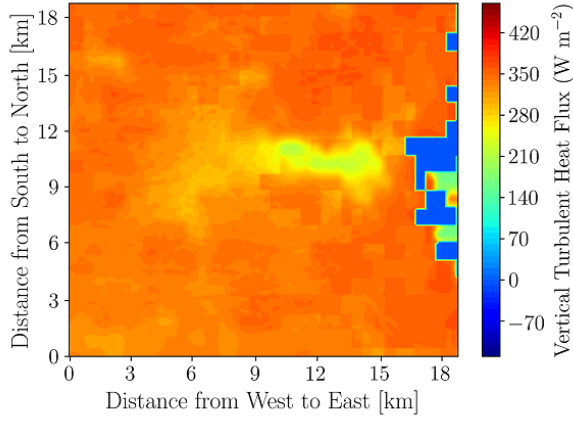
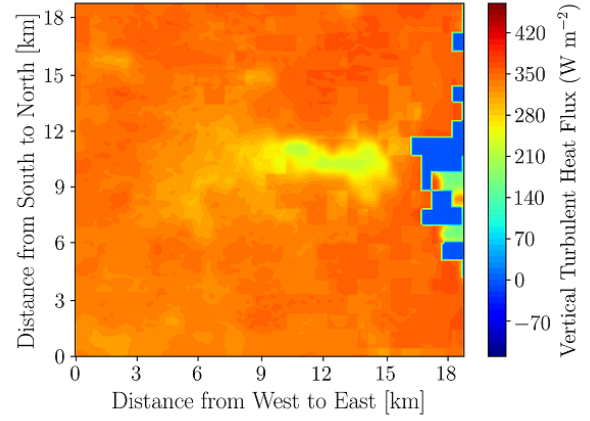


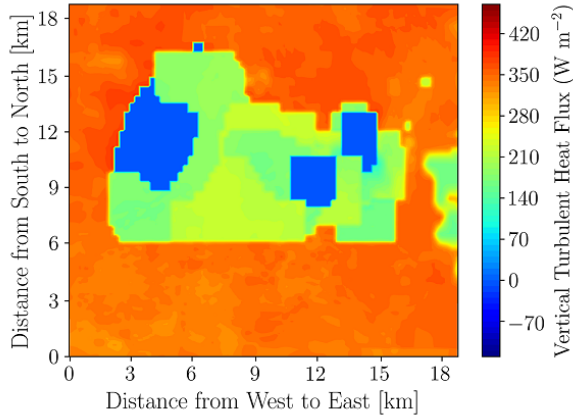
Figure B.17: Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 0600 MST in domain 5 on May 18, 2018.



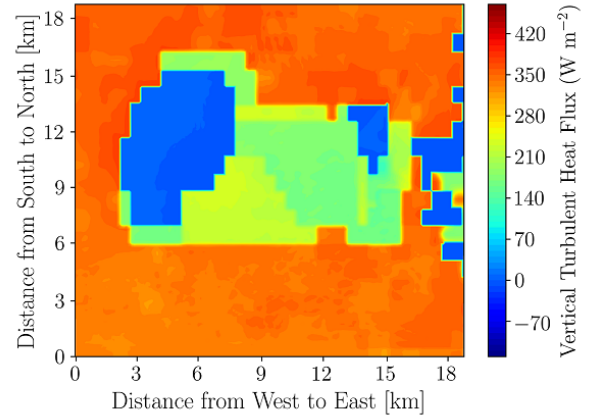
(a) GTOPO



(b) SRTM-LIDAR

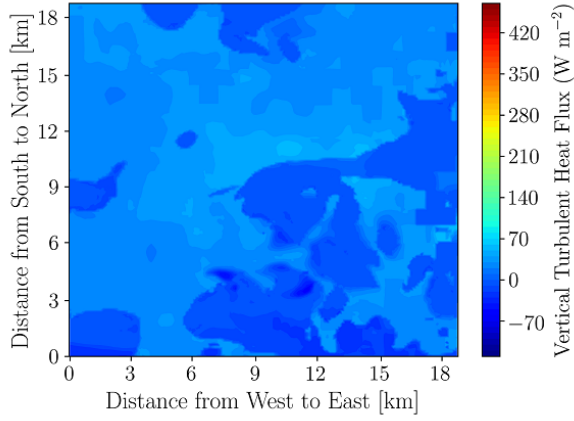


(c) SRTM-LIDAR with land use

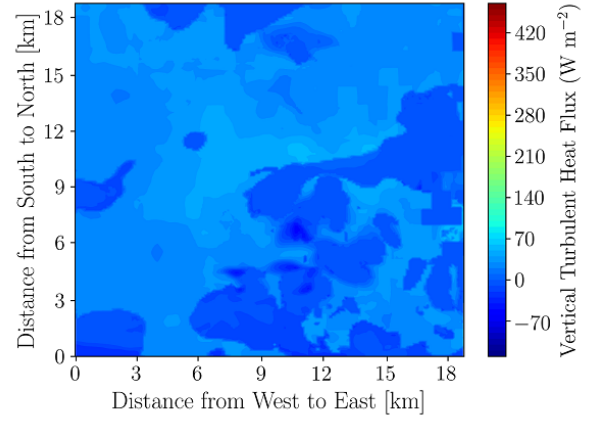


(d) SRTM-LIDAR with land use and lake model

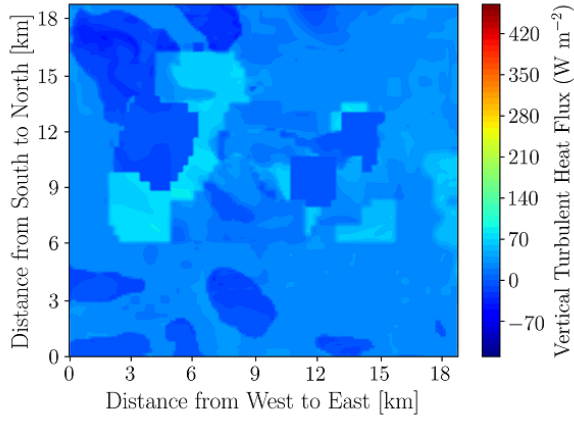
Figure B.18: Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 1000 MST in domain 5 on May 18, 2018.



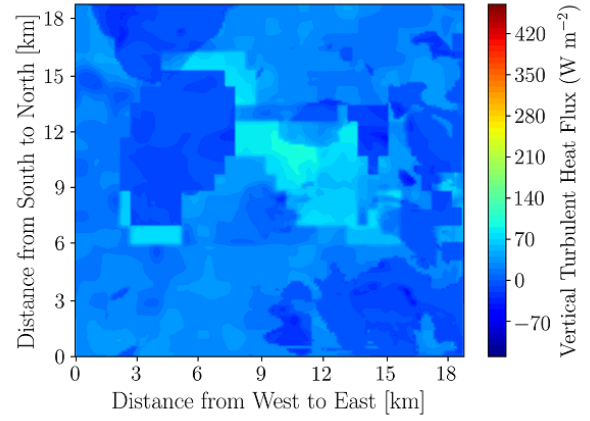
(a) GTOPO



(b) SRTM-LIDAR

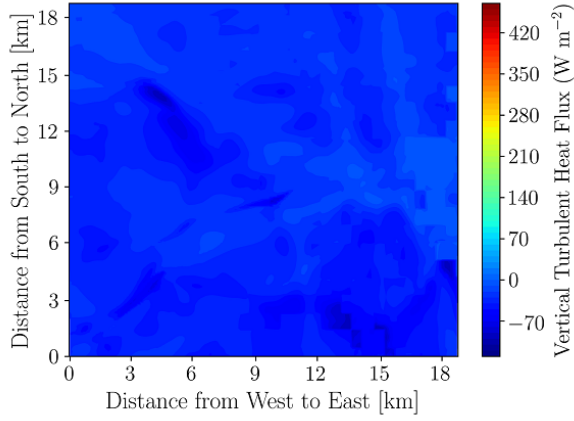


(c) SRTM-LIDAR with land use

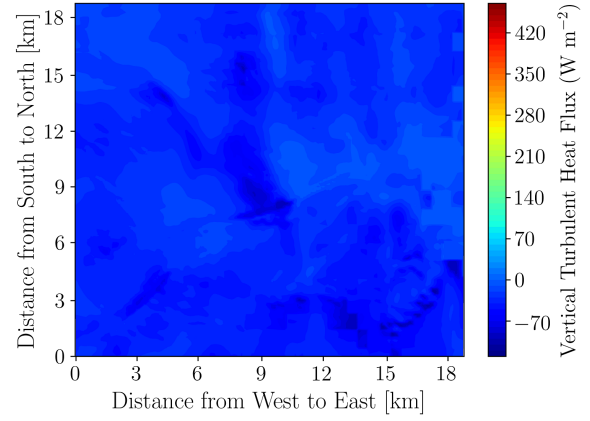


(d) SRTM-LIDAR with land use and lake model

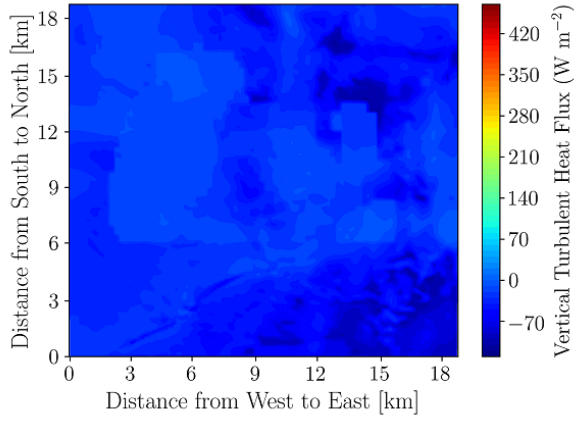
Figure B.19: Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 1800 MST in domain 5 on May 18, 2018.



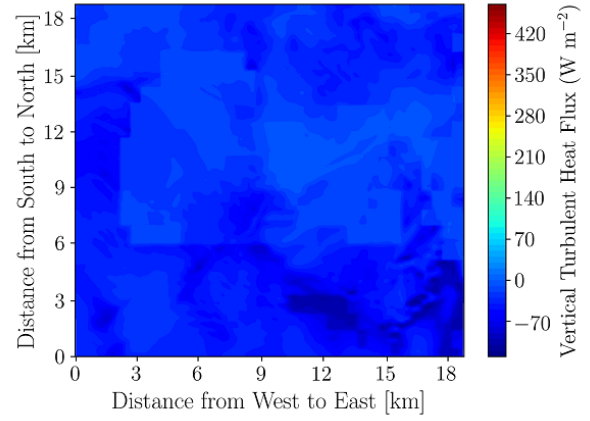
(a) GTOPO



(b) SRTM-LIDAR



(c) SRTM-LIDAR with land use



(d) SRTM-LIDAR with land use and lake model

Figure B.20: Effect of topography and land use changes on 10-m sensible vertical turbulent heat flux at 2200 MST in domain 5 on May 18, 2018.

## Appendix C

### Spatial Distributions of Various Atmospheric Properties over a Mining Facility in Winter 2019



## C.1 Winter: Horizontal Wind Velocity at 10 m

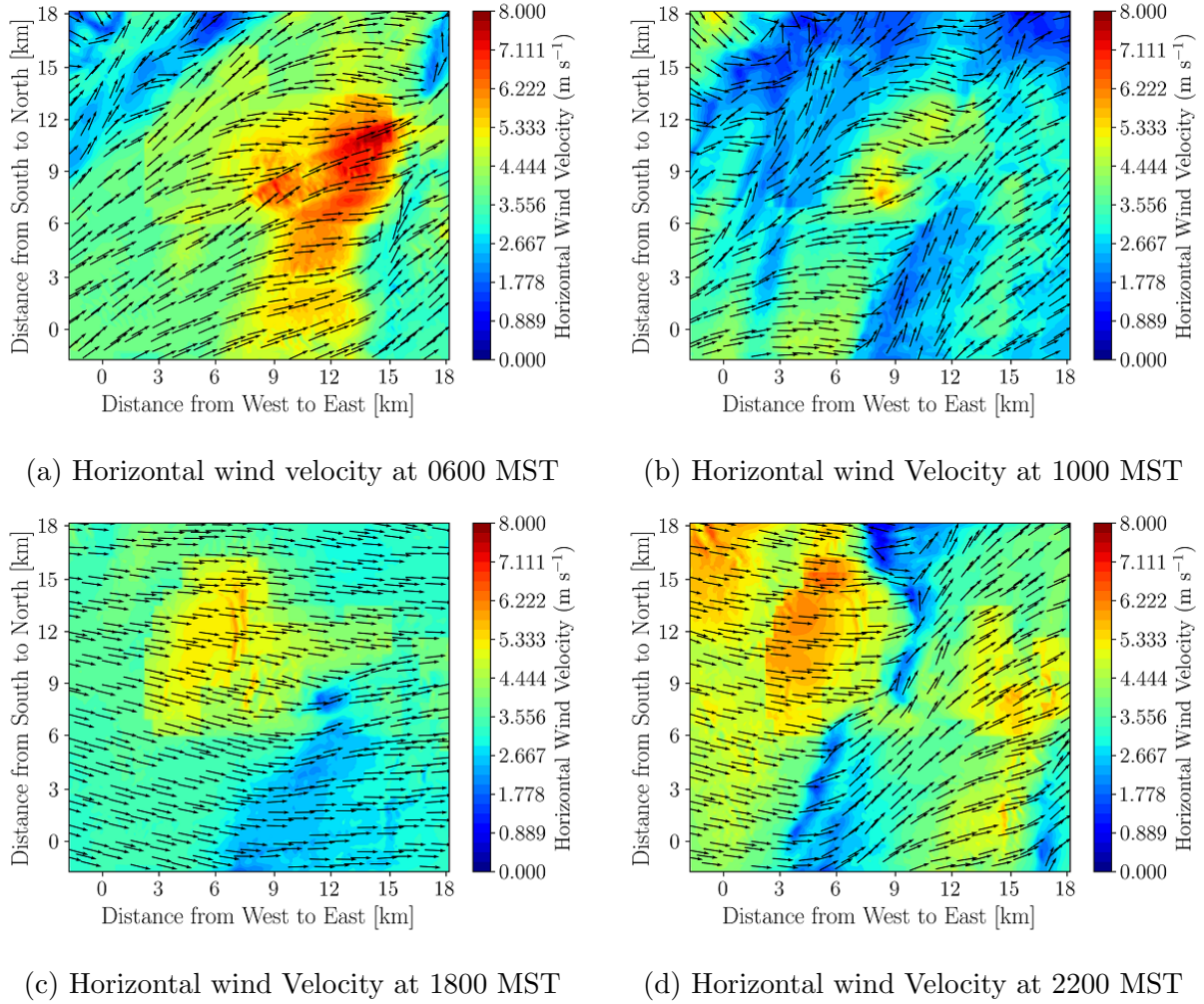
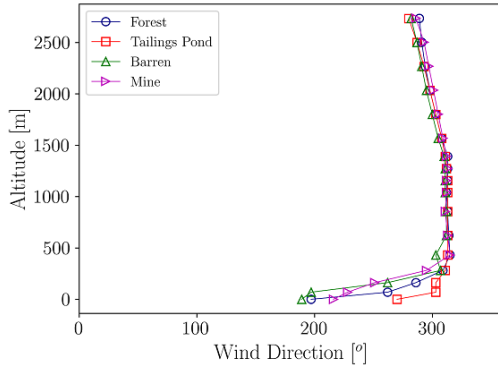
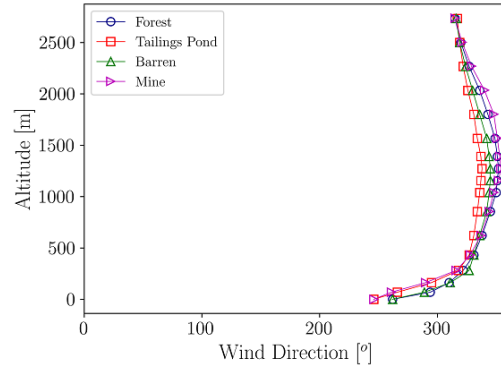


Figure C.1: Horizontal wind velocity and wind direction at 10 m on March 16, 2019.

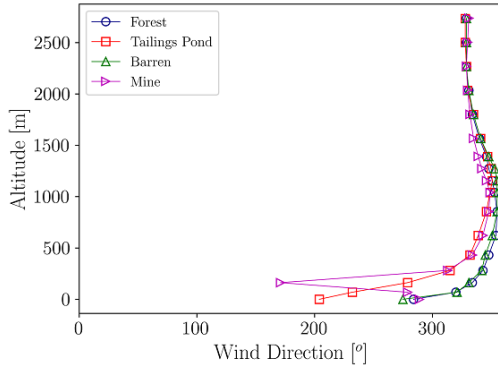
## C.2 Winter: Wind Direction



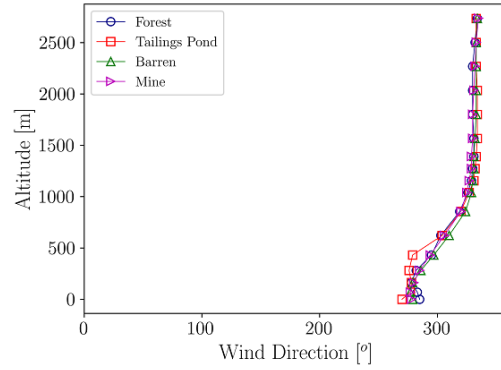
(a) Wind direction at 0200 MST



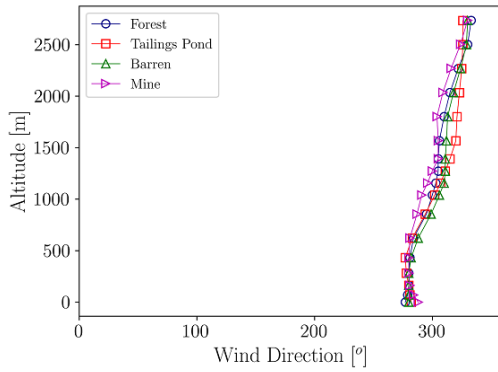
(b) Wind direction at 0600 MST



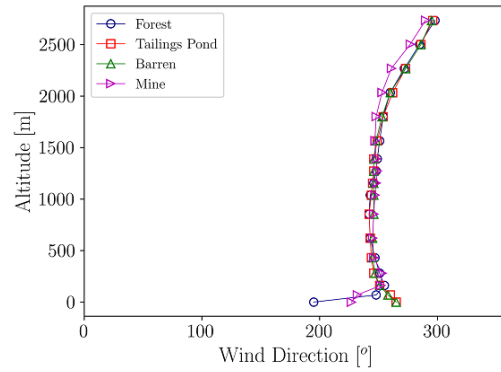
(c) Wind direction at 1000 MST



(d) Wind direction at 1400 MST



(e) Wind direction at 1800 MST



(f) Wind direction at 2200 MST

Figure C.2: Wind direction profiles based on pressure height at different locations on March 16, 2019.

### C.3 Winter: Surface Skin Temperature

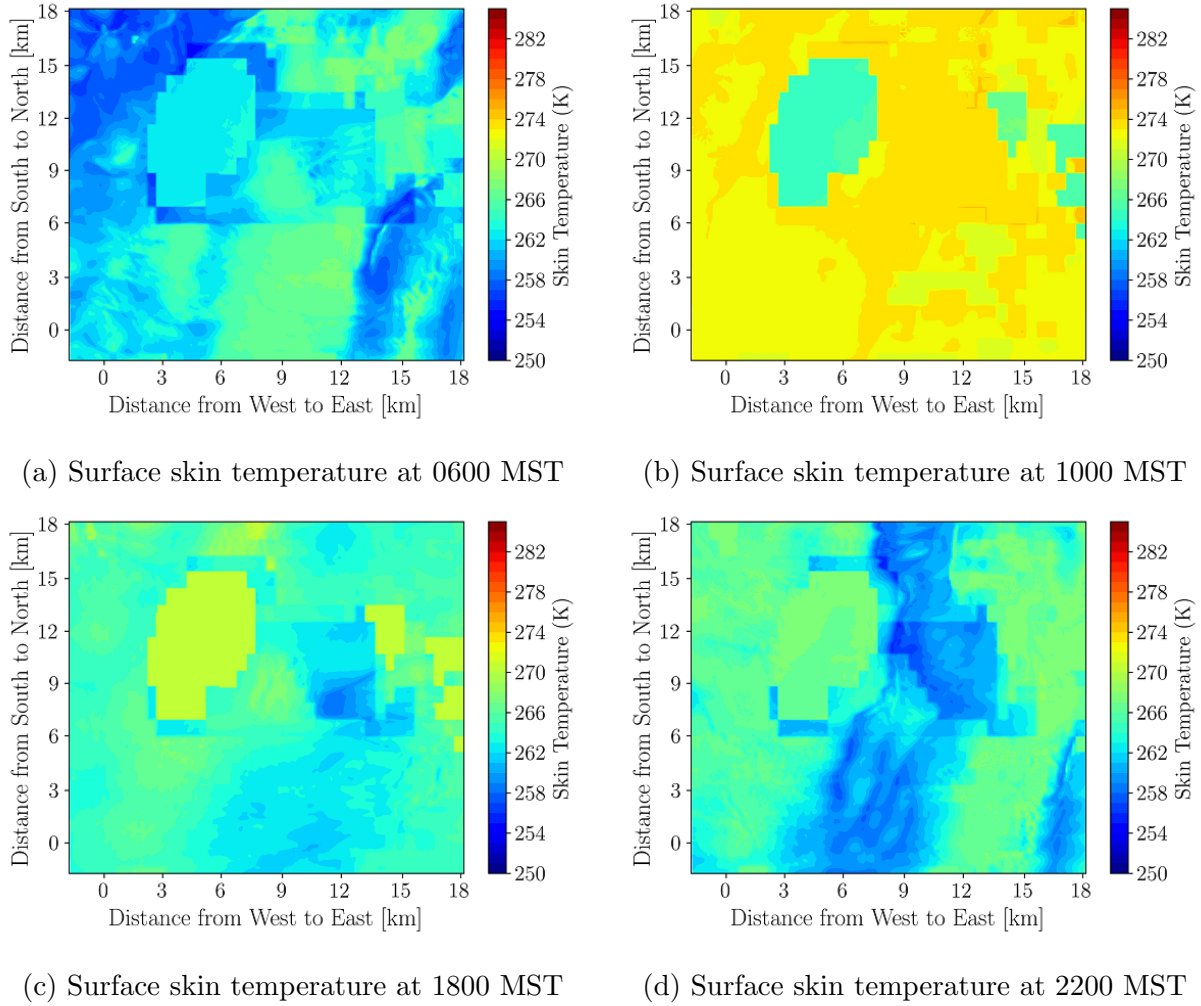


Figure C.3: Surface skin temperature on March 16, 2019.

## C.4 Winter: Potential Temperature at 2 m

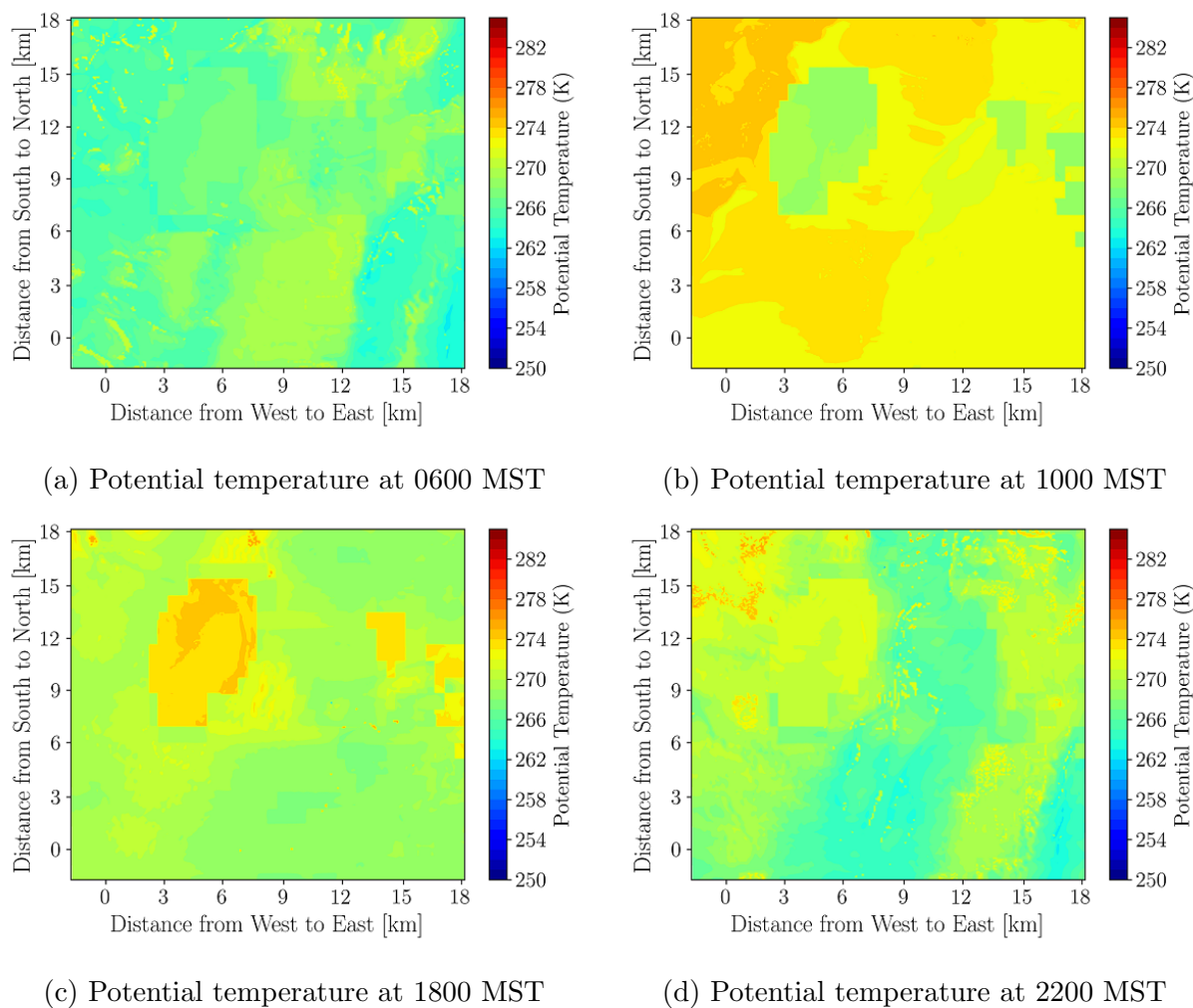


Figure C.4: Potential temperature at 2 mon March 16, 2019.

## C.5 Winter: Relative Humidity at 2 m

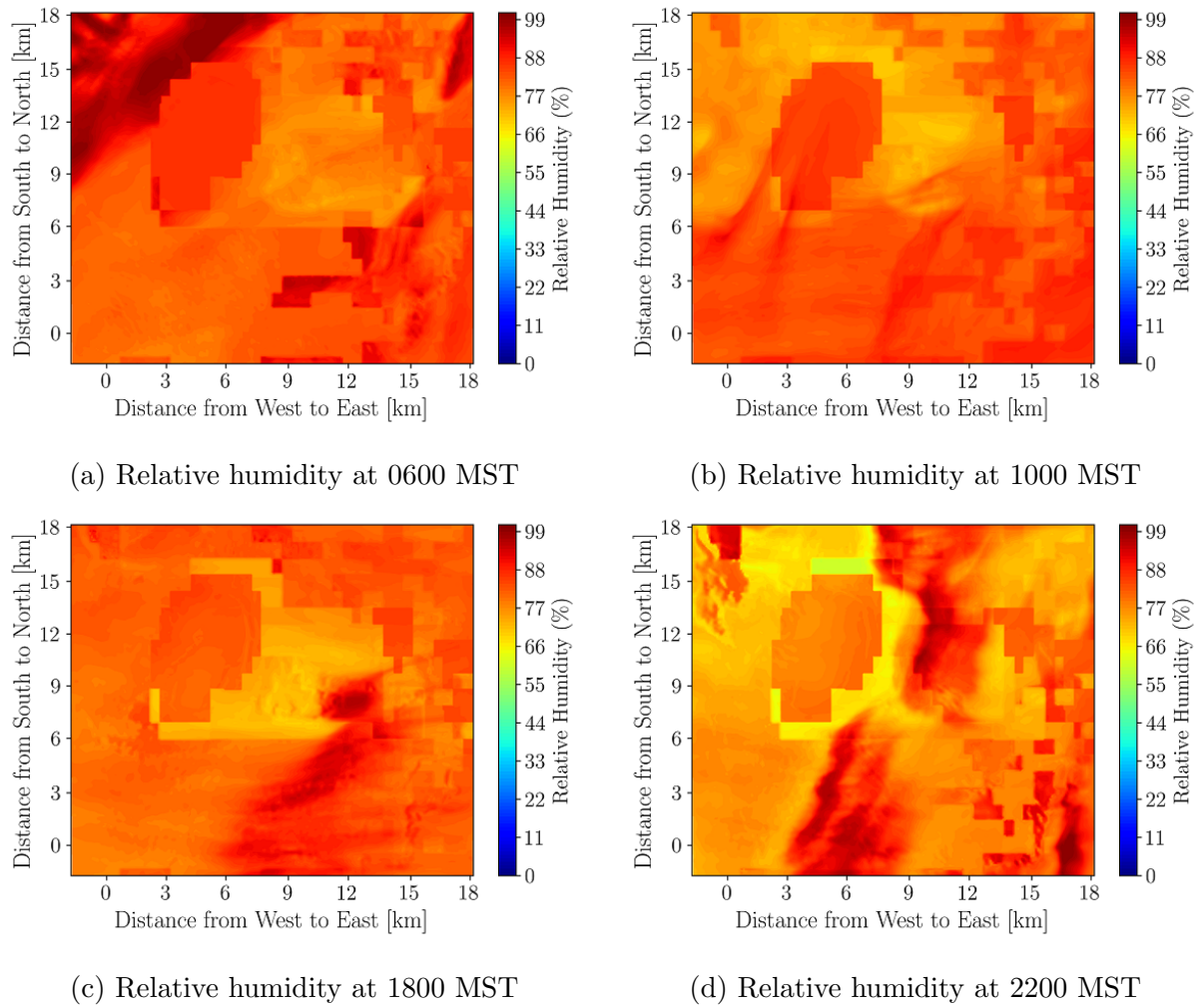


Figure C.5: Relative Humidity at 2 m on March 16, 2019.

## C.6 Winter: Sensible Vertical Turbulent Heat Flux at 10 m

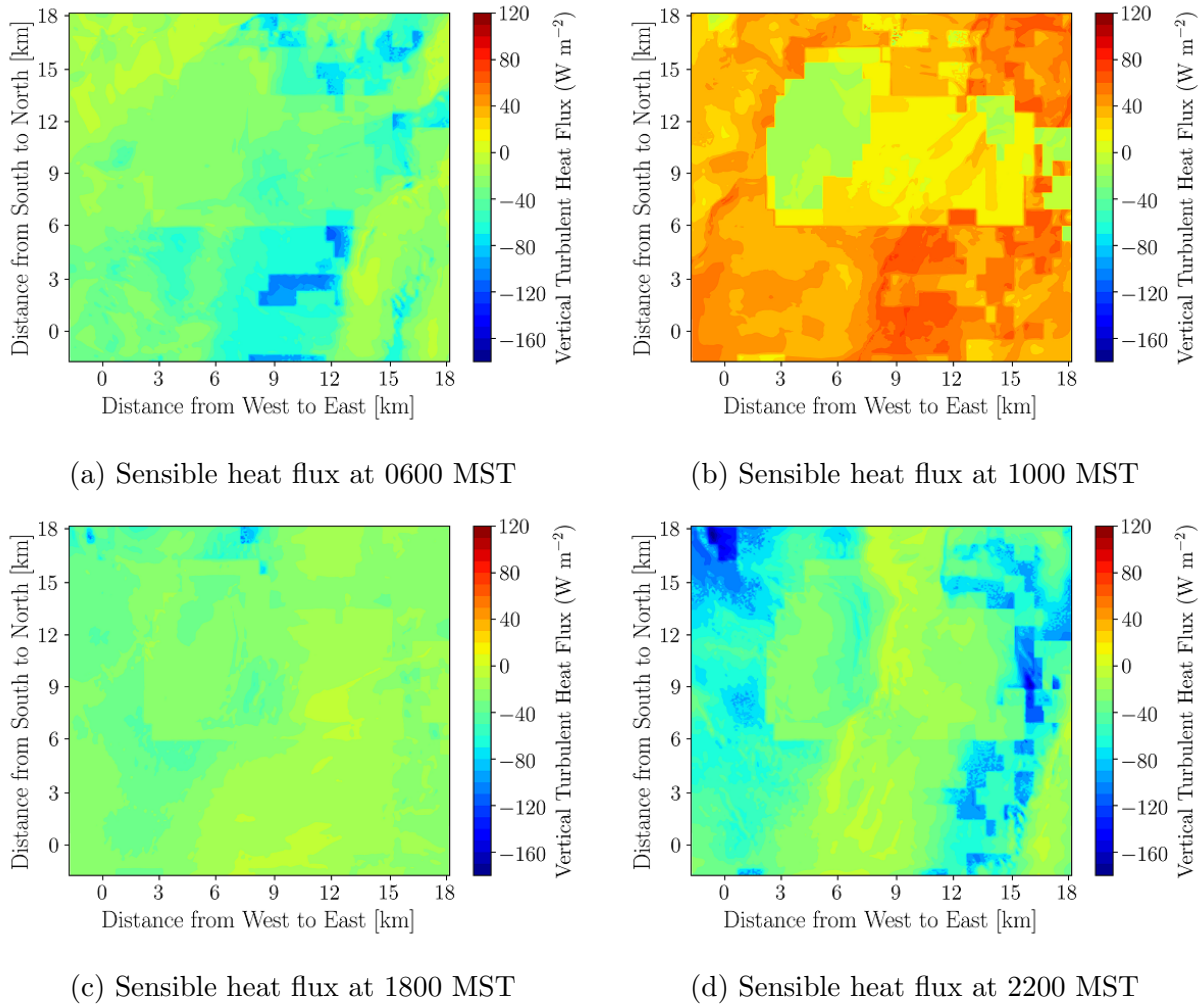


Figure C.6: 10-m sensible vertical turbulent heat flux on March 16, 2019.

# Appendix D

## Source Code

### D.1 Code to Extract Columns of Meteorological Solution Variables from WRF

#### D.1.1 Surface Level Extractor

```
import netCDF4
import matplotlib.pyplot as plt
import datetime as dt
import numpy as np

Outputfile="Mine_98/wrfout_2019-03-16_UltrasonicAnemometer1.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_UltrasonicAnemometer2.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_MineTrailer.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_RWDISonicAnemometer1.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_RWDISonicAnemometer2.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_RWDISonicAnemometer3.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_WBEACupAnemometer.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_TANAB_Surface.txt"
#Outputfile="Mine_98/wrfout_2019-03-16_TANAB_Surface.txt"

# Load the NetCDF files for each saved time
wrfout0='Mine_98/wrfprd/wrfout_d05_2019-03-16_07.00.00'
wrfout1='Mine_98/wrfprd/wrfout_d05_2019-03-16_08.00.00'
wrfout2='Mine_98/wrfprd/wrfout_d05_2019-03-16_09.00.00'
wrfout3='Mine_98/wrfprd/wrfout_d05_2019-03-16_10.00.00'
wrfout4='Mine_98/wrfprd/wrfout_d05_2019-03-16_11.00.00'
wrfout5='Mine_98/wrfprd/wrfout_d05_2019-03-16_12.00.00'
wrfout6='Mine_98/wrfprd/wrfout_d05_2019-03-16_13.00.00'
wrfout7='Mine_98/wrfprd/wrfout_d05_2019-03-16_14.00.00'
wrfout8='Mine_98/wrfprd/wrfout_d05_2019-03-16_15.00.00'
wrfout9='Mine_98/wrfprd/wrfout_d05_2019-03-16_16.00.00'
```



```

wrfout10='Mine_98/wrfprd/wrfout_d05.2019-03-16_17.00.00 '
wrfout11='Mine_98/wrfprd/wrfout_d05.2019-03-16_18.00.00 '
wrfout12='Mine_98/wrfprd/wrfout_d05.2019-03-16_19.00.00 '
wrfout13='Mine_98/wrfprd/wrfout_d05.2019-03-16_20.00.00 '
wrfout14='Mine_98/wrfprd/wrfout_d05.2019-03-16_21.00.00 '
wrfout15='Mine_98/wrfprd/wrfout_d05.2019-03-16_22.00.00 '
wrfout16='Mine_98/wrfprd/wrfout_d05.2019-03-16_23.00.00 '
wrfout17='Mine_98/wrfprd/wrfout_d05.2019-03-17_00.00.00 '
wrfout18='Mine_98/wrfprd/wrfout_d05.2019-03-17_01.00.00 '
wrfout19='Mine_98/wrfprd/wrfout_d05.2019-03-17_02.00.00 '
wrfout20='Mine_98/wrfprd/wrfout_d05.2019-03-17_03.00.00 '
wrfout21='Mine_98/wrfprd/wrfout_d05.2019-03-17_04.00.00 '
wrfout22='Mine_98/wrfprd/wrfout_d05.2019-03-17_05.00.00 '
wrfout23='Mine_98/wrfprd/wrfout_d05.2019-03-17_06.00.00 '

nTimes=24

#Define a position tolerance that is acceptable when colocating the WRF and experimental
    instrument [degrees]
PosTol=0.0005          #For High horizontal resolution
#PosTol=0.001          #For Medium horizontal resolution
#PosTol=0.002          #For low horizontal resolution (coarse grid)

#Number of grid elements in the horizontal directions X and Y
nHorizontalGrid=231    #For High horizontal resolution
#nHorizontalGrid=153    #For Medium horizontal resolution
#nHorizontalGrid=84     #For low horizontal resolution (coarse grid)

#UofA Ultrasonic 1 Location for March 2019 campaign (N Mine)
instrument_lat = 57.3526
instrument_lon = -111.8438

#UofA Ultrasonic 2 Location for March 2019 campaign (W pond)
#instrument_lat = 57.3452
#instrument_lon = -111.9411

#UofA Ultrasonic 2 Location for May 2018 campaign (SW pond) moved out of water
#instrument_lat = 57.3250
#instrument_lon = -111.9445

#SAIT Mine Trailer Location for May 2018 campaign
#instrument_lat = 57.3421
#instrument_lon = -111.8490

#RWDI Ultrasonic 1 Location for May 2018 campaign (SW Mine)
#instrument_lat = 57.3090
#instrument_lon = -111.8521

#RWDI Ultrasonic 2 Location for May 2018 campaign (E Pond)
#instrument_lat = 57.3468
#instrument_lon = -111.8705

```



```

#RWDI Ultrasonic 3 Location for May 2018 campaign (NW Pond)
#instrument_lat = 57.3684
#instrument_lon = -111.9552

#AIR Monitoring Station Location for May 2018 campaign (Wood Buffalo Environmental
    Association)
#instrument_lat = 57.3037
#instrument_lon = -111.7395


nc0 = netCDF4.Dataset(wrfout0)
nc0.variables.keys()
#Print the list of available variables in the net CDF file , do not repeat this command
print(nc0.variables.keys())

nc1 = netCDF4.Dataset(wrfout1)
nc1.variables.keys()

nc2 = netCDF4.Dataset(wrfout2)
nc2.variables.keys()

nc3 = netCDF4.Dataset(wrfout3)
nc3.variables.keys()

nc4 = netCDF4.Dataset(wrfout4)
nc4.variables.keys()

nc5 = netCDF4.Dataset(wrfout5)
nc5.variables.keys()

nc6 = netCDF4.Dataset(wrfout6)
nc6.variables.keys()

nc7 = netCDF4.Dataset(wrfout7)
nc7.variables.keys()

nc8 = netCDF4.Dataset(wrfout8)
nc8.variables.keys()

nc9 = netCDF4.Dataset(wrfout9)
nc9.variables.keys()

nc10 = netCDF4.Dataset(wrfout10)
nc10.variables.keys()

nc11 = netCDF4.Dataset(wrfout11)
nc11.variables.keys()

nc12 = netCDF4.Dataset(wrfout12)
nc12.variables.keys()

nc13 = netCDF4.Dataset(wrfout13)

```

```

nc13.variables.keys()

nc14 = netCDF4.Dataset(wrfout14)
nc14.variables.keys()

nc15 = netCDF4.Dataset(wrfout15)
nc15.variables.keys()

nc16 = netCDF4.Dataset(wrfout16)
nc16.variables.keys()

nc17 = netCDF4.Dataset(wrfout17)
nc17.variables.keys()

nc18 = netCDF4.Dataset(wrfout18)
nc18.variables.keys()

nc19 = netCDF4.Dataset(wrfout19)
nc19.variables.keys()

nc20 = netCDF4.Dataset(wrfout20)
nc20.variables.keys()

nc21 = netCDF4.Dataset(wrfout21)
nc21.variables.keys()

nc22 = netCDF4.Dataset(wrfout22)
nc22.variables.keys()

nc23 = netCDF4.Dataset(wrfout23)
nc23.variables.keys()

#lat and long of observation location (SODAR) they are the same for all hours, do not
    redefine
lat = nc0.variables['XLAT'][:]
lon = nc0.variables['XLONG'][:]

#Find the model lat and long closest to the observation location and their index
indx_lat = []
indx_lon = []
for i in range(0,len(lat[0][:])):
    indx_lat.append([x for x in range(len(lat[0][i,:])) if abs(lat[0][i,x]-instrument_lat) <
        PosTol])
    indx_lon.append([x for x in range(len(lon[0][i,:])) if abs(lon[0][i,x]-instrument_lon) <
        PosTol])

row = []
col = []
for i in range(0,nHorizontalGrid):
    if indx_lat[i] != []:
        row.append(i)
        col.append(indx_lat[i])

```

```

row_lon = []
col_lon = []
for i in range(0,nHorizontalGrid):
    if indx_lon[i] != []:
        row_lon.append(i)
        col_lon.append(indx_lon[i])

#Define wind speed at 10 m for nTimes
SMean_10=np.zeros((nTimes,1))

#Define U component of wind speed at 10 m for nTimes
UMean_10=np.zeros((nTimes,1))

#Define V component of wind speed at 10 m for nTimes
VMean_10=np.zeros((nTimes,1))

#Define wind direction at 10 m for nTimes
dMean_10=np.zeros((nTimes,1))

#Define temperature at 2 m for nTimes
TMean_2=np.zeros((nTimes,1))

#Define potential temperature at 2 m for nTimes
PT_2=np.zeros((nTimes,1))

#Define relative humidity at 2 m for nTimes
RHMean_2=np.zeros((nTimes,1))

#Define surface sensible heat flux for nTimes
HF_s=np.zeros((nTimes,1))

#Define surface latent heat flux for nTimes
LH_s=np.zeros((nTimes,1))

#Define bulk Richardson number for nTimes
RB=np.zeros((nTimes,1))

#Define planetary boundary-layer height number for nTimes
PBLH=np.zeros((nTimes,1))

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):
    if j==0:
        nc=nc0
    if j==1:
        nc=nc1
    if j == 2:
        nc = nc2
    if j == 3:
        nc = nc3
    if j == 4:

```

```

        nc = nc4
    if j == 5:
        nc = nc5
    if j == 6:
        nc = nc6
    if j == 7:
        nc = nc7
    if j == 8:
        nc = nc8
    if j == 9:
        nc = nc9
    if j == 10:
        nc = nc10
    if j == 11:
        nc = nc11
    if j == 12:
        nc = nc12
    if j == 13:
        nc = nc13
    if j == 14:
        nc = nc14
    if j == 15:
        nc = nc15
    if j == 16:
        nc = nc16
    if j == 17:
        nc = nc17
    if j == 18:
        nc = nc18
    if j == 19:
        nc = nc19
    if j == 20:
        nc = nc20
    if j == 21:
        nc = nc21
    if j == 22:
        nc = nc22
    if j == 23:
        nc = nc23
    if j == 24:
        nc = nc24

SMean_10[j] = nc.variables['S10_MEAN'][0,row[0],col_lon[0]]
UMean_10[j] = nc.variables['U10_MEAN'][0,row[0],col_lon[0]]
VMean_10[j] = nc.variables['V10_MEAN'][0,row[0],col_lon[0]]
TMean_2[j] = nc.variables['T02_MEAN'][0,row[0],col_lon[0]]
PT_2[j] = nc.variables['TH2'][0,row[0],col_lon[0]]
RHMean_2[j] = nc.variables['RH02_MEAN'][0,row[0],col_lon[0]]
HF_s[j] = nc.variables['HFX'][0,row[0],col_lon[0]]
LH_s[j] = nc.variables['LH'][0,row[0],col_lon[0]]
RB[j] = nc.variables['BR'][0,row[0],col_lon[0]]
PBLH[j] = nc.variables['PBLH'][0,row[0],col_lon[0]]

```

```

    if UMean_10[j] >= 0:
        dMean_10[j] = 270 - (np.arctan(VMean_10[j] / UMean_10[j]) * 57.3)
    else:
        dMean_10[j] = 90 - (np.arctan(VMean_10[j] / UMean_10[j]) * 57.3)

outputFile = open(Outputfile, "w")
outputFile.write("#Times: Local Standard Time Mountain Standard Time (MST) \n")
outputFile.write("#0:time(hr) \t #1:SMean_10(m_s^-1) \t #2:UMean_10(m_s^-1) \t #3:
    VMean_10(m_s^-1) \t #4:dMean_10(deg) \t #5:TMean_2(C) \t #6:PT_2(C) \t #7:RHMean_2(
    (%)) \t #8:HF_s(W_m^-2) \t #9:LH_s(W_m^-2) \t #10:RB \t #11:PBLH(m) \t \n")

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):
    outputFile.write("%0f\t%.3f\t%.3f\t%.3f\t%.0f\t%.3f\t%.3f\t%.3f\t%.3f\t
        \t%.3f\t%.3f\t%.3f\t\n" \
        % (j, SMean_10[j], UMean_10[j], VMean_10[j], dMean_10[j], \
        TMean_2[j], PT_2[j], RHMean_2[j]*100, HF_s[j], LH_s[j], RB[j],
        PBLH[j]))

```

## D.1.2 High Level Extractor for SODAR and TANAB

```

import netCDF4
import matplotlib.pyplot as plt
import datetime as dt
import numpy as np

#Outputfile="Mine_97/wrfout_2018-05-30.SODAR.txt"
Outputfile="Mine_97/wrfout_2018-05-30.TANAB.HigherLevel.txt"

# Load the NetCDF files for each saved time
wrfout0='Mine_97/wrfprd/wrfout_d05.2018-05-30_07.00.00'
wrfout1='Mine_97/wrfprd/wrfout_d05.2018-05-30_08.00.00'
wrfout2='Mine_97/wrfprd/wrfout_d05.2018-05-30_09.00.00'
wrfout3='Mine_97/wrfprd/wrfout_d05.2018-05-30_10.00.00'
wrfout4='Mine_97/wrfprd/wrfout_d05.2018-05-30_11.00.00'
wrfout5='Mine_97/wrfprd/wrfout_d05.2018-05-30_12.00.00'
wrfout6='Mine_97/wrfprd/wrfout_d05.2018-05-30_13.00.00'
wrfout7='Mine_97/wrfprd/wrfout_d05.2018-05-30_14.00.00'
wrfout8='Mine_97/wrfprd/wrfout_d05.2018-05-30_15.00.00'
wrfout9='Mine_97/wrfprd/wrfout_d05.2018-05-30_16.00.00'
wrfout10='Mine_97/wrfprd/wrfout_d05.2018-05-30_17.00.00'
wrfout11='Mine_97/wrfprd/wrfout_d05.2018-05-30_18.00.00'
wrfout12='Mine_97/wrfprd/wrfout_d05.2018-05-30_19.00.00'
wrfout13='Mine_97/wrfprd/wrfout_d05.2018-05-30_20.00.00'
wrfout14='Mine_97/wrfprd/wrfout_d05.2018-05-30_21.00.00'
wrfout15='Mine_97/wrfprd/wrfout_d05.2018-05-30_22.00.00'
wrfout16='Mine_97/wrfprd/wrfout_d05.2018-05-30_23.00.00'
wrfout17='Mine_97/wrfprd/wrfout_d05.2018-05-31_00.00.00'
wrfout18='Mine_97/wrfprd/wrfout_d05.2018-05-31_01.00.00'
wrfout19='Mine_97/wrfprd/wrfout_d05.2018-05-31_02.00.00'

```

```

wrfout20='Mine_97/wrfprd/wrfout_d05.2018-05-31_03.00.00'
wrfout21='Mine_97/wrfprd/wrfout_d05.2018-05-31_04.00.00'
wrfout22='Mine_97/wrfprd/wrfout_d05.2018-05-31_05.00.00'
wrfout23='Mine_97/wrfprd/wrfout_d05.2018-05-31_06.00.00'

#nlevels is to match the elevations of SODAR
#nLevels=13           #For low vertical resolution (45 Levels)
nLevels=17            #For mid vertical resolution (90 Levels)
#nLevels=20           #For high vertical resolution (120 Levels)
nTimes=24

#Number of grid elements in the horizontal directions X and Y
#nHorizontalGrid=231   #For High horizontal resolution
nHorizontalGrid=153    #For Medium horizontal resolution
#nHorizontalGrid=84    #For low horizontal resolution (coarse grid)

#Define a position tolerance that is acceptable when colocating the WRF and experimental
    instrument [degrees]
#PosTol=0.0005         #For High horizontal resolution
PosTol=0.001           #For Medium horizontal resolution
#PosTol=0.002          #For low horizontal resolution (coarse grid)

#SODAR Location for May 2018 campaign
#instrument_lat = 57.3394
#instrument_lon = -111.9455

#UofG TANAB Location for May 2018 campaign (May 18 and 24)
#instrument_lat = 57.3237
#instrument_lon = -111.8197

#UofG TANAB Location for May 2018 campaign (May 30)
instrument_lat = 57.3509
instrument_lon = -111.8694

nc0 = netCDF4.Dataset(wrfout0)
nc0.variables.keys()

#Print the list of available variables in the net CDF file, do not repeat this command
print (nc0.variables.keys())

nc1 = netCDF4.Dataset(wrfout1)
nc1.variables.keys()

nc2 = netCDF4.Dataset(wrfout2)
nc2.variables.keys()

nc3 = netCDF4.Dataset(wrfout3)
nc3.variables.keys()

nc4 = netCDF4.Dataset(wrfout4)
nc4.variables.keys()

```

```

nc5 = netCDF4.Dataset(wrfout5)
nc5.variables.keys()

nc6 = netCDF4.Dataset(wrfout6)
nc6.variables.keys()

nc7 = netCDF4.Dataset(wrfout7)
nc7.variables.keys()

nc8 = netCDF4.Dataset(wrfout8)
nc8.variables.keys()

nc9 = netCDF4.Dataset(wrfout9)
nc9.variables.keys()

nc10 = netCDF4.Dataset(wrfout10)
nc10.variables.keys()

nc11 = netCDF4.Dataset(wrfout11)
nc11.variables.keys()

nc12 = netCDF4.Dataset(wrfout12)
nc12.variables.keys()

nc13 = netCDF4.Dataset(wrfout13)
nc13.variables.keys()

nc14 = netCDF4.Dataset(wrfout14)
nc14.variables.keys()

nc15 = netCDF4.Dataset(wrfout15)
nc15.variables.keys()

nc16 = netCDF4.Dataset(wrfout16)
nc16.variables.keys()

nc17 = netCDF4.Dataset(wrfout17)
nc17.variables.keys()

nc18 = netCDF4.Dataset(wrfout18)
nc18.variables.keys()

nc19 = netCDF4.Dataset(wrfout19)
nc19.variables.keys()

nc20 = netCDF4.Dataset(wrfout20)
nc20.variables.keys()

nc21 = netCDF4.Dataset(wrfout21)
nc21.variables.keys()

nc22 = netCDF4.Dataset(wrfout22)

```

```

nc22.variables.keys()

nc23 = netCDF4.Dataset(wrfout23)
nc23.variables.keys()

#lat and long of observation location (SODAR) they are the same for all hours, do not
    redefine
lat = nc0.variables['XLAT'][:]
lon = nc0.variables['XLONG'][:]

#Find the model lat and long closest to the observation location and their index
indx_lat = []
indx_lon = []
for i in range(0,len(lat[0][0,:])):
    indx_lat.append([x for x in range(len(lat[0][i,:])) if abs(lat[0][i,x]-instrument_lat) <
        PosTol])
    indx_lon.append([x for x in range(len(lon[0][i,:])) if abs(lon[0][i,x]-instrument_lon) <
        PosTol])

row = []
col = []
for i in range(0,nHorizontalGrid):
    if indx_lat[i] != []:
        row.append(i)
        col.append(indx_lat[i])

row_lon = []
col_lon = []
for i in range(0,nHorizontalGrid):
    if indx_lon[i] != []:
        row_lon.append(i)
        col_lon.append(indx_lon[i])

#Define pressure for nLevels (rows) and nTimes (column), i.e. Pr=zeros((rows,columns))
Pr=np.zeros((nLevels, nTimes))
#Define altitude for nLevels (rows) and nTimes (column)
Z=np.zeros((nLevels, nTimes))
#Define U component of velocity for nLevels (rows) and nTimes (column)
U=np.zeros((nLevels, nTimes))
#Define V component of velocity for nLevels (rows) and nTimes (column)
V=np.zeros((nLevels, nTimes))
#Define S component of velocity for nLevels (rows) and nTimes (column)
S=np.zeros((nLevels, nTimes))
#Define W component of velocity for nLevels (rows) and nTimes (column)
W=np.zeros((nLevels, nTimes))
#Define perturbation potential temperature for nLevels (rows) and nTimes (column)
#This is deviation from some average or reference potential temperature
PPT=np.zeros((nLevels, nTimes))
#Define d or wind direction for nLevels (rows) and nTimes (column)
d=np.zeros((nLevels, nTimes))

#Elevation loop i: from 0 to nLevels

```



```

for i in range(0,nLevels):
    #Time loop j: from 0 to nTimes
    for j in range(0, nTimes):
        if j==0:
            nc=nc0
        if j==1:
            nc=nc1
        if j == 2:
            nc = nc2
        if j == 3:
            nc = nc3
        if j == 4:
            nc = nc4
        if j == 5:
            nc = nc5
        if j == 6:
            nc = nc6
        if j == 7:
            nc = nc7
        if j == 8:
            nc = nc8
        if j == 9:
            nc = nc9
        if j == 10:
            nc = nc10
        if j == 11:
            nc = nc11
        if j == 12:
            nc = nc12
        if j == 13:
            nc = nc13
        if j == 14:
            nc = nc14
        if j == 15:
            nc = nc15
        if j == 16:
            nc = nc16
        if j == 17:
            nc = nc17
        if j == 18:
            nc = nc18
        if j == 19:
            nc = nc19
        if j == 20:
            nc = nc20
        if j == 21:
            nc = nc21
        if j == 22:
            nc = nc22
        if j == 23:
            nc = nc23

```

```

#Calculate pressure and altitude for the current level and time
Pr[i][j]= nc.variables['PHYD'][0,i,row[0],col_lon[0]]
Z[i][j] = 0.3048 * (1 - (Pr[i][j] / Pr[0][j]) ** 0.190284) * 145366.45
U[i][j] = nc.variables['U'][0, i, row[0], col_lon[0]]
V[i][j] = nc.variables['V'][0, i, row[0], col_lon[0]]
W[i][j] = nc.variables['W'][0, i, row[0], col_lon[0]]
S[i][j] = np.sqrt((U[i][j] ** 2) + (V[i][j] ** 2))
#PT[i][j] = (nc.variables['T'][0, i, row[0], col_lon[0]]+273.15)*(Pr[0][j] / Pr[i][j]
    ])**0.286
PPT[i][j] = (nc.variables['T'][0, i, row[0], col_lon[0]])

if U[i][j] >= 0:
    d[i][j] = 270 - (np.arctan(V[i][j] / U[i][j]) * 57.3)
else:
    d[i][j] = 90 - (np.arctan(V[i][j] / U[i][j]) * 57.3)

#Economized way to write results in text files given nLevels and nTimes
outputFile = open(Outputfile, "w")
outputFile.write ("##Times:_Local_Standard_Time=_Mountain_Standard_Time_(MST)_\n")
outputFile.write ("#0:Time_(hr)_\t_#1:Altitude_(m)_\t_#2:U_(m_s^-1)_\t_#3:V_(m_s^-1)_\t_#4:S
    _\t_#5:W_(m_s^-1)_\t_#6:_Wind_Direction_(degree)_\t_#7.Pr_(Pa)_\t_#8.Pert_PT_(K
    )_\t_\n")

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):
#Elevation loop i: from 0 to nLevels
    for i in range(0,nLevels):

        outputFile.write("%0f_\t_%0f_\t_%3f_\t_%3f_\t_%3f_\t_%3f_\t_%0f_\t_%3f_\t_
            %3f_\t_\n" \
                % (j, Z[i][j], U[i][j], V[i][j], S[i][j], W[i][j], d[i][j], Pr[i][j]
                    ], PPT[i][j]))

```

### D.1.3 Surface Level Extractor for Contour Plots

```

# Code to extract surface temperatures from WRF at a single hour for a single domain

import netCDF4
import matplotlib.pyplot as plt
import datetime as dt
import numpy as np

# Load the NetCDF files for each saved time
wrfout='Mine_98/wrfprd/wrfout_d05_2019-03-16_09_00_00'
#wrfout='Mine_98/wrfprd/wrfout_d05_2019-03-16_13_00_00'
#wrfout='Mine_98/wrfprd/wrfout_d05_2019-03-16_17_00_00'
#wrfout='Mine_98/wrfprd/wrfout_d05_2019-03-16_21_00_00'
#wrfout='Mine_98/wrfprd/wrfout_d05_2019-03-17_01_00_00'
#wrfout='Mine_98/wrfprd/wrfout_d05_2019-03-17_05_00_00'

# Designate output file name and location

```



## D.1.4 High Level Extractor for Profile Plots

```
#Extract profiles of potential temperature and wind speed for profile analysis

import netCDF4
import matplotlib.pyplot as plt
import datetime as dt
import numpy as np

Outputfile="Mine_68/wrfout_2018-05-18_ForestProfile.txt"
#Outputfile="Mine_68/wrfout_2018-05-18_PondProfile.txt"
#Outputfile="Mine_68/wrfout_2018-05-18_BarrenProfile.txt"
#Outputfile="Mine_68/wrfout_2018-05-18_MineProfile.txt"

# Load the NetCDF files for each saved time
wrfout2='Mine_68/wrfprd/wrfout_d05_2018-05-18_09_00_00'
wrfout6='Mine_68/wrfprd/wrfout_d05_2018-05-18_13_00_00'
wrfout10='Mine_68/wrfprd/wrfout_d05_2018-05-18_17_00_00'
wrfout14='Mine_68/wrfprd/wrfout_d05_2018-05-18_21_00_00'
wrfout18='Mine_68/wrfprd/wrfout_d05_2018-05-19_01_00_00'
wrfout22='Mine_68/wrfprd/wrfout_d05_2018-05-19_05_00_00'

#nlevels is to match the elevations of SODAR
#nLevels=13           #For medium vertical resolution (45 Levels)
nLevels=17           #For high vertical resolution (90 Levels)
nTimes=6

#Number of grid elements in the horizontal directions X and Y
nHorizontalGrid=153   #For Medium horizontal resolution
#nHorizontalGrid=84   #For low horizontal resolution (coarse grid)

#Define a position tolerance that is acceptable when colocating the WRF and experimental
instrument [degrees]
PosTol=0.001         #For Medium horizontal resolution
#PosTol=0.002         #For low horizontal resolution (coarse grid)

#Forest Location for May 2018 campaign
profile_lat = 57.3678
profile_lon = -111.8311

#Pond Location for May 2018 campaign
#profile_lat = 57.3408
#profile_lon = -111.9085

#Barren Location for May 2018 campaign
#profile_lat = 57.3419
#profile_lon = -111.8600

#Mine Location for May 2018 campaign
#profile_lat = 57.3265
#profile_lon = -111.8234
```

```

nc0 = netCDF4.Dataset(wrfout2)
nc0.variables.keys()

#Print the list of available variables in the net CDF file , do not repeat this command
print (nc0.variables.keys())

nc1 = netCDF4.Dataset(wrfout6)
nc1.variables.keys()

nc2 = netCDF4.Dataset(wrfout10)
nc2.variables.keys()

nc3 = netCDF4.Dataset(wrfout14)
nc3.variables.keys()

nc4 = netCDF4.Dataset(wrfout18)
nc4.variables.keys()

nc5 = netCDF4.Dataset(wrfout22)
nc5.variables.keys()

#lat and long of observation location (SODAR) they are the same for all hours, do not
    redefine
lat = nc0.variables['XLAT'][:]
lon = nc0.variables['XLONG'][:]

#Find the model lat and long closest to the location specified above and their index
indx_lat = []
indx_lon = []
for i in range(0,len(lat[0][0,:])):
    indx_lat.append([x for x in range(len(lat[0][i,:])) if abs(lat[0][i,x]-profile_lat) <
        PosTol])
    indx_lon.append([x for x in range(len(lon[0][i,:])) if abs(lon[0][i,x]-profile_lon) <
        PosTol])

row = []
col = []
for i in range(0,nHorizontalGrid):
    if indx_lat[i] != []:
        row.append(i)
        col.append(indx_lat[i])

row_lon = []
col_lon = []
for i in range(0,nHorizontalGrid):
    if indx_lon[i] != []:
        row_lon.append(i)
        col_lon.append(indx_lon[i])

#Define pressure for nLevels (rows) and nTimes (column), i.e. Pr=zeros((rows,columns))
Pr=np.zeros((nLevels, nTimes))
#Define altitude for nLevels (rows) and nTimes (column)

```

```

Z=np.zeros((nLevels, nTimes))
#Define U component of velocity for nLevels (rows) and nTimes (column)
U=np.zeros((nLevels, nTimes))
#Define V component of velocity for nLevels (rows) and nTimes (column)
V=np.zeros((nLevels, nTimes))
#Define S component of velocity for nLevels (rows) and nTimes (column)
S=np.zeros((nLevels, nTimes))
#Define W component of velocity for nLevels (rows) and nTimes (column)
W=np.zeros((nLevels, nTimes))
#Define perturbation potential temperature for nLevels (rows) and nTimes (column)
#This is deviation from some average or reference potential temperature
PPT=np.zeros((nLevels, nTimes))
#Define potential temperature at 2m altitude
PT_2=np.zeros((nTimes,1))
#Define potential temperature profile
PT=np.zeros((nLevels, nTimes))
#Define d or wind direction for nLevels (rows) and nTimes (column)
d=np.zeros((nLevels, nTimes))

#Elevation loop i: from 0 to nLevels
for i in range(0,nLevels):
    #Time loop j: from 0 to nTimes
    for j in range(0, nTimes):
        if j == 0:
            nc = nc0
        if j == 1:
            nc = nc1
        if j == 2:
            nc = nc2
        if j == 3:
            nc = nc3
        if j == 4:
            nc = nc4
        if j == 5:
            nc = nc5

        #Calculate pressure and altitude for the current level and time
        Pr[i][j]= nc.variables['PHYD'][0,i,row[1],col_lon[0]]
        Z[i][j] = 0.3048 * (1 - (Pr[i][j] / Pr[0][j]) ** 0.190284) * 145366.45
        U[i][j] = nc.variables['U'][0, i, row[1], col_lon[0]]
        V[i][j] = nc.variables['V'][0, i, row[1], col_lon[0]]
        W[i][j] = nc.variables['W'][0, i, row[1], col_lon[0]]
        S[i][j] = np.sqrt((U[i][j] ** 2) + (V[i][j] ** 2))
        PT_2[j] = nc.variables['TH2'][0, row[1], col_lon[0]]
        PPT[i][j] = (nc.variables['T'][0, i, row[1], col_lon[0]])
        #Add potential temperature at 2m to perturbation potential temperature for total
        potential temperature
        PT[i][j] = PT_2[j]+PPT[i][j]

        if U[i][j] >= 0:
            d[i][j] = 270 - (np.arctan(V[i][j] / U[i][j]) * 57.3)
        else:

```

```

d[i][j] = 90 - (np.arctan(V[i][j] / U[i][j]) * 57.3)

#Economized way to write results in text files given nLevels and nTimes
outputFile = open(Outputfile, "w")
outputFile.write ("#Times:_Local_Standard_Time=_Mountain_Standard_Time_(MST)_\n")
outputFile.write ("#0:Time_(hr)_\t_#1:Altitude_(m)_\t_#2:U_(m_s^-1)_\t_#3:V_(m_s^-1)_\t_#4:S
\t_#5:W_(m_s^-1)_\t_#6:Wind_Direction_(degree)_\t_#7:Pr_(Pa)_\t_#8:Pert_PT_(K)
\t_#9:PT_(K)_\n")

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):
#Elevation loop i: from 0 to nLevels
    for i in range(0, nLevels):

        outputFile.write ("%0f_\t_%0f_\t_%3f_\t_%3f_\t_%3f_\t_%3f_\t_%0f_\t_%3f_\t_
        %3f_\t_%3f_\n" \
            % ((24/nTimes)*j+24/nTimes/2, Z[i][j], U[i][j], V[i][j], S[i][j], W
            [i][j], d[i][j], Pr[i][j], PPT[i][j], PT[i][j]))

```

## D.2 Code to Perform Statistical Analysis Between Observations and WRF

### D.2.1 WRF Compare to SODAR for Percentile Calculations

```

import matplotlib.pyplot as plt
import numpy as np

#Define magnetic declination for northern Alberta
#If you rerun this code add this to the experimental wind direction
#Magnetic North is to the east of true North
phiMagDec=14

#Define output filename associated with WRF/SODAR comparison
Outputfile="StatisticalAnalysis/WRFCompareToSODAR_95_96_97.txt"

#Define filenames associated with SODAR measurements on May 18, 24, 30
InputSODARfile1="Instruments/SODAR20180518.txt"
InputSODARfile2="Instruments/SODAR20180524.txt"
InputSODARfile3="Instruments/SODAR20180530.txt"

#Define filenames associated with WRF runs of interest on May 18, 24, 30

InputWRFfile1="Mine_95/wrfout_2018-05-18.SODAR.txt"
InputWRFfile2="Mine_96/wrfout_2018-05-24.SODAR.txt"
InputWRFfile3="Mine_97/wrfout_2018-05-30.SODAR.txt"

#Input calibration equation file
InputCalibration="StatisticalAnalysis/WRFSODARCalibration.txt"

```

```

#Define number of altitudes and time windows for comparison
#Note: it is possible to consider fewer levels for comparison given the uncertainty of SODAR
      data at high altitude
#nLevels = 4 works very well
#We are choosing 12 levels for all vertical grid resolutions
nLevels = 4
nTimes = 6

#Must ensure there is exactly the same number of observations and model in each altitude
      range
#For 45 and 90 levels: use this
SODARLevel=[100, 200, 300, 400, 600, 900, 1100, 1400, 1600, 1900, 2300, 2700]
WRFLowLevel=[55, 150, 250, 400, 600, 840, 1100, 1350, 1550, 1760, 2250, 2700]
WRFLowLevel=[75, 180, 295, 450, 650, 900, 1190, 1430, 1650, 1940, 2350, 2880]
TimeWindowLow=[0, 4, 8, 12, 16, 20]
TimeWindowHigh=[3, 7, 11, 15, 19, 23]

#Read the SODAR data files
InputSODARdata1 = np.loadtxt(InputSODARfile1, usecols=[0,1,2,3,4])
timeSODAR1=InputSODARdata1[:,0]
alSODAR1=InputSODARdata1[:,1]
sSODAR1=InputSODARdata1[:,2]
wSODAR1=InputSODARdata1[:,3]
#If rerun the SODAR comparison, add the magnetic declination angle here
dSODAR1=InputSODARdata1[:,4]

InputSODARdata2 = np.loadtxt(InputSODARfile2, usecols=[0,1,2,3,4])
timeSODAR2=InputSODARdata2[:,0]
alSODAR2=InputSODARdata2[:,1]
sSODAR2=InputSODARdata2[:,2]
wSODAR2=InputSODARdata2[:,3]
#If rerun the SODAR comparison, add the magnetic declination angle here
dSODAR2=InputSODARdata2[:,4]

InputSODARdata3 = np.loadtxt(InputSODARfile3, usecols=[0,1,2,3,4])
timeSODAR3=InputSODARdata3[:,0]
alSODAR3=InputSODARdata3[:,1]
sSODAR3=InputSODARdata3[:,2]
wSODAR3=InputSODARdata3[:,3]
#If rerun the SODAR comparison, add the magnetic declination angle here
dSODAR3=InputSODARdata3[:,4]

#Read the WRF data files
InputWRFdata1 = np.loadtxt(InputWRFfile1, usecols=[0,1,4,5,6])
timeWRF1=InputWRFdata1[:,0]
alWRF1=InputWRFdata1[:,1]
sWRF1=InputWRFdata1[:,2]
wWRF1=InputWRFdata1[:,3]
dWRF1=InputWRFdata1[:,4]

InputWRFdata2 = np.loadtxt(InputWRFfile2, usecols=[0,1,4,5,6])
timeWRF2=InputWRFdata2[:,0]

```



```

alWRF2=InputWRFdata2[:,1]
sWRF2=InputWRFdata2[:,2]
wWRF2=InputWRFdata2[:,3]
dWRF2=InputWRFdata2[:,4]

InputWRFdata3 = np.loadtxt(InputWRFfile3, usecols=[0,1,4,5,6])
timeWRF3=InputWRFdata3[:,0]
alWRF3=InputWRFdata3[:,1]
sWRF3=InputWRFdata3[:,2]
wWRF3=InputWRFdata3[:,3]
dWRF3=InputWRFdata3[:,4]

InputCalibrationdata=np.loadtxt(InputCalibration, usecols=[2,3])
a=InputCalibrationdata[:,0]
b=InputCalibrationdata[:,1]

#####

#we need 2D matrices with nLevel and nTime (6) for the following variables
#S SODAR, S STD SODAR, S WRF, S STD WRF, FB S SODAR-WRF, NMSE S SODAR-WRF
#W SODAR, W STD SODAR, W WRF, W STD WRF, FB W SODAR-WRF, NMSE W SODAR-WRF
#delta phi SODAR-WRF, median or average of delta phi SODAR-WRF, if average STD delta phi
  SODAR-WRF

S50SODAR=np.zeros((nTimes,nLevels))
S25SODAR=np.zeros((nTimes,nLevels))
S75SODAR=np.zeros((nTimes,nLevels))
S50WRF=np.zeros((nTimes,nLevels))
S25WRF=np.zeros((nTimes,nLevels))
S75WRF=np.zeros((nTimes,nLevels))
SFBSDARWRF=np.zeros((nTimes,nLevels))
SNMSESODARWRF=np.zeros((nTimes,nLevels))
SBSODARWRF=np.zeros((nTimes,nLevels))

W50SODAR=np.zeros((nTimes,nLevels))
W25SODAR=np.zeros((nTimes,nLevels))
W75SODAR=np.zeros((nTimes,nLevels))
W50WRF=np.zeros((nTimes,nLevels))
W25WRF=np.zeros((nTimes,nLevels))
W75WRF=np.zeros((nTimes,nLevels))
WFBSDARWRF=np.zeros((nTimes,nLevels))
WNMSESODARWRF=np.zeros((nTimes,nLevels))
WBSODARWRF=np.zeros((nTimes,nLevels))

DSODAR=np.zeros((nTimes,nLevels))
DWRF=np.zeros((nTimes,nLevels))
deltaPhi50SODARWRF=np.zeros((nTimes,nLevels))
deltaPhi25SODARWRF=np.zeros((nTimes,nLevels))
deltaPhi75SODARWRF=np.zeros((nTimes,nLevels))

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):

```

```

#Elevation loop i: from 0 to nLevels
for i in range(0,nLevels):

    #Extract index for the current level and time from SODAR data for all three
    days
    IndexSODAR1=[x for x in range(0,len(alSODAR1)) if (alSODAR1[x]==SODARLevel[i]
        and TimeWindowLow[j]<=timeSODAR1[x]<=TimeWindowHigh[j])]
    IndexSODAR2=[x for x in range(0,len(alSODAR2)) if (alSODAR2[x]==SODARLevel[i]
        and TimeWindowLow[j]<=timeSODAR2[x]<=TimeWindowHigh[j])]
    IndexSODAR3=[x for x in range(0,len(alSODAR3)) if (alSODAR3[x]==SODARLevel[i]
        and TimeWindowLow[j]<=timeSODAR3[x]<=TimeWindowHigh[j])]

    IndexWRF1 = [x for x in range(0, len(alWRF1)) if (WRFLowLevel[i] <= alWRF1[x]
        <= WRFHighLevel[i] and TimeWindowLow[j] <= timeWRF1[x] <= TimeWindowHigh[j]
        )]
    IndexWRF2 = [x for x in range(0, len(alWRF2)) if (WRFLowLevel[i] <= alWRF2[x]
        <= WRFHighLevel[i] and TimeWindowLow[j] <= timeWRF2[x] <= TimeWindowHigh[j]
        )]
    IndexWRF3 = [x for x in range(0, len(alWRF3)) if (WRFLowLevel[i] <= alWRF3[x]
        <= WRFHighLevel[i] and TimeWindowLow[j] <= timeWRF3[x] <= TimeWindowHigh[j]
        )]

    #Calculation for Horizontal Wind Velocity

    ArraySSODAR=np.concatenate((a[i]*sSODAR1[IndexSODAR1]+b[i],a[i]*sSODAR2[
        IndexSODAR2]+b[i],a[i]*sSODAR3[IndexSODAR3]+b[i]),axis=None)

    S50SODAR[j][i] = np.nanmedian(ArraySSODAR)
    S25SODAR[j][i] = np.nanpercentile(ArraySSODAR, 25)
    S75SODAR[j][i] = np.nanpercentile(ArraySSODAR, 75)

    ArraySWRF=np.concatenate((sWRF1[IndexWRF1],sWRF2[IndexWRF2],sWRF3[IndexWRF3]),
        axis=None)

    S50WRF[j][i] = np.nanmedian(ArraySWRF)
    S25WRF[j][i] = np.nanpercentile(ArraySWRF, 25)
    S75WRF[j][i] = np.nanpercentile(ArraySWRF, 75)

    SFBSSODARWRF[j][i] = (np.nanmean(ArraySWRF)-np.nanmean(ArraySSODAR))/(0.5*(np.
        nanmean(ArraySWRF)+np.nanmean(ArraySSODAR)))
    SNMSESODARWRF[j][i] = np.absolute((np.nanmean(ArraySWRF-ArraySSODAR)**2)/((np.
        nanmean(ArraySWRF)*np.nanmean(ArraySSODAR))))
    SBSODARWRF[j][i] = np.nanmean(ArraySWRF) - np.nanmean(ArraySSODAR)

    #Calculation for Vertical Wind Velocity

    ArrayWSODAR = np.concatenate((wSODAR1[IndexSODAR1], wSODAR2[IndexSODAR2],
        wSODAR3[IndexSODAR3]),axis=None)

    W50SODAR[j][i] = np.nanmedian(ArrayWSODAR)
    W25SODAR[j][i] = np.nanpercentile(ArrayWSODAR, 25)
    W75SODAR[j][i] = np.nanpercentile(ArrayWSODAR, 75)

```

```

ArrayWWRF = np.concatenate((wWRF1[IndexWRF1], wWRF2[IndexWRF2], wWRF3[
    IndexWRF3]), axis=None)
W50WRF[j][i] = np.nanmedian(ArrayWWRF)
W25WRF[j][i] = np.nanpercentile(ArrayWWRF, 25)
W75WRF[j][i] = np.nanpercentile(ArrayWWRF, 75)

WFBSDARWRF[j][i] = (np.nanmean(ArrayWWRF) - np.nanmean(ArrayWSODAR)) / (0.5 *
    (np.nanmean(ArrayWWRF) + np.nanmean(ArrayWSODAR)))
WNMSESODARWRF[j][i] = np.absolute((np.nanmean(ArrayWWRF - ArrayWSODAR) ** 2) /
    ((np.nanmean(ArrayWWRF) * np.nanmean(ArrayWSODAR))))
WBSODARWRF[j][i] = np.nanmean(ArrayWWRF) - np.nanmean(ArrayWSODAR)

# Calculation of bias for Wind Direction
ArrayDSODAR = np.concatenate((dSODAR1[IndexSODAR1], dSODAR2[IndexSODAR2],
    dSODAR3[IndexSODAR3]), axis=None)
ArrayDWRF = np.concatenate((dWRF1[IndexWRF1], dWRF2[IndexWRF2], dWRF3[
    IndexWRF3]), axis=None)

# Correct for magnetic inclination
ArrayWD10mSurfaceLevelCorrected = ArrayDSODAR + phiMagDec

for x in range(0, len(ArrayWD10mSurfaceLevelCorrected)):
    if ArrayWD10mSurfaceLevelCorrected[x] >= 360:
        ArrayWD10mSurfaceLevelCorrected[x] = ArrayWD10mSurfaceLevelCorrected[x]
            - 360

DiffPhiSODARWRF = ArrayDWRF - ArrayWD10mSurfaceLevelCorrected

for x in range(0, len(DiffPhiSODARWRF)):
    if DiffPhiSODARWRF[x] <= -180:
        DiffPhiSODARWRF[x] = DiffPhiSODARWRF[x] + 360
    if DiffPhiSODARWRF[x] >= 180:
        DiffPhiSODARWRF[x] = DiffPhiSODARWRF[x] - 360

#Make sure this is computed inside the i and j loop, not inside the x loop
deltaPhi50SODARWRF[j][i] = np.nanmedian(DiffPhiSODARWRF)
deltaPhi25SODARWRF[j][i] = np.nanpercentile(DiffPhiSODARWRF, 25)
deltaPhi75SODARWRF[j][i] = np.nanpercentile(DiffPhiSODARWRF, 75)

# Calculate the entire fractional bias and normalized mean square error of S for all the
    data of horizontal wind speed

print('Bias, Fractional Bias (FB) and Normalized Mean Square Error (NMSE) for all the
    medians of horizontal wind speed')
print('SFB=', (np.nanmean(S50WRF) - np.nanmean(S50SODAR)) / (0.5 * (np.nanmean(S50WRF) + np.
    nanmean(S50SODAR))))
print('SNMSE=', np.absolute((np.nanmean(S50WRF - S50SODAR) ** 2) / ((np.nanmean(S50WRF) * np
    .nanmean(S50SODAR)))))
print('SB=', np.nanmean(S50WRF) - np.nanmean(S50SODAR))

```

```
# Calculate the entire fractional bias and normalized mean square error for all the data of
vertical wind speed

print( 'Bias, _Fractional_Bias_(FB) _and _Normalized _Mean _Square _Error_(NMSE) _for _all _the _
medians_of _vertical _wind _speed' )
print( 'WFB=', (np.nanmean(W50WRF) - np.nanmean(W50SODAR)) / (0.5 * (np.nanmean(W50WRF) + np.
nanmean(W50SODAR))) )
print( 'WNMSE=', np.absolute((np.nanmean(W50WRF - W50SODAR) ** 2) / ((np.nanmean(W50WRF) * np.
nanmean(W50SODAR)))) )
print( 'WB=', np.nanmean(W50WRF) - np.nanmean(W50SODAR) )

# Calculate the entire bias for all the data of Wind Direction

print( 'Median_of_Bias_(B) _for _all _the _differences _of _wind _direction _(Degree)' )
print( 'DB50=', (np.nanmean(deltaPhi50SODARWRF)) )

#Economized way to write results in text files given nLevels and nTimes
outputFile = open( Outputfile , "w" )
outputFile.write( "#Times:_Local_Standard_Time_=_Mountain_Standard_Time_(MST)_\n" )
outputFile.write( "#0:TimeWindowLow_(hr)_\t_1:TimeWindowHigh_(hr)_\t_2:SodarLevel_(hr)_\t_3:
S50SODAR_(m_s^-1)_\t_4:S25SODAR_(m_s^-1)_\t_5:S75SODAR_(m_s^-1)_\t_6:S50WRF_(m_s^-1)_\t_7:
S25WRF_(m_s^-1)_\t_8:S75WRF_(m_s^-1)_\t_#9:SFB50SODARWRF_\t_#10:SNMSE50SODARWRF_\t_
" 11:W50SODAR_(m_s^-1)_\t_12:W25SODAR_(m_s^-1)_\t_13:W75SODAR_(m_s^-1)_\t_
14:W50WRF_(m_s^-1)_\t_15:W25WRF_(m_s^-1)_\t_16:W75WRF_(m_s^-1)_\t_#17:
WFB50SODARWRF_\t_#18:WNMSE50SODARWRF_\t_#19:deltaPhi50SODARWRF_(^o)_\t_
#20:deltaPhi25SODARWRF_(^o)_\t_#21:deltaPhi75SODARWRF_(^o)_\t_
" 22:SBSODARWRF_(m_s^-1)_\t_#23:WBSODARWRF_(m_s^-1)_\n" )

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):
#Elevation loop i: from 0 to nLevels
for i in range(0, nLevels):

outputFile.write("%0f_\t_%0f_\t_%0f_\t_%3f_\t_%3f_\t_%3f_\t_%3f_\t_%3f_\t_
%3f_\t_%8f_\t_%8f_\t_%3f_\t_%3f_\t_%3f_\t_%3f_\t_%3f_\t_%8f_\t_
%8f_\t_%3f_\t_%3f_\t_%3f_\t_%5f_\t_%5f_\n" \
% (TimeWindowLow[j], TimeWindowHigh[j], SODARLevel[i], S50SODAR[j][i]
, S25SODAR[j][i], S75SODAR[j][i], S50WRF[j][i], S25WRF[j][i],
S75WRF[j][i], SFB50SODARWRF[j][i], SNMSE50SODARWRF[j][i],
W50SODAR[j][i], W25SODAR[j][i], W75SODAR[j][i], W50WRF[j][i],
W25WRF[j][i], W75WRF[j][i], WFB50SODARWRF[j][i], WNMSE50SODARWRF
[j][i], deltaPhi50SODARWRF[j][i], deltaPhi25SODARWRF[j][i],
deltaPhi75SODARWRF[j][i],
SBSODARWRF[j][i], WBSODARWRF[j][i]))
```

### D.2.2 WRF Compare to SODAR for Error Calculations

```
import numpy as np
```

```
#Define magnetic declaration for northern Alberta
```

```

#If you rerun this code add this to the experimental wind direction
#Magnetic North is to the east of true North
phiMagDec=14

#Define output filename associated with WRF/SODAR comparison
#Outputfile="StatisticalAnalysis/WRFCompareToSODAR_95_96_97.txt"

#Define filenames associated with SODAR measurements on May 18, 24, 30
InputSODARfile1="Instruments/SODAR20180518.txt"
InputSODARfile2="Instruments/SODAR20180524.txt"
InputSODARfile3="Instruments/SODAR20180530.txt"

#Define filenames associated with WRF runs of interest on May 18, 24, 30

InputWRFfile1="Mine_95/wrfout_2018-05-18.SODAR.txt"
InputWRFfile2="Mine_96/wrfout_2018-05-24.SODAR.txt"
InputWRFfile3="Mine_97/wrfout_2018-05-30.SODAR.txt"

#Input calibration equation file
InputCalibration="StatisticalAnalysis/WRFSODARCalibration.txt"

#Define number of altitudes and time windows for comparison
#Note: it is possible to consider fewer levels for comparison given the uncertainty of SODAR
data at high altitude
#nLevels = 4 works very well
#We are choosing 12 levels for all vertical grid resolutions
nLevels = 4
nTimes = 6

#Must ensure there is exactly the same number of observations and model in each altitude
range
#For 45 and 90 levels: use this
SODARLevel=[100, 200, 300, 400, 600, 900, 1100, 1400, 1600, 1900, 2300, 2700]
WRFLowLevel=[55, 150, 250, 400, 600, 840, 1100, 1350, 1550, 1760, 2250, 2700]
WRFHighLevel=[75, 180, 295, 450, 650, 900, 1190, 1430, 1650, 1940, 2350, 2880]
TimeWindowLow=[0, 4, 8, 12, 16, 20]
TimeWindowHigh=[3, 7, 11, 15, 19, 23]

#Read the SODAR data files
InputSODARdata1 = np.loadtxt(InputSODARfile1, usecols=[0,1,2,3,4])
timeSODAR1=InputSODARdata1[:,0]
alSODAR1=InputSODARdata1[:,1]
sSODAR1=InputSODARdata1[:,2]
wSODAR1=InputSODARdata1[:,3]
#If rerun the SODAR comparison, add the magnetic declination angle here
dSODAR1=InputSODARdata1[:,4]

InputSODARdata2 = np.loadtxt(InputSODARfile2, usecols=[0,1,2,3,4])
timeSODAR2=InputSODARdata2[:,0]
alSODAR2=InputSODARdata2[:,1]
sSODAR2=InputSODARdata2[:,2]
wSODAR2=InputSODARdata2[:,3]

```

```

#If rerun the SODAR comparison, add the magnetic declination angle here
dSODAR2=InputSODARdata2[:,4]

InputSODARdata3 = np.loadtxt(InputSODARfile3, usecols=[0,1,2,3,4])
timeSODAR3=InputSODARdata3[:,0]
alSODAR3=InputSODARdata3[:,1]
sSODAR3=InputSODARdata3[:,2]
wSODAR3=InputSODARdata3[:,3]
#If rerun the SODAR comparison, add the magnetic declination angle here
dSODAR3=InputSODARdata3[:,4]

#Read the WRF data files
InputWRFdata1 = np.loadtxt(InputWRFfile1, usecols=[0,1,4,5,6])
timeWRF1=InputWRFdata1[:,0]
alWRF1=InputWRFdata1[:,1]
sWRF1=InputWRFdata1[:,2]
wWRF1=InputWRFdata1[:,3]
dWRF1=InputWRFdata1[:,4]

InputWRFdata2 = np.loadtxt(InputWRFfile2, usecols=[0,1,4,5,6])
timeWRF2=InputWRFdata2[:,0]
alWRF2=InputWRFdata2[:,1]
sWRF2=InputWRFdata2[:,2]
wWRF2=InputWRFdata2[:,3]
dWRF2=InputWRFdata2[:,4]

InputWRFdata3 = np.loadtxt(InputWRFfile3, usecols=[0,1,4,5,6])
timeWRF3=InputWRFdata3[:,0]
alWRF3=InputWRFdata3[:,1]
sWRF3=InputWRFdata3[:,2]
wWRF3=InputWRFdata3[:,3]
dWRF3=InputWRFdata3[:,4]

InputCalibrationdata=np.loadtxt(InputCalibration, usecols=[2,3])
a=InputCalibrationdata[:,0]
b=InputCalibrationdata[:,1]

#####

#we need 2D matrices with nLevel and nTime (6) for the following variables
#S SODAR, S STD SODAR, S WRF, S STD WRF, FB S SODAR-WRF, NMSE S SODAR-WRF
#W SODAR, W STD SODAR, W WRF, W STD WRF, FB W SODAR-WRF, NMSE W SODAR-WRF

SSODARLevel0=np.nan
SWRFLevel0=np.nan
WDSODARLevel0=np.nan
WDWRFLevel0=np.nan

SSODARLevel1=np.nan
SWRFLevel1=np.nan
WDSODARLevel1=np.nan

```

```

WDWRFLevel1=np . nan

SSODARLevel2=np . nan
SWRFLevel2=np . nan
WDSODARLevel2=np . nan
WDWRFLevel2=np . nan

SSODARLevel3=np . nan
SWRFLevel3=np . nan
WDSODARLevel3=np . nan
WDWRFLevel3=np . nan

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):
    #Elevation loop i: from 0 to nLevels
    for i in range(0,nLevels):

        #Extract index for the current level and time from SODAR data for all three
        days
        IndexSODAR1=[x for x in range(0,len(alSODAR1)) if (alSODAR1[x]==SODARLevel[i]
            and TimeWindowLow[j]<=timeSODAR1[x]<=TimeWindowHigh[j])]
        IndexSODAR2=[x for x in range(0,len(alSODAR2)) if (alSODAR2[x]==SODARLevel[i]
            and TimeWindowLow[j]<=timeSODAR2[x]<=TimeWindowHigh[j])]
        IndexSODAR3=[x for x in range(0,len(alSODAR3)) if (alSODAR3[x]==SODARLevel[i]
            and TimeWindowLow[j]<=timeSODAR3[x]<=TimeWindowHigh[j])]

        IndexWRF1 = [x for x in range(0, len(alWRF1)) if (WRFLowLevel[i] <= alWRF1[x]
            <= WRFHighLevel[i] and TimeWindowLow[j] <= timeWRF1[x] <= TimeWindowHigh[j]
            )]]
        IndexWRF2 = [x for x in range(0, len(alWRF2)) if (WRFLowLevel[i] <= alWRF2[x]
            <= WRFHighLevel[i] and TimeWindowLow[j] <= timeWRF2[x] <= TimeWindowHigh[j]
            )]]
        IndexWRF3 = [x for x in range(0, len(alWRF3)) if (WRFLowLevel[i] <= alWRF3[x]
            <= WRFHighLevel[i] and TimeWindowLow[j] <= timeWRF3[x] <= TimeWindowHigh[j]
            )]]

        #Calculation for Horizontal Wind Velocity

        if i==0:
            SSODARLevel0=np . concatenate ((SSODARLevel0, a[i]*sSODAR1[IndexSODAR1]+b[i]
                ), a[i]*sSODAR2[IndexSODAR2]+b[i] , a[i]*sSODAR3[IndexSODAR3]+b[i] ) ,
                axis=None)
            SWRFLevel0=np . concatenate ((SWRFLevel0,sWRF1[IndexWRF1] ,sWRF2[IndexWRF2] ,
                sWRF3[IndexWRF3]) , axis=None)
            WDSODARLevel0=np . concatenate ((WDSODARLevel0,dSODAR1[IndexSODAR1] ,dSODAR2
                [IndexSODAR2] ,dSODAR3[IndexSODAR3]) , axis=None)
            WDWRFLevel0=np . concatenate ((WDWRFLevel0,dWRF1[IndexWRF1] ,dWRF2[IndexWRF2]
                ) ,dWRF3[IndexWRF3]) , axis=None)

        if i==1:
            SSODARLevel1=np . concatenate ((SSODARLevel1, a[i]*sSODAR1[IndexSODAR1]+b[i]
                ), a[i]*sSODAR2[IndexSODAR2]+b[i] , a[i]*sSODAR3[IndexSODAR3]+b[i] ) ,
                axis=None)

```

```

SWRFLevel1=np.concatenate((SWRFLevel1,sWRF1[IndexWRF1],sWRF2[IndexWRF2],
sWRF3[IndexWRF3]),axis=None)
WDSODARLevel1=np.concatenate((WDSODARLevel1,dSODAR1[IndexSODAR1],
dSODAR2[IndexSODAR2], dSODAR3[IndexSODAR3]),axis=None)
WDWRFLevel1 = np.concatenate((WDWRFLevel1,dWRF1[IndexWRF1], dWRF2[
IndexWRF2], dWRF3[IndexWRF3]),axis=None)
if i==2:
    SSODARLevel2=np.concatenate((SSODARLevel2,a[i]*sSODAR1[IndexSODAR1]+b[i]
],a[i]*sSODAR2[IndexSODAR2]+b[i],a[i]*sSODAR3[IndexSODAR3]+b[i]),
axis=None)
    SWRFLevel2=np.concatenate((SWRFLevel2,sWRF1[IndexWRF1],sWRF2[IndexWRF2],
sWRF3[IndexWRF3]),axis=None)
    WDSODARLevel2 = np.concatenate((WDSODARLevel2,dSODAR1[IndexSODAR1],
dSODAR2[IndexSODAR2], dSODAR3[IndexSODAR3]),axis=None)
    WDWRFLevel2 = np.concatenate((WDWRFLevel2,dWRF1[IndexWRF1], dWRF2[
IndexWRF2], dWRF3[IndexWRF3]),axis=None)
if i==3:
    SSODARLevel3=np.concatenate((SSODARLevel3,a[i]*sSODAR1[IndexSODAR1]+b[i]
],a[i]*sSODAR2[IndexSODAR2]+b[i],a[i]*sSODAR3[IndexSODAR3]+b[i]),
axis=None)
    SWRFLevel3=np.concatenate((SWRFLevel3,sWRF1[IndexWRF1],sWRF2[IndexWRF2],
sWRF3[IndexWRF3]),axis=None)
    WDSODARLevel3 = np.concatenate((WDSODARLevel3,dSODAR1[IndexSODAR1],
dSODAR2[IndexSODAR2], dSODAR3[IndexSODAR3]),axis=None)
    WDWRFLevel3 = np.concatenate((WDWRFLevel3,dWRF1[IndexWRF1], dWRF2[
IndexWRF2], dWRF3[IndexWRF3]),axis=None)

#Calculate Bias and RMSE
S10BiasLevel0 = np.nanmean(SWRFLevel0-SSODARLevel0)
S10RMSELevel0 = np.sqrt(np.nanmean((SWRFLevel0-SSODARLevel0)**2))

S10BiasLevel1 = np.nanmean(SWRFLevel1-SSODARLevel1)
S10RMSELevel1 = np.sqrt(np.nanmean((SWRFLevel1-SSODARLevel1)**2))

S10BiasLevel2 = np.nanmean(SWRFLevel2-SSODARLevel2)
S10RMSELevel2 = np.sqrt(np.nanmean((SWRFLevel2-SSODARLevel2)**2))

S10BiasLevel3 = np.nanmean(SWRFLevel3-SSODARLevel3)
S10RMSELevel3 = np.sqrt(np.nanmean((SWRFLevel3-SSODARLevel3)**2))

S10Bias=(S10BiasLevel0+S10BiasLevel1+S10BiasLevel2+S10BiasLevel3)/4
S10RMSE=(S10RMSELevel0+S10RMSELevel1+S10RMSELevel2+S10RMSELevel3)/4

#Correct for magnetic inclination
WDSODARLevel0 = WDSODARLevel0 + phiMagDec

for x in range(0, len(WDSODARLevel0)):
    if WDSODARLevel0[x] >= 360:
        WDSODARLevel0[x] = WDSODARLevel0[x] - 360

WDSODARLevel1 = WDSODARLevel1 + phiMagDec

```



```

for x in range(0, len(WDSODARLevel1)):
    if WDSODARLevel1[x] >= 360:
        WDSODARLevel1[x] = WDSODARLevel1[x] - 360

WDSODARLevel2 = WDSODARLevel2 + phiMagDec

for x in range(0, len(WDSODARLevel2)):
    if WDSODARLevel2[x] >= 360:
        WDSODARLevel2[x] = WDSODARLevel2[x] - 360

WDSODARLevel3 = WDSODARLevel3 + phiMagDec

for x in range(0, len(WDSODARLevel3)):
    if WDSODARLevel3[x] >= 360:
        WDSODARLevel3[x] = WDSODARLevel3[x] - 360

#Calculate difference in WD
DiffPhiLevel0 = WDWRFLLevel0 - WDSODARLevel0

for x in range(0, len(DiffPhiLevel0)):

    if DiffPhiLevel0[x] <= -180:
        DiffPhiLevel0[x] = DiffPhiLevel0[x] + 360
    if DiffPhiLevel0[x] >= 180:
        DiffPhiLevel0[x] = DiffPhiLevel0[x] - 360

#Calculate Bias and RMSE
WDBiasLevel0 = np.nanmean(DiffPhiLevel0)
WDRMSELevel0 = np.sqrt(np.nanmean((DiffPhiLevel0)**2))

DiffPhiLevel1 = WDWRFLLevel1 - WDSODARLevel1

for x in range(0, len(DiffPhiLevel1)):

    if DiffPhiLevel1[x] <= -180:
        DiffPhiLevel1[x] = DiffPhiLevel1[x] + 360
    if DiffPhiLevel1[x] >= 180:
        DiffPhiLevel1[x] = DiffPhiLevel1[x] - 360

#Calculate Bias and RMSE
WDBiasLevel1 = np.nanmean(DiffPhiLevel1)
WDRMSELevel1 = np.sqrt(np.nanmean((DiffPhiLevel1)**2))

DiffPhiLevel2 = WDWRFLLevel2 - WDSODARLevel2

for x in range(0, len(DiffPhiLevel2)):

    if DiffPhiLevel2[x] <= -180:
        DiffPhiLevel2[x] = DiffPhiLevel2[x] + 360
    if DiffPhiLevel2[x] >= 180:
        DiffPhiLevel2[x] = DiffPhiLevel2[x] - 360

```

```

#Calculate Bias and RMSE
WDBiasLevel2 = np.nanmean(DiffPhiLevel2)
WDRMSELevel2 = np.sqrt(np.nanmean((DiffPhiLevel2)**2))

DiffPhiLevel3 = WDWRFLevel3 - WDSODARLevel3

for x in range(0, len(DiffPhiLevel3)):

    if DiffPhiLevel3[x] <= -180:
        DiffPhiLevel3[x] = DiffPhiLevel3[x] + 360
    if DiffPhiLevel3[x] >= 180:
        DiffPhiLevel3[x] = DiffPhiLevel3[x] - 360

#Calculate Bias and RMSE
WDBiasLevel3 = np.nanmean(DiffPhiLevel3)
WDRMSELevel3 = np.sqrt(np.nanmean((DiffPhiLevel3)**2))

WDBias=(WDBiasLevel0+WDBiasLevel1+WDBiasLevel2+WDBiasLevel3)/4
WDRMSE=(WDRMSELevel0+WDRMSELevel1+WDRMSELevel2+WDRMSELevel3)/4

#Print the entire bias and root mean square error of S10m for all the data of horizontal
    wind speed at 10 m

print('Bias , and Root Mean Square Error (RMSE) for all the medians of Horizontal Wind Speed '
      )
print('S10mBias=', S10Bias)
print('S10mRMSE=', S10RMSE)

#Print the entire bias for all the data of Wind Direction at 10m

print('Bias (B) and RMSE for all the differences of wind direction (Degree)')
print('WD10m50Bias=', WDBias)
print('WD10m50RMSE=', WDRMSE)

```

## D.2.3 WRF Compare to TANAB for Percentile Calculations

```

import matplotlib.pyplot as plt
import numpy as np

#Define output filename associated with WRF/TANAB comparison
Outputfile="StatisticalAnalysis/WRFCompareToTANAB_95_96_97.txt"

#Define filenames associated with TANAB measurements on May 18, 24, 30
InputTANABfile1="Instruments/TANAB20180518.txt"
InputTANABfile2="Instruments/TANAB20180524.txt"
InputTANABfile3="Instruments/TANAB20180530.txt"

#Define filenames associated with WRF runs of interest on May 18, 24, 30

InputWRFfile1="Mine_95/wrfout_2018-05-18-TANAB-HigherLevel.txt"

```

```

InputWRFfile2=" Mine_96/ wrfout_2018-05-24_TANAB_HigherLevel.txt"
InputWRFfile3=" Mine_97/ wrfout_2018-05-30_TANAB_HigherLevel.txt"

#Define number of time windows (1 hour) for comparison
nTimes = 24

TimeWindow=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]

#Number of first few layers of WRF altitudes to check for matching Mean TANAB altitude
nzWRFCheck=5
#Define number of layers in WRF simulations
#nzLayersWRF=13           #For low vertical resolution (45 Levels)
nzLayersWRF=17           #For mid vertical resolution (90 Levels)
#nzLayersWRF=20           #For high vertical resolution (90 Levels)

#Read the TANAB data files
InputTANABData1 = np.loadtxt(InputTANABfile1)
timeTANAB1=InputTANABData1[:,0]
zMeanTANAB1=InputTANABData1[:,3]
SMeanTANAB1=InputTANABData1[:,6]
ThetaMeanTANAB1=InputTANABData1[:,7]
dThetadzBTANAB1=InputTANABData1[:,8]

InputTANABData2 = np.loadtxt(InputTANABfile2)
timeTANAB2=InputTANABData2[:,0]
zMeanTANAB2=InputTANABData2[:,3]
SMeanTANAB2=InputTANABData2[:,6]
ThetaMeanTANAB2=InputTANABData2[:,7]
dThetadzBTANAB2=InputTANABData2[:,8]

InputTANABData3 = np.loadtxt(InputTANABfile3)
timeTANAB3=InputTANABData3[:,0]
zMeanTANAB3=InputTANABData3[:,3]
SMeanTANAB3=InputTANABData3[:,6]
ThetaMeanTANAB3=InputTANABData3[:,7]
dThetadzBTANAB3=InputTANABData3[:,8]

#Read the WRF data files
InputWRFdata1 = np.loadtxt(InputWRFfile1)
timeWRF1=InputWRFdata1[:,0]
zWRF1=InputWRFdata1[:,1]
SWRF1=InputWRFdata1[:,4]
PertThetaWRF1=InputWRFdata1[:,8]

InputWRFdata2 = np.loadtxt(InputWRFfile2)
timeWRF2=InputWRFdata2[:,0]
zWRF2=InputWRFdata2[:,1]
SWRF2=InputWRFdata2[:,4]
PertThetaWRF2=InputWRFdata2[:,8]

InputWRFdata3 = np.loadtxt(InputWRFfile3)
timeWRF3=InputWRFdata3[:,0]

```

```

zWRF3=InputWRFdata3[:,1]
SWRF3=InputWRFdata3[:,4]
PertThetaWRF3=InputWRFdata3[:,8]

#####

SMean50TANAB=np.zeros((nTimes,1))
SMean25TANAB=np.zeros((nTimes,1))
SMean75TANAB=np.zeros((nTimes,1))
SMean50WRF=np.zeros((nTimes,1))
SMean25WRF=np.zeros((nTimes,1))
SMean75WRF=np.zeros((nTimes,1))
SMeanFBTANABWRF=np.zeros((nTimes,1))
SMeanNMSETANABWRF=np.zeros((nTimes,1))
SMeanBTANABWRF=np.zeros((nTimes,1))

dThetadzB50TANAB=np.zeros((nTimes,1))
dThetadzB25TANAB=np.zeros((nTimes,1))
dThetadzB75TANAB=np.zeros((nTimes,1))
dThetadzB50WRF=np.zeros((nTimes,1))
dThetadzB25WRF=np.zeros((nTimes,1))
dThetadzB75WRF=np.zeros((nTimes,1))
dThetadzBFBTANABWRF=np.zeros((nTimes,1))
dThetadzBNMSETANABWRF=np.zeros((nTimes,1))
dThetadzBBTANABWRF=np.zeros((nTimes,1))

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):

    #Look up TANAB Wind Speed
    ArraySMeanTANAB=np.concatenate((SMeanTANAB1[j],SMeanTANAB2[j],SMeanTANAB3[j]),
                                     axis=None)

    SMean50TANAB[j] = np.nanmedian(ArraySMeanTANAB)
    SMean25TANAB[j] = np.nanpercentile(ArraySMeanTANAB,25)
    SMean75TANAB[j] = np.nanpercentile(ArraySMeanTANAB, 75)

    #Look up TANAB temperature gradients
    ArraydThetadzBTANAB=np.concatenate((dThetadzBTANAB1[j],dThetadzBTANAB2[j],
                                           dThetadzBTANAB3[j]),axis=None)

    dThetadzB50TANAB[j] = np.nanmedian(ArraydThetadzBTANAB)
    dThetadzB25TANAB[j] = np.nanpercentile(ArraydThetadzBTANAB,25)
    dThetadzB75TANAB[j] = np.nanpercentile(ArraydThetadzBTANAB, 75)

    #Find the closest altitude in WRF associated with Mean TANAB altitude for each
    day
    #Check the first few layers of WRF for the closest match
    ArrayElementSWRF1=[np.nan]
    ArrayElementSWRF2=[np.nan]
    ArrayElementSWRF3=[np.nan]
    ArrayElementdThetadzBWRF1=[np.nan]

```

```

ArrayElementdThetadzBWRF2 = [np.nan]
ArrayElementdThetadzBWRF3 = [np.nan]
for i in range(1, nzWRFCheck):
    if np.isnan(zMeanTANAB1[j]):
        break
    elif zMeanTANAB1[j] < (zWRF1[(j)*nzLayersWRF+i]+zWRF1[(j)*nzLayersWRF+i-1])
        /2:
        iMatch=i-1
        #Now start making the arrays of data
        ArrayElementSWRF1 = SWRF1[(j)*nzLayersWRF+iMatch]
        #Option 1: use the closest vertical location for calculating the
            gradient
        #Option 2: always use the first two layers for calculating the
            gradient
        #iMatch=0
        ArrayElementdThetadzBWRF1 = (PertThetaWRF1[(j)*nzLayersWRF+iMatch+1]-
            PertThetaWRF1[(j)*nzLayersWRF+iMatch]) / (zWRF1[(j)*nzLayersWRF+
            iMatch+1]-zWRF1[(j)*nzLayersWRF+iMatch])

        break

for i in range(1, nzWRFCheck):
    if np.isnan(zMeanTANAB2[j]):
        break
    elif zMeanTANAB2[j] < (zWRF2[(j) * nzLayersWRF + i] + zWRF2[(j) *
        nzLayersWRF + i - 1]) / 2:
        iMatch = i - 1
        # Now start making the arrays of data
        ArrayElementSWRF2 = SWRF2[(j) * nzLayersWRF + iMatch]
        #Option 1: use the closest vertical location for calculating the
            gradient
        #Option 2: always use the first two layers for calculating the
            gradient
        #iMatch=0
        ArrayElementdThetadzBWRF2 = (PertThetaWRF2[(j) * nzLayersWRF + iMatch
            + 1] - PertThetaWRF2[(j) * nzLayersWRF + iMatch]) / (zWRF2[(j) *
            nzLayersWRF + iMatch + 1] - zWRF2[(j) * nzLayersWRF + iMatch])
        break

for i in range(1, nzWRFCheck):
    if np.isnan(zMeanTANAB3[j]):
        break
    elif zMeanTANAB3[j] < (zWRF3[(j) * nzLayersWRF + i] + zWRF3[(j) *
        nzLayersWRF + i - 1]) / 2:
        iMatch = i - 1
        # Now start making the arrays of data
        ArrayElementSWRF3 = SWRF3[(j) * nzLayersWRF + iMatch]
        #Option 1: use the closest vertical location for calculating the
            gradient
        #Option 2: always use the first two layers for calculating the
            gradient

```

```

#iMatch=0
ArrayElementdThetadzBWRf3 = (PertThetaWRF3[(j) * nzLayersWRF + iMatch
+ 1] - PertThetaWRF3[(j) * nzLayersWRF + iMatch]) / (zWRF3[(j) *
nzLayersWRF + iMatch + 1] - zWRF3[(j) * nzLayersWRF + iMatch])
break

#Now assemble the arrays
ArraySWRF = np.concatenate((ArrayElementSWRF1, ArrayElementSWRF2,
ArrayElementSWRF3), axis=None)
ArraydThetadzBWRf = np.concatenate((ArrayElementdThetadzBWRf1,
ArrayElementdThetadzBWRf2, ArrayElementdThetadzBWRf3), axis=None)

SMean50WRF[j] = np.nanmedian(ArraySWRF)
SMean25WRF[j] = np.nanpercentile(ArraySWRF, 25)
SMean75WRF[j] = np.nanpercentile(ArraySWRF, 75)

dThetadzB50WRF[j] = np.nanmedian(ArraydThetadzBWRf)
dThetadzB25WRF[j] = np.nanpercentile(ArraydThetadzBWRf, 25)
dThetadzB75WRF[j] = np.nanpercentile(ArraydThetadzBWRf, 75)

SMeanFBTANABWRF[j] = (np.nanmean(ArraySWRF) - np.nanmean(ArraySMeanTANAB))
/ (0.5 * (np.nanmean(ArraySWRF) + np.nanmean(ArraySMeanTANAB)))
SMeanNMSETANABWRF[j] = np.absolute((np.nanmean(ArraySWRF - ArraySMeanTANAB) ** 2)
/ ((np.nanmean(ArraySWRF) * np.nanmean(ArraySMeanTANAB))))
SMeanBTANABWRF[j] = (np.nanmean(ArraySWRF) - np.nanmean(ArraySMeanTANAB))

dThetadzBFBTANABWRF[j] = (np.nanmean(ArraydThetadzBWRf) - np.nanmean(
ArraydThetadzBTANAB)) / (0.5 * (np.nanmean(ArraydThetadzBWRf) + np.nanmean(
ArraydThetadzBTANAB)))
dThetadzBNMSETANABWRF[j] = np.absolute((np.nanmean(ArraydThetadzBWRf -
ArraydThetadzBTANAB) ** 2) / ((np.nanmean(ArraydThetadzBWRf) * np.nanmean(
ArraydThetadzBTANAB))))
dThetadzBBTANABWRF[j] = (np.nanmean(ArraydThetadzBWRf) - np.nanmean(
ArraydThetadzBTANAB))

# Calculate the entire fractional bias and normalized mean square error of SMean for all the
data of horizontal wind speed

print('Bias, Fractional Bias (FB) and Normalized Mean Square Error (NMSE) for all the
medians of Horizontal Wind Speed')
print('SMean50FB=', (np.nanmean(SMean50WRF) - np.nanmean(SMean50TANAB)) / (0.5 * (np.nanmean(
SMean50WRF) + np.nanmean(SMean50TANAB))))
print('SMean50NMSE=', np.absolute((np.nanmean(SMean50WRF - SMean50TANAB) ** 2) / ((np.nanmean(
SMean50WRF) * np.nanmean(SMean50TANAB)))))
print('SMean50Bias=', (np.nanmean(SMean50WRF) - np.nanmean(SMean50TANAB)))

# Calculate the entire fractional bias and normalized mean square error of dThetadzB for all
the data of Temperature Gradient

print('Bias, Fractional Bias (FB) and Normalized Mean Square Error (NMSE) for all the
medians of Temperature Gradient')
print('dThetadzB50FB=', (np.nanmean(dThetadzB50WRF) - np.nanmean(dThetadzB50TANAB)) / (0.5 *

```

[illegible]

```

#Define number of time windows (1 hour) for comparison
nTimes = 24

TimeWindow=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]

#Number of first few layers of WRF altitudes to check for matching Mean TANAB altitude
nzWRFCheck=5
#Define number of layers in WRF simulations
#nzLayersWRF=13           #For low vertical resolution (45 Levels)
nzLayersWRF=17           #For mid vertical resolution (90 Levels)
#nzLayersWRF=20           #For high vertical resolution (90 Levels)

#Read the TANAB data files
InputTANABData1 = np.loadtxt(InputTANABfile1)
timeTANAB1=InputTANABData1[:,0]
zMeanTANAB1=InputTANABData1[:,3]
SMeanTANAB1=InputTANABData1[:,6]
ThetaMeanTANAB1=InputTANABData1[:,7]
dThetadzBTANAB1=InputTANABData1[:,8]

InputTANABData2 = np.loadtxt(InputTANABfile2)
timeTANAB2=InputTANABData2[:,0]
zMeanTANAB2=InputTANABData2[:,3]
SMeanTANAB2=InputTANABData2[:,6]
ThetaMeanTANAB2=InputTANABData2[:,7]
dThetadzBTANAB2=InputTANABData2[:,8]

InputTANABData3 = np.loadtxt(InputTANABfile3)
timeTANAB3=InputTANABData3[:,0]
zMeanTANAB3=InputTANABData3[:,3]
SMeanTANAB3=InputTANABData3[:,6]
ThetaMeanTANAB3=InputTANABData3[:,7]
dThetadzBTANAB3=InputTANABData3[:,8]

#Read the WRF data files
InputWRFdata1 = np.loadtxt(InputWRFfile1)
timeWRF1=InputWRFdata1[:,0]
zWRF1=InputWRFdata1[:,1]
SWRF1=InputWRFdata1[:,4]
PertThetaWRF1=InputWRFdata1[:,8]

InputWRFdata2 = np.loadtxt(InputWRFfile2)
timeWRF2=InputWRFdata2[:,0]
zWRF2=InputWRFdata2[:,1]
SWRF2=InputWRFdata2[:,4]
PertThetaWRF2=InputWRFdata2[:,8]

InputWRFdata3 = np.loadtxt(InputWRFfile3)
timeWRF3=InputWRFdata3[:,0]
zWRF3=InputWRFdata3[:,1]
SWRF3=InputWRFdata3[:,4]

```



```

PertThetaWRF3=InputWRFdata3[:,8]

#####

SMeanTANAB=np.nan
SMeanWRF=np.nan

dThetadzBTANAB=np.nan
dThetadzBWRF=np.nan

#Time loop j: from 0 to nTimes
for j in range(0, nTimes):

    #Look up TANAB Wind Speed
    SMeanTANAB=np.concatenate((SMeanTANAB,SMeanTANAB1[j],SMeanTANAB2[j],
                                SMeanTANAB3[j]),axis=None)

    #Look up TANAB temperature gradients
    dThetadzBTANAB=np.concatenate((dThetadzBTANAB,dThetadzBTANAB1[j],
                                    dThetadzBTANAB2[j],dThetadzBTANAB3[j]),axis=None)

    #Find the closest altitude in WRF associated with Mean TANAB altitude for each
    day
    #Check the first few layers of WRF for the closest match
    ArrayElementSWRF1=[np.nan]
    ArrayElementSWRF2=[np.nan]
    ArrayElementSWRF3=[np.nan]
    ArrayElementdThetadzBWRF1=[np.nan]
    ArrayElementdThetadzBWRF2=[np.nan]
    ArrayElementdThetadzBWRF3=[np.nan]
    for i in range(1, nzWRFCheck):
        if np.isnan(zMeanTANAB1[j]):
            break
        elif zMeanTANAB1[j]<(zWRF1[(j)*nzLayersWRF+i]+zWRF1[(j)*nzLayersWRF+i-1])
            /2:
            iMatch=i-1
            #Now start making the arrays of data
            ArrayElementSWRF1=SWRF1[(j)*nzLayersWRF+iMatch]
            #Option 1: use the closest vertical location for calculating the
            gradient
            #Option 2: always use the first two layers for calculating the
            gradient
            #iMatch=0
            ArrayElementdThetadzBWRF1=(PertThetaWRF1[(j)*nzLayersWRF+iMatch+1]-
                PertThetaWRF1[(j)*nzLayersWRF+iMatch])/(zWRF1[(j)*nzLayersWRF+
                iMatch+1]-zWRF1[(j)*nzLayersWRF+iMatch])

            break

    for i in range(1, nzWRFCheck):
        if np.isnan(zMeanTANAB2[j]):

```

```

        break
    elif zMeanTANAB2[j] < (zWRF2[(j) * nzLayersWRF + i] + zWRF2[(j) *
        nzLayersWRF + i - 1]) / 2:
        iMatch = i - 1
        # Now start making the arrays of data
        ArrayElementSWRF2 = SWRF2[(j) * nzLayersWRF + iMatch]
        #Option 1: use the closest vertical location for calculating the
            gradient
        #Option 2: always use the first two layers for calculating the
            gradient
        #iMatch=0
        ArrayElementdTetadzBWRF2 = (PertThetaWRF2[(j) * nzLayersWRF + iMatch
            + 1] - PertThetaWRF2[(j) * nzLayersWRF + iMatch]) / (zWRF2[(j) *
            nzLayersWRF + iMatch + 1] - zWRF2[(j) * nzLayersWRF + iMatch])
        break

for i in range(1, nzWRFCheck):
    if np.isnan(zMeanTANAB3[j]):
        break
    elif zMeanTANAB3[j] < (zWRF3[(j) * nzLayersWRF + i] + zWRF3[(j) *
        nzLayersWRF + i - 1]) / 2:
        iMatch = i - 1
        # Now start making the arrays of data
        ArrayElementSWRF3 = SWRF3[(j) * nzLayersWRF + iMatch]
        #Option 1: use the closest vertical location for calculating the
            gradient
        #Option 2: always use the first two layers for calculating the
            gradient
        #iMatch=0
        ArrayElementdTetadzBWRF3 = (PertThetaWRF3[(j) * nzLayersWRF + iMatch
            + 1] - PertThetaWRF3[(j) * nzLayersWRF + iMatch]) / (zWRF3[(j) *
            nzLayersWRF + iMatch + 1] - zWRF3[(j) * nzLayersWRF + iMatch])
        break

#Now assemble the arrays
SMeanWRF = np.concatenate((SMeanWRF, ArrayElementSWRF1, ArrayElementSWRF2,
    ArrayElementSWRF3), axis=None)
dThetadzBWRF = np.concatenate((dThetadzBWRF, ArrayElementdTetadzBWRF1,
    ArrayElementdTetadzBWRF2, ArrayElementdTetadzBWRF3), axis=None)

#Calculate Bias and RMSE
SBias = np.nanmean(SMeanWRF-SMeanTANAB)
SRMSE = np.sqrt(np.nanmean((SMeanWRF-SMeanTANAB)**2))

dThetadzBBias = np.nanmean(dThetadzBWRF-dThetadzBTANAB)
dThetadzRMSE = np.sqrt(np.nanmean((dThetadzBWRF-dThetadzBTANAB)**2))

#Print the entire bias and root mean square error of S10m for all the data of horizontal
    wind speed at 10 m

print(' Bias , and Root Mean Square Error (RMSE) for all the medians of Horizontal Wind Speed '
    )

```

```

print( 'SBias=', SBias)
print( 'SRMSE=', SRMSE)

#Print the entire bias for all the data of Temperature gradient

print( 'Bias_(B)_and_RMSE_for_all_the_differences_of_Temperature_gradient_(K/m)')
print( 'dThetadzBias=', dThetadzBBias)
print( 'dThetadzRMSE=', dThetadzRMSE)

```

## D.2.5 WRF Compare to Surface Level for Percentile Calculations

```

import matplotlib.pyplot as plt
import numpy as np

#Define integral time-scales of turbulence [s]
TauSHF=180

#Define sampling time for instruments measuring turbulence variances and fluxes [s]
TAvgSHF=900

#Calculate correction factor for turbulent variances and fluxes
CorrFactSHF=-1/(2*TauSHF/TAvgSHF-1)

#Define magnetic declination for northern Alberta
#If you rerun this code add this to the experimental wind direction
#Magnetic North is to the east of true North
phiMagDec=14

#Define output filename associated with WRF/Surface Level comparison
Outputfile="StatisticalAnalysis/WRFCompareToUofASonic1_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToUofASonic2_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToMineTrailer_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToRWDISonic1_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToRWDISonic2_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToRWDISonic3_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToWBEACup_95_96_97.txt"

#Define filenames associated with surface level measurements on May 18, 24, 30

InputSurfaceLevelfile1="Instruments/UofASonic120180518.txt"
InputSurfaceLevelfile2="Instruments/UofASonic120180524.txt"
InputSurfaceLevelfile3="Instruments/UofASonic120180530.txt"

#InputSurfaceLevelfile1="Instruments/UofASonic220180518.txt"
#InputSurfaceLevelfile2="Instruments/UofASonic220180524.txt"
#InputSurfaceLevelfile3="Instruments/UofASonic220180530.txt"

#InputSurfaceLevelfile1="Instruments/MineTrailer20180518.txt"
#InputSurfaceLevelfile2="Instruments/MineTrailer20180524.txt"
#InputSurfaceLevelfile3="Instruments/MineTrailer20180530.txt"

```

```

#InputSurfaceLevelfile1="Instruments/RWDISonic120180518.txt"
#InputSurfaceLevelfile2="Instruments/RWDISonic120180524.txt"
#InputSurfaceLevelfile3="Instruments/RWDISonic120180530.txt"

#InputSurfaceLevelfile1="Instruments/RWDISonic220180518.txt"
#InputSurfaceLevelfile2="Instruments/RWDISonic220180524.txt"
#InputSurfaceLevelfile3="Instruments/RWDISonic220180530.txt"

#InputSurfaceLevelfile1="Instruments/RWDISonic320180518.txt"
#InputSurfaceLevelfile2="Instruments/RWDISonic320180524.txt"
#InputSurfaceLevelfile3="Instruments/RWDISonic320180530.txt"

#InputSurfaceLevelfile1="Instruments/WBEACup20180518.txt"
#InputSurfaceLevelfile2="Instruments/WBEACup20180524.txt"
#InputSurfaceLevelfile3="Instruments/WBEACup20180530.txt"

#Define filenames associated with WRF runs of interest on May 18, 24, 30

InputWRFfile1="Mine_95/wrfout_2018-05-18_UltrasonicAnemometer1.txt"
InputWRFfile2="Mine_96/wrfout_2018-05-24_UltrasonicAnemometer1.txt"
InputWRFfile3="Mine_97/wrfout_2018-05-30_UltrasonicAnemometer1.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_UltrasonicAnemometer2.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_UltrasonicAnemometer2.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_UltrasonicAnemometer2.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_MineTrailer.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_MineTrailer.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_MineTrailer.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_RWDISonicAnemometer1.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_RWDISonicAnemometer1.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_RWDISonicAnemometer1.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_RWDISonicAnemometer2.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_RWDISonicAnemometer2.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_RWDISonicAnemometer2.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_RWDISonicAnemometer3.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_RWDISonicAnemometer3.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_RWDISonicAnemometer3.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_WBEACupAnemometer.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_WBEACupAnemometer.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_WBEACupAnemometer.txt"

#Define number of time windows (1 hour) for comparison
nTimes = 24

TimeWindow=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]

#Read the surface level data files

```

```

InputSurfaceLevelData1 = np.loadtxt(InputSurfaceLevelfile1)
timeSurfaceLevel1=InputSurfaceLevelData1[:,0]
T2mSurfaceLevel1=InputSurfaceLevelData1[:,3]+273.15
RH2mSurfaceLevel1=InputSurfaceLevelData1[:,4]
S2mSurfaceLevel1=InputSurfaceLevelData1[:,5]
WD2mSurfaceLevel1=InputSurfaceLevelData1[:,6]
T10mSurfaceLevel1=InputSurfaceLevelData1[:,7]+273.15
S10mSurfaceLevel1=InputSurfaceLevelData1[:,8]
#If you rerun the surface level comparison add the magnetic declination angle here
WD10mSurfaceLevel1=InputSurfaceLevelData1[:,9]
SHF10mSurfaceLevel1=InputSurfaceLevelData1[:,10]*CorrFactSHF

InputSurfaceLevelData2 = np.loadtxt(InputSurfaceLevelfile2)
timeSurfaceLevel2=InputSurfaceLevelData2[:,0]
T2mSurfaceLevel2=InputSurfaceLevelData2[:,3]+273.15
RH2mSurfaceLevel2=InputSurfaceLevelData2[:,4]
S2mSurfaceLevel2=InputSurfaceLevelData2[:,5]
WD2mSurfaceLevel2=InputSurfaceLevelData2[:,6]
T10mSurfaceLevel2=InputSurfaceLevelData2[:,7]+273.15
S10mSurfaceLevel2=InputSurfaceLevelData2[:,8]
#If you rerun the surface level comparison add the magnetic declination angle here
WD10mSurfaceLevel2=InputSurfaceLevelData2[:,9]
SHF10mSurfaceLevel2=InputSurfaceLevelData2[:,10]*CorrFactSHF

InputSurfaceLevelData3 = np.loadtxt(InputSurfaceLevelfile3)
timeSurfaceLevel3=InputSurfaceLevelData3[:,0]
T2mSurfaceLevel3=InputSurfaceLevelData3[:,3]+273.15
RH2mSurfaceLevel3=InputSurfaceLevelData3[:,4]
S2mSurfaceLevel3=InputSurfaceLevelData3[:,5]
WD2mSurfaceLevel3=InputSurfaceLevelData3[:,6]
T10mSurfaceLevel3=InputSurfaceLevelData3[:,7]+273.15
S10mSurfaceLevel3=InputSurfaceLevelData3[:,8]
#If you rerun the surface level comparison add the magnetic declination angle here
WD10mSurfaceLevel3=InputSurfaceLevelData3[:,9]
SHF10mSurfaceLevel3=InputSurfaceLevelData3[:,10]*CorrFactSHF

#Read the WRF data files
InputWRFdata1 = np.loadtxt(InputWRFfile1)
timeWRF1=InputWRFdata1[:,0]
S10mWRF1=InputWRFdata1[:,1]
WD10mWRF1=InputWRFdata1[:,4]
T2mWRF1=InputWRFdata1[:,5]
RH2mWRF1=InputWRFdata1[:,7]
SHF10mWRF1=InputWRFdata1[:,8]

InputWRFdata2 = np.loadtxt(InputWRFfile2)
timeWRF2=InputWRFdata2[:,0]
S10mWRF2=InputWRFdata2[:,1]
WD10mWRF2=InputWRFdata2[:,4]
T2mWRF2=InputWRFdata2[:,5]
RH2mWRF2=InputWRFdata2[:,7]
SHF10mWRF2=InputWRFdata2[:,8]

```

```

InputWRFdata3 = np.loadtxt(InputWRFfile3)
timeWRF3=InputWRFdata3[:,0]
S10mWRF3=InputWRFdata3[:,1]
WD10mWRF3=InputWRFdata3[:,4]
T2mWRF3=InputWRFdata3[:,5]
RH2mWRF3=InputWRFdata3[:,7]
SHF10mWRF3=InputWRFdata3[:,8]

#####

S10mSurfacelevel=np.zeros((3*nTimes,1))
S10mWRF=np.zeros((3*nTimes,1))

T2mSurfacelevel=np.zeros((3*nTimes,1))
T2mWRF=np.zeros((3*nTimes,1))

RH2mSurfacelevel=np.zeros((3*nTimes,1))
RH2mWRF=np.zeros((3*nTimes,1))

SHF10mSurfacelevel=np.zeros((3*nTimes,1))
SHF10mWRF=np.zeros((3*nTimes,1))

WD10mSurfaceLevel=np.zeros((3*nTimes,1))
WD10mWRF=np.zeros((3*nTimes,1))

#Create arrays with 72 measurements and WRF outputs for Wind velocity

S10mSurfaceLevel123=np.concatenate((S10mSurfaceLevel1,S10mSurfaceLevel2,S10mSurfaceLevel3),
axis=None)
S10mWRF123 = np.concatenate((S10mWRF1, S10mWRF2, S10mWRF3),axis=None)

#Calculate Bias and RMSE
S10Bias123 = np.nanmean(S10mWRF123-S10mSurfaceLevel123)
S10RMSE123 = np.sqrt(np.nanmean((S10mWRF123-S10mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs for Temperature

T2mSurfaceLevel123=np.concatenate((T2mSurfaceLevel1,T2mSurfaceLevel2,T2mSurfaceLevel3),axis=
None)
T2mWRF123 = np.concatenate((T2mWRF1, T2mWRF2, T2mWRF3),axis=None)

#Calculate Bias and RMSE
T2Bias123 = np.nanmean(T2mWRF123-T2mSurfaceLevel123)
T2RMSE123 = np.sqrt(np.nanmean((T2mWRF123-T2mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs for Relative Humidity

RH2mSurfaceLevel123=np.concatenate((RH2mSurfaceLevel1,RH2mSurfaceLevel2,RH2mSurfaceLevel3),
axis=None)
RH2mWRF123 = np.concatenate((RH2mWRF1, RH2mWRF2, RH2mWRF3),axis=None)

```

```

#Calculate Bias and RMSE
RH2Bias123 = np.nanmean(RH2mWRF123-RH2mSurfaceLevel123)
RH2RMSE123 = np.sqrt(np.nanmean((RH2mWRF123-RH2mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs for Sensible Heat Flux

SHF10mSurfaceLevel123=np.concatenate((SHF10mSurfaceLevel1,SHF10mSurfaceLevel2,
    SHF10mSurfaceLevel3),axis=None)
SHF10mWRF123 = np.concatenate((SHF10mWRF1, SHF10mWRF2, SHF10mWRF3),axis=None)

#Calculate Bias and RMSE
SHF10Bias123 = np.nanmean(SHF10mWRF123-SHF10mSurfaceLevel123)
SHF10RMSE123 = np.sqrt(np.nanmean((SHF10mWRF123-SHF10mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs of wind direction

WD10mSurfaceLevel123 = np.concatenate((WD10mSurfaceLevel1, WD10mSurfaceLevel2,
    WD10mSurfaceLevel3), axis=None)
WD10mWRF123 = np.concatenate((WD10mWRF1, WD10mWRF2, WD10mWRF3), axis=None)

#Correct for magnetic inclination
WD10mSurfaceLevel123 = WD10mSurfaceLevel123 + phiMagDec

for x in range(0, len(WD10mSurfaceLevel123)):
    if WD10mSurfaceLevel123[x] >= 360:
        WD10mSurfaceLevel123[x] = WD10mSurfaceLevel123[x] - 360

#Calculate difference in WD
DiffPhi10mSurfacelevelWRF123 = WD10mWRF123 - WD10mSurfaceLevel123

for x in range(0, len(DiffPhi10mSurfacelevelWRF123)):
    if DiffPhi10mSurfacelevelWRF123[x] <= -180:
        DiffPhi10mSurfacelevelWRF123[x] = DiffPhi10mSurfacelevelWRF123[x] + 360
    if DiffPhi10mSurfacelevelWRF123[x] >= 180:
        DiffPhi10mSurfacelevelWRF123[x] = DiffPhi10mSurfacelevelWRF123[x] - 360

#Calculate Bias and RMSE
WD10Bias123 = np.nanmean(DiffPhi10mSurfacelevelWRF123)
WD10RMSE123 = np.sqrt(np.nanmean((DiffPhi10mSurfacelevelWRF123)**2))

#Print the entire bias and root mean square error of S10m for all the data of horizontal
    wind speed at 10 m

print('Bias , and Root Mean Square Error (RMSE) for all the medians of Horizontal Wind Speed'
    )
print('S10mBias=', S10Bias123)
print('S10mRMSE=', S10RMSE123)

# Print the entire bias and root mean square error of T2m for all the data of temperature at

```

```

2 m

print( 'Bias , and Root Mean Square Error (RMSE) for all the medians of Temperature ')
print( 'T2mBias= ', T2Bias123 )
print( 'T2mRMSE= ', T2RMSE123 )

# Print the entire bias and root mean square error of RH2m for all the data of relative
  humidity at 2 m

print( 'Bias , and Root Mean Square Error (RMSE) for all the medians of Relative Humidity ')
print( 'RH2mBias= ', RH2Bias123 )
print( 'RH2mNMSE= ', RH2RMSE123 )

# Print the entire bias and root mean square error of SHF10m for all the data of Sensible
  Heat Flux at 10 m

print( 'Bias , Fractional Bias (FB) and Normalized Mean Square Error (NMSE) for all the
  medians of Sensible Heat Flux ')
print( 'SHF10mBias= ', SHF10Bias123 )
print( 'SHF10mRMSE= ', SHF10RMSE123 )

# Print the entire bias for all the data of Wind Direction at 10m

print( 'Bias (B) and RMSE for all the differences of wind direction (Degree) ')
print( 'WD10m50Bias= ', WD10Bias123 )
print( 'WD10m50RMSE= ', WD10RMSE123 )

```

## D.2.6 WRF Compare to Surface Level for Error Calculations

```

import matplotlib.pyplot as plt
import numpy as np

#Define integral time-scales of turbulence [s]
TauSHF=180

#Define sampling time for instruments measuring turbulence variances and fluxes [s]
TAvgSHF=900

#Calculate correction factor for turbulent variances and fluxes
CorrFactSHF=-1/(2*TauSHF/TAvgSHF-1)

#Define magnetic declination for northern Alberta
#If you rerun this code add this to the experimental wind direction
#Magnetic North is to the east of true North
phiMagDec=14

#Define output filename associated with WRF/Surface Level comparison
Outputfile="StatisticalAnalysis/WRFCompareToUofASonic1_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToUofASonic2_95_96_97.txt"

```



```

#Outputfile="StatisticalAnalysis/WRFCompareToMineTrailer_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToRWDISonic1_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToRWDISonic2_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToRWDISonic3_95_96_97.txt"
#Outputfile="StatisticalAnalysis/WRFCompareToWBEACup_95_96_97.txt"

#Define filenames associated with surface level measurements on May 18, 24, 30

InputSurfaceLevelfile1="Instruments/UofASonic120180518.txt"
InputSurfaceLevelfile2="Instruments/UofASonic120180524.txt"
InputSurfaceLevelfile3="Instruments/UofASonic120180530.txt"

#InputSurfaceLevelfile1="Instruments/UofASonic220180518.txt"
#InputSurfaceLevelfile2="Instruments/UofASonic220180524.txt"
#InputSurfaceLevelfile3="Instruments/UofASonic220180530.txt"

#InputSurfaceLevelfile1="Instruments/MineTrailer20180518.txt"
#InputSurfaceLevelfile2="Instruments/MineTrailer20180524.txt"
#InputSurfaceLevelfile3="Instruments/MineTrailer20180530.txt"

#InputSurfaceLevelfile1="Instruments/RWDISonic120180518.txt"
#InputSurfaceLevelfile2="Instruments/RWDISonic120180524.txt"
#InputSurfaceLevelfile3="Instruments/RWDISonic120180530.txt"

#InputSurfaceLevelfile1="Instruments/RWDISonic220180518.txt"
#InputSurfaceLevelfile2="Instruments/RWDISonic220180524.txt"
#InputSurfaceLevelfile3="Instruments/RWDISonic220180530.txt"

#InputSurfaceLevelfile1="Instruments/RWDISonic320180518.txt"
#InputSurfaceLevelfile2="Instruments/RWDISonic320180524.txt"
#InputSurfaceLevelfile3="Instruments/RWDISonic320180530.txt"

#InputSurfaceLevelfile1="Instruments/WBEACup20180518.txt"
#InputSurfaceLevelfile2="Instruments/WBEACup20180524.txt"
#InputSurfaceLevelfile3="Instruments/WBEACup20180530.txt"

#Define filenames associated with WRF runs of interest on May 18, 24, 30

InputWRFfile1="Mine_95/wrfout_2018-05-18_UltrasonicAnemometer1.txt"
InputWRFfile2="Mine_96/wrfout_2018-05-24_UltrasonicAnemometer1.txt"
InputWRFfile3="Mine_97/wrfout_2018-05-30_UltrasonicAnemometer1.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_UltrasonicAnemometer2.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_UltrasonicAnemometer2.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_UltrasonicAnemometer2.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_MineTrailer.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_MineTrailer.txt"
#InputWRFfile3="Mine_97/wrfout_2018-05-30_MineTrailer.txt"

#InputWRFfile1="Mine_95/wrfout_2018-05-18_RWDISonicAnemometer1.txt"
#InputWRFfile2="Mine_96/wrfout_2018-05-24_RWDISonicAnemometer1.txt"

```

```

#InputWRFfile3="Mine 97/wrfout_2018-05-30.RWDISonicAnemometer1.txt"

#InputWRFfile1="Mine 95/wrfout_2018-05-18.RWDISonicAnemometer2.txt"
#InputWRFfile2="Mine 96/wrfout_2018-05-24.RWDISonicAnemometer2.txt"
#InputWRFfile3="Mine 97/wrfout_2018-05-30.RWDISonicAnemometer2.txt"

#InputWRFfile1="Mine 95/wrfout_2018-05-18.RWDISonicAnemometer3.txt"
#InputWRFfile2="Mine 96/wrfout_2018-05-24.RWDISonicAnemometer3.txt"
#InputWRFfile3="Mine 97/wrfout_2018-05-30.RWDISonicAnemometer3.txt"

#InputWRFfile1="Mine 95/wrfout_2018-05-18.WBEACupAnemometer.txt"
#InputWRFfile2="Mine 96/wrfout_2018-05-24.WBEACupAnemometer.txt"
#InputWRFfile3="Mine 97/wrfout_2018-05-30.WBEACupAnemometer.txt"

#Define number of time windows (1 hour) for comparison
nTimes = 24

TimeWindow=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]

#Read the surface level data files
InputSurfaceLevelData1 = np.loadtxt(InputSurfaceLevelfile1)
timeSurfaceLevel1=InputSurfaceLevelData1[:,0]
T2mSurfaceLevel1=InputSurfaceLevelData1[:,3]+273.15
RH2mSurfaceLevel1=InputSurfaceLevelData1[:,4]
S2mSurfaceLevel1=InputSurfaceLevelData1[:,5]
WD2mSurfaceLevel1=InputSurfaceLevelData1[:,6]
T10mSurfaceLevel1=InputSurfaceLevelData1[:,7]+273.15
S10mSurfaceLevel1=InputSurfaceLevelData1[:,8]
#If you rerun the surface level comparison add the magnetic declination angle here
WD10mSurfaceLevel1=InputSurfaceLevelData1[:,9]
SHF10mSurfaceLevel1=InputSurfaceLevelData1[:,10]*CorrFactSHF

InputSurfaceLevelData2 = np.loadtxt(InputSurfaceLevelfile2)
timeSurfaceLevel2=InputSurfaceLevelData2[:,0]
T2mSurfaceLevel2=InputSurfaceLevelData2[:,3]+273.15
RH2mSurfaceLevel2=InputSurfaceLevelData2[:,4]
S2mSurfaceLevel2=InputSurfaceLevelData2[:,5]
WD2mSurfaceLevel2=InputSurfaceLevelData2[:,6]
T10mSurfaceLevel2=InputSurfaceLevelData2[:,7]+273.15
S10mSurfaceLevel2=InputSurfaceLevelData2[:,8]
#If you rerun the surface level comparison add the magnetic declination angle here
WD10mSurfaceLevel2=InputSurfaceLevelData2[:,9]
SHF10mSurfaceLevel2=InputSurfaceLevelData2[:,10]*CorrFactSHF

InputSurfaceLevelData3 = np.loadtxt(InputSurfaceLevelfile3)
timeSurfaceLevel3=InputSurfaceLevelData3[:,0]
T2mSurfaceLevel3=InputSurfaceLevelData3[:,3]+273.15
RH2mSurfaceLevel3=InputSurfaceLevelData3[:,4]
S2mSurfaceLevel3=InputSurfaceLevelData3[:,5]
WD2mSurfaceLevel3=InputSurfaceLevelData3[:,6]
T10mSurfaceLevel3=InputSurfaceLevelData3[:,7]+273.15
S10mSurfaceLevel3=InputSurfaceLevelData3[:,8]

```

```

#If you rerun the surface level comparison add the magnetic declination angle here
WD10mSurfaceLevel3=InputSurfaceLevelData3[:,9]
SHF10mSurfaceLevel3=InputSurfaceLevelData3[:,10]*CorrFactSHF

#Read the WRF data files
InputWRFdata1 = np.loadtxt(InputWRFfile1)
timeWRF1=InputWRFdata1[:,0]
S10mWRF1=InputWRFdata1[:,1]
WD10mWRF1=InputWRFdata1[:,4]
T2mWRF1=InputWRFdata1[:,5]
RH2mWRF1=InputWRFdata1[:,7]
SHF10mWRF1=InputWRFdata1[:,8]

InputWRFdata2 = np.loadtxt(InputWRFfile2)
timeWRF2=InputWRFdata2[:,0]
S10mWRF2=InputWRFdata2[:,1]
WD10mWRF2=InputWRFdata2[:,4]
T2mWRF2=InputWRFdata2[:,5]
RH2mWRF2=InputWRFdata2[:,7]
SHF10mWRF2=InputWRFdata2[:,8]

InputWRFdata3 = np.loadtxt(InputWRFfile3)
timeWRF3=InputWRFdata3[:,0]
S10mWRF3=InputWRFdata3[:,1]
WD10mWRF3=InputWRFdata3[:,4]
T2mWRF3=InputWRFdata3[:,5]
RH2mWRF3=InputWRFdata3[:,7]
SHF10mWRF3=InputWRFdata3[:,8]

#####

S10mSurfacelevel=np.zeros((3*nTimes,1))
S10mWRF=np.zeros((3*nTimes,1))

T2mSurfacelevel=np.zeros((3*nTimes,1))
T2mWRF=np.zeros((3*nTimes,1))

RH2mSurfacelevel=np.zeros((3*nTimes,1))
RH2mWRF=np.zeros((3*nTimes,1))

SHF10mSurfacelevel=np.zeros((3*nTimes,1))
SHF10mWRF=np.zeros((3*nTimes,1))

WD10mSurfaceLevel=np.zeros((3*nTimes,1))
WD10mWRF=np.zeros((3*nTimes,1))

#Create arrays with 72 measurements and WRF outputs for Wind velocity

S10mSurfaceLevel123=np.concatenate((S10mSurfaceLevel1,S10mSurfaceLevel2,S10mSurfaceLevel3),
axis=None)
S10mWRF123 = np.concatenate((S10mWRF1, S10mWRF2, S10mWRF3),axis=None)

```

```

#Calculate Bias and RMSE
S10Bias123 = np.nanmean(S10mWRF123-S10mSurfaceLevel123)
S10RMSE123 = np.sqrt(np.nanmean((S10mWRF123-S10mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs for Temperature

T2mSurfaceLevel123=np.concatenate((T2mSurfaceLevel1,T2mSurfaceLevel2,T2mSurfaceLevel3),axis=
    None)
T2mWRF123 = np.concatenate((T2mWRF1, T2mWRF2, T2mWRF3),axis=None)

#Calculate Bias and RMSE
T2Bias123 = np.nanmean(T2mWRF123-T2mSurfaceLevel123)
T2RMSE123 = np.sqrt(np.nanmean((T2mWRF123-T2mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs for Relative Humidity

RH2mSurfaceLevel123=np.concatenate((RH2mSurfaceLevel1,RH2mSurfaceLevel2,RH2mSurfaceLevel3),
    axis=None)
RH2mWRF123 = np.concatenate((RH2mWRF1, RH2mWRF2, RH2mWRF3),axis=None)

#Calculate Bias and RMSE
RH2Bias123 = np.nanmean(RH2mWRF123-RH2mSurfaceLevel123)
RH2RMSE123 = np.sqrt(np.nanmean((RH2mWRF123-RH2mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs for Sensible Heat Flux

SHF10mSurfaceLevel123=np.concatenate((SHF10mSurfaceLevel1,SHF10mSurfaceLevel2,
    SHF10mSurfaceLevel3),axis=None)
SHF10mWRF123 = np.concatenate((SHF10mWRF1, SHF10mWRF2, SHF10mWRF3),axis=None)

#Calculate Bias and RMSE
SHF10Bias123 = np.nanmean(SHF10mWRF123-SHF10mSurfaceLevel123)
SHF10RMSE123 = np.sqrt(np.nanmean((SHF10mWRF123-SHF10mSurfaceLevel123)**2))

#Create arrays with 72 measurements and WRF outputs of wind direction

WD10mSurfaceLevel123 = np.concatenate((WD10mSurfaceLevel1, WD10mSurfaceLevel2,
    WD10mSurfaceLevel3), axis=None)
WD10mWRF123 = np.concatenate((WD10mWRF1, WD10mWRF2, WD10mWRF3), axis=None)

#Correct for magnetic inclination
WD10mSurfaceLevel123 = WD10mSurfaceLevel123 + phiMagDec

for x in range(0, len(WD10mSurfaceLevel123)):
    if WD10mSurfaceLevel123[x] >= 360:
        WD10mSurfaceLevel123[x] = WD10mSurfaceLevel123[x] - 360

#Calculate difference in WD
DiffPhi10mSurfacelevelWRF123 = WD10mWRF123 - WD10mSurfaceLevel123

for x in range(0, len(DiffPhi10mSurfacelevelWRF123)):

```

```

    if DiffPhi10mSurfacelevelWRF123[x] <= -180:
        DiffPhi10mSurfacelevelWRF123[x] = DiffPhi10mSurfacelevelWRF123[x] + 360
    if DiffPhi10mSurfacelevelWRF123[x] >= 180:
        DiffPhi10mSurfacelevelWRF123[x] = DiffPhi10mSurfacelevelWRF123[x] - 360

#Calculate Bias and RMSE
WD10Bias123 = np.nanmean(DiffPhi10mSurfacelevelWRF123)
WD10RMSE123 = np.sqrt(np.nanmean((DiffPhi10mSurfacelevelWRF123)**2))

#Print the entire bias and root mean square error of S10m for all the data of horizontal
    wind speed at 10 m

print( 'Bias ,_and_Root_Mean_Square_Error_(RMSE)_for_all_the_medians_of_Horizontal_Wind_Speed'
    )
print( 'S10mBias=', S10Bias123)
print( 'S10mRMSE=', S10RMSE123)

# Print the entire bias and root mean square error of T2m for all the data of temperature at
    2 m

print( 'Bias ,_and_Root_Mean_Square_Error_(RMSE)_for_all_the_medians_of_Temperature' )
print( 'T2mBias=', T2Bias123)
print( 'T2mRMSE=', T2RMSE123)

# Print the entire bias and root mean square error of RH2m for all the data of relative
    humidity at 2 m

print( 'Bias ,_and_Root_Mean_Square_Error_(RMSE)_for_all_the_medians_of_Relative_Humidity' )
print( 'RH2mBias=', RH2Bias123)
print( 'RH2mRMSE=', RH2RMSE123)

# Print the entire bias and root mean square error of SHF10m for all the data of Sensible
    Heat Flux at 10 m

print( 'Bias ,_Fractional_Bias_(FB)_and_Normalized_Mean_Square_Error_(NMSE)_for_all_the_
    medians_of_Sensible_Heat_Flux' )
print( 'SHF10mBias=', SHF10Bias123)
print( 'SHF10mRMSE=', SHF10RMSE123)

# Print the entire bias for all the data of Wind Direction at 10m

print( 'Bias_(B)_and_RMSE_for_all_the_differences_of_wind_direction_(Degree)' )
print( 'WD10m50Bias=', WD10Bias123)
print( 'WD10m50RMSE=', WD10RMSE123)

```

## D.2.7 Two Samples Statistical Estimation

```

import matplotlib.pyplot as plt
import numpy as np

#Define the Bias, RMSE, sample size, and alpha

x1 = -0.29    #Bias (GTOPO)
s1 = 1.58     #RMSE (GTOPO)
x2 = 0.02     #Bias (SRTM-LIDAR/LU/Lakes)
s2 = 1.57     #RMSE (SRTM-LIDAR/LU/Lakes)
n = 72        #Sample size n1=n2=n
alpha = 0.1   #(1-0.1)= 0.9 means 90% Confidence interval

#Determine degree of freedom
#This equation is simplified as sample size n1 and n2 are same

V = ((s1**2+s2**2)**2)/((s1**4+s2**4)/(n-1))

print('Degree of freedom, V=',V)

#Determine the critical value of the t-distribution
#From t-distribution table (en.wikipedia.org/wiki/Student's_t-distribution), for alpha/2
    which is 0.05 (as alpha = 0.1) and calculated V

t_distribution = 1.645

#Determine the minimum and maximum value

MinValue = (x1-x2) - (t_distribution*np.sqrt((s1**2+s2**2)/n))
MaxValue = (x1-x2) + (t_distribution*np.sqrt((s1**2+s2**2)/n))

print('Minimum value = ', MinValue)
print('Maximum value = ', MaxValue)

```

## D.3 Code to Plot and Show the Comparison between Observations and WRF

### D.3.1 WRF Plotter to Observe the Hourly Variations of Meteorological Properties

```

import numpy as np
import matplotlib.pyplot as plt
#plt.rcParams.update({'figure.max_open_warning': 0})

#Define text and font

plt.rc('text', usetex=True)
plt.rc('font', family='Times New Roman', size='16')

```

```

#Correction of deltaPhi
CorrDeltaPhi=14

#TANAB
#Define file name

filename1 =" StatisticalAnalysis /WRFCompareToTANAB_59_60_61.txt"

#Load all data in a matrix

data = np.loadtxt(filename1)
Time_GTOPO=data[:,0]
S50TANAB_GTOPO=data[:,2]
S50WRF_GTOPO=data[:,5]
SMeanFB_GTOPO=data[:,7]
SMeanNMSE_GTOPO=data[:,8]
dThetadz50TANAB_GTOPO=data[:,10]
dThetadz50WRF_GTOPO=data[:,13]
dThetadzFB_GTOPO=data[:,15]
dThetadzNMSE_GTOPO=data[:,16]

filename2 =" StatisticalAnalysis /WRFCompareToTANAB_62_63_64.txt"

data = np.loadtxt(filename2)
Time_SRTM=data[:,0]
S50TANAB_SRTM=data[:,2]
S50WRF_SRTM=data[:,5]
SMeanFB_SRTM=data[:,7]
SMeanNMSE_SRTM=data[:,8]
dThetadz50TANAB_SRTM=data[:,10]
dThetadz50WRF_SRTM=data[:,13]
dThetadzFB_SRTM=data[:,15]
dThetadzNMSE_SRTM=data[:,16]

filename3 =" StatisticalAnalysis /WRFCompareToTANAB_65_66_67.txt"

data = np.loadtxt(filename3)
Time_Land=data[:,0]
S50TANAB_Land=data[:,2]
S50WRF_Land=data[:,5]
SMeanFB_Land=data[:,7]
SMeanNMSE_Land=data[:,8]
dThetadz50TANAB_Land=data[:,10]
dThetadz50WRF_Land=data[:,13]
dThetadzFB_Land=data[:,15]
dThetadzNMSE_Land=data[:,16]

filename4 =" StatisticalAnalysis /WRFCompareToTANAB_68_69_70.txt"

data = np.loadtxt(filename4)

```

```

Time_LandLake=data[:,0]
S50TANAB_LandLake=data[:,2]
S50WRF_LandLake=data[:,5]
SMeanFB_LandLake=data[:,7]
SMeanNMSE_LandLake=data[:,8]
dThetadz50TANAB_LandLake=data[:,10]
dThetadz50WRF_LandLake=data[:,13]
dThetadzFB_LandLake=data[:,15]
dThetadzNMSE_LandLake=data[:,16]

fig = plt.figure()
plt.plot(Time.GTOPO,S50TANAB.GTOPO,color='darkblue',marker='o',markersize=7, markerfacecolor=
        'none',linewidth=0.6,label='Experimental_(TANAB)')
plt.plot(Time.GTOPO,S50WRF.GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none'
        ,linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,S50WRF.SRTM,color='green',marker='^',markersize=7, markerfacecolor='none'
        ,linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,S50WRF_Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='
        none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,S50WRF_LandLake,color='m',marker='>',markersize=7, markerfacecolor='none'
        ,linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (TANAB)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(TANAB).png',dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,dThetadz50TANAB.GTOPO,color='darkblue',marker='o',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='Experimental_(TANAB)')
plt.plot(Time.GTOPO,dThetadz50WRF.GTOPO,color='red',marker='s',markersize=7, markerfacecolor=
        'none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,dThetadz50WRF.SRTM,color='green',marker='^',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,dThetadz50WRF_Land,color='goldenrod',marker='<',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,dThetadz50WRF_LandLake,color='m',marker='>',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Temperature_Gradient_[K_m$^{-1}$]',fontsize=18)
#plt.title('Effect of Topography on Temperature Gradient (TANAB)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()

```



```

plt.savefig('Plots/Effect_of_Topography_Temperature_Gradient_(TANAB).png',dpi=800)
fig.show()

#UofASonic1
#Define file name for UofAUltraSonic1

filename5 ="StatisticalAnalysis/WRFCompareToUofASonic1_59_60_61.txt"

#Load all data in a matrix

data = np.loadtxt(filename5)
Time_GTOPO=data[:,0]
S50UofASonic1_GTOPO=data[:,2]
S50WRF_GTOPO=data[:,5]
SMeanFB_GTOPO=data[:,7]
SMeanNMSE_GTOPO=data[:,8]
SHF50UofASonic1_GTOPO=data[:,26]
SHF50WRF_GTOPO=data[:,29]
SHF50UofASonic1FB_GTOPO=data[:,31]
SHF50UofASonic1NMSE_GTOPO=data[:,32]
deltaPhi50SurfacelevelWRF_GTOPO=data[:,34]-CorrDeltaPhi

filename6 ="StatisticalAnalysis/WRFCompareToUofASonic1_62_63_64.txt"

data = np.loadtxt(filename6)
Time_SRTM=data[:,0]
S50UofASonic1_SRTM=data[:,2]
S50WRF_SRTM=data[:,5]
SMeanFB_SRTM=data[:,7]
SMeanNMSE_SRTM=data[:,8]
SHF50UofASonic1_SRTM=data[:,26]
SHF50WRF_SRTM=data[:,29]
SHF50UofASonic1FB_SRTM=data[:,31]
SHF50UofASonic1NMSE_SRTM=data[:,32]
deltaPhi50SurfacelevelWRF_SRTM=data[:,34]-CorrDeltaPhi

filename7 ="StatisticalAnalysis/WRFCompareToUofASonic1_65_66_67.txt"

data = np.loadtxt(filename7)
Time_Land=data[:,0]
S50UofASonic1_Land=data[:,2]
S50WRF_Land=data[:,5]
SMeanFB_Land=data[:,7]
SMeanNMSE_Land=data[:,8]
SHF50UofASonic1_Land=data[:,26]
SHF50WRF_Land=data[:,29]
SHF50UofASonic1FB_Land=data[:,31]
SHF50UofASonic1NMSE_Land=data[:,32]
deltaPhi50SurfacelevelWRF_Land=data[:,34]-CorrDeltaPhi

filename8="StatisticalAnalysis/WRFCompareToUofASonic1_68_69_70.txt"

```

```

data = np.loadtxt(filename8)
Time_LandLake=data[:,0]
S50UofASonic1_LandLake=data[:,2]
S50WRF_LandLake=data[:,5]
SMeanFB_LandLake=data[:,7]
SMeanNMSE_LandLake=data[:,8]
SHF50UofASonic1_LandLake=data[:,26]
SHF50WRF_LandLake=data[:,29]
SHF50UofASonic1FB_LandLake=data[:,31]
SHF50UofASonic1NMSE_LandLake=data[:,32]
deltaPhi50SurfacelevelWRF_LandLake=data[:,34]-CorrDeltaPhi

fig = plt.figure()
plt.plot(Time.GTOPO,S50UofASonic1.GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(3D_Ultrasonic1)')
plt.plot(Time.GTOPO,S50WRF.GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,S50WRF.SRTM,color='green',marker='^',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,S50WRF_Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,S50WRF_LandLake,color='m',marker='>',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Horizontal_Wind_Velocity_[m\s$^{-1}$]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (3D Ultrasonic1)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(3D_Ultrasonic1).png',
           dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.GTOPO,color='red',marker='s',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.SRTM,color='green',marker='^',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_Land,color='goldenrod',marker='<',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_LandLake,color='m',marker='>',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Bias_of_Wind_Direction_\Delta\phi$[_{o}$]',fontsize=18)
#plt.title('Effect of Topography on Wind Direction (3D Ultrasonic1)',fontsize=18)
plt.legend(fontsize='10.5')

```

```

fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(3D_Ultrasonic1).png',dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,SHF50UofASonic1.GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(3D_Ultrasonic1)')
plt.plot(Time.GTOPO,SHF50WRF.GTOPO,color='red',marker='s',markersize=7, markerfacecolor='
         none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,SHF50WRF.SRTM,color='green',marker='^',markersize=7, markerfacecolor='
         none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,SHF50WRF.Land,color='goldenrod',marker='<',markersize=7, markerfacecolor=
         'none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,SHF50WRF.LandLake,color='m',marker='>',markersize=7, markerfacecolor='
         none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lake)')

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5,23.5)
plt.ylim(-70,599)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Sensible_Heat_Flux_[W_m$^{-2}$]',fontsize=18)
#plt.title('Effect of Topography on Sensible Heat Flux (3D Ultrasonic1)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Sensible_Heat_Flux_(3D_Ultrasonic1).png',dpi=800)
fig.show()

#UofASonic2

filename9 ="StatisticalAnalysis/WRFCompareToUofASonic2_59_60_61.txt"

#Load all data in a matrix

data = np.loadtxt(filename9)
Time.GTOPO=data[:,0]
S50UofASonic2.GTOPO=data[:,2]
S50WRF.GTOPO=data[:,5]
SMeanFB.GTOPO=data[:,7]
SMeanNMSE.GTOPO=data[:,8]
SHF50UofASonic2.GTOPO=data[:,26]
SHF50WRF.GTOPO=data[:,29]
SHF50UofASonic2FB.GTOPO=data[:,31]
SHF50UofASonic2NMSE.GTOPO=data[:,32]
deltaPhi50SurfacelevelWRF.GTOPO=data[:,34]-CorrDeltaPhi

filename10 ="StatisticalAnalysis/WRFCompareToUofASonic2_62_63_64.txt"

data = np.loadtxt(filename10)
Time.SRTM=data[:,0]

```

```

S50UofASonic2_SRTM=data[:,2]
S50WRF_SRTM=data[:,5]
SMeanFB_SRTM=data[:,7]
SMeanNMSE_SRTM=data[:,8]
SHF50UofASonic2_SRTM=data[:,26]
SHF50WRF_SRTM=data[:,29]
SHF50UofASonic2FB_SRTM=data[:,31]
SHF50UofASonic2NMSE_SRTM=data[:,32]
deltaPhi50SurfacelevelWRF_SRTM=data[:,34]-CorrDeltaPhi

filename11="StatisticalAnalysis/WRFCompareToUofASonic2.65.66.67.txt"

data=np.loadtxt(filename11)
Time_Land=data[:,0]
S50UofASonic2_Land=data[:,2]
S50WRF_Land=data[:,5]
SMeanFB_Land=data[:,7]
SMeanNMSE_Land=data[:,8]
SHF50UofASonic2_Land=data[:,26]
SHF50WRF_Land=data[:,29]
SHF50UofASonic2FB_Land=data[:,31]
SHF50UofASonic2NMSE_Land=data[:,32]
deltaPhi50SurfacelevelWRF_Land=data[:,34]-CorrDeltaPhi

filename12="StatisticalAnalysis/WRFCompareToUofASonic2.68.69.70.txt"

data=np.loadtxt(filename12)
Time_LandLake=data[:,0]
S50UofASonic2_LandLake=data[:,2]
S50WRF_LandLake=data[:,5]
SMeanFB_LandLake=data[:,7]
SMeanNMSE_LandLake=data[:,8]
SHF50UofASonic2_LandLake=data[:,26]
SHF50WRF_LandLake=data[:,29]
SHF50UofASonic2FB_LandLake=data[:,31]
SHF50UofASonic2NMSE_LandLake=data[:,32]
deltaPhi50SurfacelevelWRF_LandLake=data[:,34]-CorrDeltaPhi

fig=plt.figure()
plt.plot(Time_GTOPO,S50UofASonic2_GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(3D_Ultrasonic2)')
plt.plot(Time_GTOPO,S50WRF_GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time_GTOPO,S50WRF_SRTM,color='green',marker='^',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time_GTOPO,S50WRF_Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time_GTOPO,S50WRF_LandLake,color='m',marker='>',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)

```

```

plt.ylabel('Horizontal_Wind_Velocity_[$m\ s^{-1}$]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (3D Ultrasonic2)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(3D_Ultrasonic2).png',
            dpi=800)
fig.show()

```

```

fig = plt.figure()
plt.plot(Time.GTOPO, deltaPhi50SurfacelevelWRF_GTOPO, color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(Time.GTOPO, deltaPhi50SurfacelevelWRF_SRTM, color='green', marker='^', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO, deltaPhi50SurfacelevelWRF_Land, color='goldenrod', marker='<', markersize=
         7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO, deltaPhi50SurfacelevelWRF_LandLake, color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

```

```

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5, 23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Bias_of_Wind_Direction_[$\Delta\ \phi\ [^{\circ}]$]', fontsize=18)
#plt.title('Effect of Topography on Wind Direction (3D Ultrasonic2)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction_(3D_Ultrasonic2).png', dpi=800)
fig.show()

```

```

fig = plt.figure()
plt.plot(Time.GTOPO, SHF50UofASonic2.GTOPO, color='darkblue', marker='o', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Experimental_(3D_Ultrasonic2)')
plt.plot(Time.GTOPO, SHF50WRF.GTOPO, color='red', marker='s', markersize=7, markerfacecolor='
         none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(Time.GTOPO, SHF50WRF.SRTM, color='green', marker='^', markersize=7, markerfacecolor='
         none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO, SHF50WRF.Land, color='goldenrod', marker='<', markersize=7, markerfacecolor=
         'none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO, SHF50WRF.LandLake, color='m', marker='>', markersize=7, markerfacecolor='
         none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lake)')

```

```

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5, 23.5)
plt.ylim(-70, 599)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Sensible_Heat_Flux_[$W\ m^{-2}$]', fontsize=18)
#plt.title('Effect of Topography on Sensible Heat Flux (3D Ultrasonic2)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Sensible_Heat_Flux_(3D_Ultrasonic2).png', dpi=800)
fig.show()

```

```
#RWDISonic1
```

```
filename13 =" StatisticalAnalysis/WRFCompareToRWDISonic1_59_60_61.txt"
```

```
#Load all data in a matrix
```

```
data = np.loadtxt(filename13)
Time_GTOPO=data[:,0]
S50RWDISonic1_GTOPO=data[:,2]
S50WRF_GTOPO=data[:,5]
SMeanFB_GTOPO=data[:,7]
SMeanNMSE_GTOPO=data[:,8]
deltaPhi50SurfacelevelWRF_GTOPO=data[:,34] - CorrDeltaPhi
```

```
filename14 =" StatisticalAnalysis/WRFCompareToRWDISonic1_62_63_64.txt"
```

```
data = np.loadtxt(filename14)
Time_SRTM=data[:,0]
S50RWDISonic1_SRTM=data[:,2]
S50WRF_SRTM=data[:,5]
SMeanFB_SRTM=data[:,7]
SMeanNMSE_SRTM=data[:,8]
deltaPhi50SurfacelevelWRF_SRTM=data[:,34] - CorrDeltaPhi
```

```
filename15 =" StatisticalAnalysis/WRFCompareToRWDISonic1_65_66_67.txt"
```

```
data = np.loadtxt(filename15)
Time_Land=data[:,0]
S50RWDISonic1_Land=data[:,2]
S50WRF_Land=data[:,5]
SMeanFB_Land=data[:,7]
SMeanNMSE_Land=data[:,8]
deltaPhi50SurfacelevelWRF_Land=data[:,34] - CorrDeltaPhi
```

```
filename16=" StatisticalAnalysis/WRFCompareToRWDISonic1_68_69_70.txt"
```

```
data = np.loadtxt(filename16)
Time_LandLake=data[:,0]
S50RWDISonic1_LandLake=data[:,2]
S50WRF_LandLake=data[:,5]
SMeanFB_LandLake=data[:,7]
SMeanNMSE_LandLake=data[:,8]
deltaPhi50SurfacelevelWRF_LandLake=data[:,34] - CorrDeltaPhi
```

```
fig = plt.figure()
```

```
plt.plot(Time_GTOPO,S50RWDISonic1_GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(2D_Ultrasonic1)')
```

```
plt.plot(Time_GTOPO,S50WRF_GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
```

```

plt.plot(Time.GTOPO,S50WRF.SRTM,color='green',marker='^',markersize=7, markerfacecolor='none',
linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,S50WRF.Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='none',
linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,S50WRF.LandLake,color='m',marker='>',markersize=7, markerfacecolor='none',
linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (2D Ultrasonic1)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(2D_Ultrasonic1).png',
dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.GTOPO,color='red',marker='s',markersize=7,
markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.SRTM,color='green',marker='^',markersize=7,
markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.Land,color='goldenrod',marker='<',markersize=7,
markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.LandLake,color='m',marker='>',markersize=7,
markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Bias_of_Wind_Direction_$\Delta\phi$_[°]',fontsize=18)
#plt.title('Effect of Topography on Wind Direction (2D Ultrasonic1)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(2D_Ultrasonic1).png',dpi=800)
fig.show()

#RWDISonic2

filename17 ="StatisticalAnalysis/WRFCompareToRWDISonic2_59_60_61.txt"

#Load all data in a matrix

data = np.loadtxt(filename17)
Time.GTOPO=data[:,0]
S50RWDISonic2.GTOPO=data[:,2]
S50WRF.GTOPO=data[:,5]
SMeanFB.GTOPO=data[:,7]
SMeanNMSE.GTOPO=data[:,8]
deltaPhi50SurfacelevelWRF.GTOPO=data[:,34]-CorrDeltaPhi

```

```

filename18 =" StatisticalAnalysis/WRFCompareToRWDISonic2.62.63.64.txt"

data = np.loadtxt( filename18)
Time_SRTM=data [:,0]
S50RWDISonic2_SRTM=data [:,2]
S50WRF_SRTM=data [:,5]
SMeanFB_SRTM=data [:,7]
SMeanNMSE_SRTM=data [:,8]
deltaPhi50SurfacelevelWRF_SRTM=data[:,34] - CorrDeltaPhi

filename19 =" StatisticalAnalysis/WRFCompareToRWDISonic2.65.66.67.txt"

data = np.loadtxt( filename19)
Time_Land=data [:,0]
S50RWDISonic2_Land=data [:,2]
S50WRF_Land=data [:,5]
SMeanFB_Land=data [:,7]
SMeanNMSE_Land=data [:,8]
deltaPhi50SurfacelevelWRF_Land=data[:,34] - CorrDeltaPhi

filename20=" StatisticalAnalysis/WRFCompareToRWDISonic2.68.69.70.txt"

data = np.loadtxt( filename20)
Time_LandLake=data [:,0]
S50RWDISonic2_LandLake=data [:,2]
S50WRF_LandLake=data [:,5]
SMeanFB_LandLake=data [:,7]
SMeanNMSE_LandLake=data [:,8]
deltaPhi50SurfacelevelWRF_LandLake=data[:,34] - CorrDeltaPhi

fig = plt.figure()
plt.plot(Time.GTOPO,S50RWDISonic2_GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(2D_Ultrasonic2)')
plt.plot(Time.GTOPO,S50WRF_GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,S50WRF_SRTM,color='green',marker='^',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,S50WRF_Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,S50WRF_LandLake,color='m',marker='>',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (2D Ultrasonic2)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(2D_Ultrasonic2).png',
           dpi=800)

```



```

fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_GTOPO , color='red',marker='s',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_SRTM , color='green',marker='^',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_Land , color='goldenrod',marker='<',markersize=
         =7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_LandLake , color='m',marker='>',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Bias_of_Wind_Direction_\Delta_phi$[{}$]', fontsize=18)
#plt.title('Effect of Topography on Wind Direction (2D Ultrasonic2)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(2D_Ultrasonic2).png',dpi=800)
fig.show()

#RWDISonic3

filename21 ="StatisticalAnalysis/WRFCompareToRWDISonic3.59.60.61.txt"

#Load all data in a matrix

data = np.loadtxt(filename21)
Time_GTOPO=data[:,0]
S50RWDISonic3_GTOPO=data[:,2]
S50WRF_GTOPO=data[:,5]
SMeanFB_GTOPO=data[:,7]
SMeanNMSE_GTOPO=data[:,8]
deltaPhi50SurfacelevelWRF_GTOPO=data[:,34]-CorrDeltaPhi

filename22 ="StatisticalAnalysis/WRFCompareToRWDISonic3.62.63.64.txt"

data = np.loadtxt(filename22)
Time_SRTM=data[:,0]
S50RWDISonic3_SRTM=data[:,2]
S50WRF_SRTM=data[:,5]
SMeanFB_SRTM=data[:,7]
SMeanNMSE_SRTM=data[:,8]
deltaPhi50SurfacelevelWRF_SRTM=data[:,34]-CorrDeltaPhi

filename23 ="StatisticalAnalysis/WRFCompareToRWDISonic3.65.66.67.txt"

data = np.loadtxt(filename23)

```

```

Time_Land=data[:,0]
S50RWDISonic3_Land=data[:,2]
S50WRF_Land=data[:,5]
SMeanFB_Land=data[:,7]
SMeanNMSE_Land=data[:,8]
deltaPhi50SurfacelevelWRF_Land=data[:,34]-CorrDeltaPhi

filename24="StatisticalAnalysis/WRFCompareToRWDISonic3_68_69_70.txt"

data = np.loadtxt(filename24)
Time_LandLake=data[:,0]
S50RWDISonic3_LandLake=data[:,2]
S50WRF_LandLake=data[:,5]
SMeanFB_LandLake=data[:,7]
SMeanNMSE_LandLake=data[:,8]
deltaPhi50SurfacelevelWRF_LandLake=data[:,34]-CorrDeltaPhi

fig = plt.figure()
plt.plot(Time_GTOPO,S50RWDISonic3_GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(2D_Ultrasonic3)')
plt.plot(Time_GTOPO,S50WRF_GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time_GTOPO,S50WRF_SRTM,color='green',marker='^',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time_GTOPO,S50WRF_Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time_GTOPO,S50WRF_LandLake,color='m',marker='>',markersize=7, markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (2D Ultrasonic3)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(2D_Ultrasonic3).png',
          dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time_GTOPO,deltaPhi50SurfacelevelWRF_GTOPO,color='red',marker='s',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time_GTOPO,deltaPhi50SurfacelevelWRF_SRTM,color='green',marker='^',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time_GTOPO,deltaPhi50SurfacelevelWRF_Land,color='goldenrod',marker='<',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time_GTOPO,deltaPhi50SurfacelevelWRF_LandLake,color='m',marker='>',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.axhline(0, color='black', lw=0.1)
plt.xlim(-0.5,23.5)

```

```

plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Bias_of_Wind_Direction_\Delta\phi$_{\phi}$', fontsize=18)
#plt.title('Effect of Topography on Wind Direction (2D Ultrasonic3)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(2D_Ultrasonic3).png', dpi=800)
fig.show()

```

```

#Mine Trailer

```

```

filename25 ="StatisticalAnalysis/WRFCompareToMineTrailer_59.60.61.txt"

```

```

data = np.loadtxt(filename25)
Time_GTOPO=data[:,0]
T50MineTrailer_GTOPO=data[:,10]
T50WRF_GTOPO=data[:,13]
TFB_GTOPO=data[:,15]
TNMSE_GTOPO=data[:,16]
RH50MineTrailer_GTOPO=data[:,18]
RH50WRF_GTOPO=data[:,21]
RHFB_GTOPO=data[:,23]
RHNMSE_GTOPO=data[:,24]

```

```

filename26 ="StatisticalAnalysis/WRFCompareToMineTrailer_62.63.64.txt"

```

```

data = np.loadtxt(filename26)
Time_SRTM=data[:,0]
T50MineTrailer_SRTM=data[:,10]
T50WRF_SRTM=data[:,13]
TFB_SRTM=data[:,15]
TNMSE_SRTM=data[:,16]
RH50MineTrailer_SRTM=data[:,18]
RH50WRF_SRTM=data[:,21]
RHFB_SRTM=data[:,23]
RHNMSE_SRTM=data[:,24]

```

```

filename27 ="StatisticalAnalysis/WRFCompareToMineTrailer_65.66.67.txt"

```

```

data = np.loadtxt(filename27)
Time_Land=data[:,0]
T50MineTrailer_Land=data[:,10]
T50WRF_Land=data[:,13]
TFB_Land=data[:,15]
TNMSE_Land=data[:,16]
RH50MineTrailer_Land=data[:,18]
RH50WRF_Land=data[:,21]
RHFB_Land=data[:,23]
RHNMSE_Land=data[:,24]

```

```

filename28 ="StatisticalAnalysis/WRFCompareToMineTrailer_68.69.70.txt"

```

```

data = np.loadtxt(filename28)
Time_LandLake=data[:,0]
T50MineTrailer_LandLake=data[:,10]
T50WRF_LandLake=data[:,13]
TFB_LandLake=data[:,15]
TNMSE_LandLake=data[:,16]
RH50MineTrailer_LandLake=data[:,18]
RH50WRF_LandLake=data[:,21]
RHFB_LandLake=data[:,23]
RHNMSE_LandLake=data[:,24]

fig = plt.figure()
plt.plot(Time.GTOPO, T50MineTrailer.GTOPO, color='darkblue', marker='o', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Experimental_(Weather_Station)')
plt.plot(Time.GTOPO, T50WRF.GTOPO, color='red', marker='s', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(Time.GTOPO, T50WRF.SRTM, color='green', marker='^', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO, T50WRF.Land, color='goldenrod', marker='<', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO, T50WRF.LandLake, color='m', marker='>', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5, 23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Temperature_[K]', fontsize=18)
#plt.title('Effect of Topography on Temperature (Weather Station)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Temperature_(Weather_Station).png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO, RH50MineTrailer.GTOPO, color='darkblue', marker='o', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Experimental_(Weather_Station)')
plt.plot(Time.GTOPO, RH50WRF.GTOPO, color='red', marker='s', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(Time.GTOPO, RH50WRF.SRTM, color='green', marker='^', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO, RH50WRF.Land, color='goldenrod', marker='<', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO, RH50WRF.LandLake, color='m', marker='>', markersize=7, markerfacecolor='none',
         linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5, 23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Relative_Humidity_[\%]', fontsize=18)
#plt.title('Effect of Topography on Relative Humidity(Weather Station)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Relative_Humidity_(Weather_Station).png', dpi=800)
fig.show()

```

```

#WBEA Cup Anemometer
filename29 =" StatisticalAnalysis/WRFCompareToWBEACup_59.60.61.txt"

#Load all data in a matrix

data = np.loadtxt(filename29)
Time_GTOPO=data[:,0]
S50WBEACupAnemometer_GTOPO=data[:,2]
S50WRF_GTOPO=data[:,5]
SMeanFB_GTOPO=data[:,7]
SMeanNMSE_GTOPO=data[:,8]
T50WBEACupAnemometer_GTOPO=data[:,10]
T50WRF_GTOPO=data[:,13]
TFB_GTOPO=data[:,15]
TNMSE_GTOPO=data[:,16]
RH50WBEACupAnemometer_GTOPO=data[:,18]
RH50WRF_GTOPO=data[:,21]
RHFB_GTOPO=data[:,23]
RHNMSE_GTOPO=data[:,24]
deltaPhi50SurfacelevelWRF_GTOPO=data[:,34]-CorrDeltaPhi


filename30 =" StatisticalAnalysis/WRFCompareToWBEACup_62.63.64.txt"

data = np.loadtxt(filename30)
Time_SRTM=data[:,0]
S50WBEACupAnemometer_SRTM=data[:,2]
S50WRF_SRTM=data[:,5]
SMeanFB_SRTM=data[:,7]
SMeanNMSE_SRTM=data[:,8]
T50WBEACupAnemometer_SRTM=data[:,10]
T50WRF_SRTM=data[:,13]
TFB_SRTM=data[:,15]
TNMSE_SRTM=data[:,16]
RH50WBEACupAnemometer_SRTM=data[:,18]
RH50WRF_SRTM=data[:,21]
RHFB_SRTM=data[:,23]
RHNMSE_SRTM=data[:,24]
deltaPhi50SurfacelevelWRF_SRTM=data[:,34]-CorrDeltaPhi


filename31 =" StatisticalAnalysis/WRFCompareToWBEACup_65.66.67.txt"

data = np.loadtxt(filename31)
Time_Land=data[:,0]
S50WBEACupAnemometer_Land=data[:,2]
S50WRF_Land=data[:,5]
SMeanFB_Land=data[:,7]
SMeanNMSE_Land=data[:,8]
T50WBEACupAnemometer_Land=data[:,10]
T50WRF_Land=data[:,13]
TFB_Land=data[:,15]

```

```

TNMSE_Land=data[:,16]
RH50WBEACupAnemometer_Land=data[:,18]
RH50WRF_Land=data[:,21]
RHFB_Land=data[:,23]
RHNMSE_Land=data[:,24]
deltaPhi50SurfacelevelWRF_Land=data[:,34]-CorrDeltaPhi

filename32="StatisticalAnalysis/WRFCompareToWBEACup_68.69.70.txt"

data = np.loadtxt(filename32)
Time_LandLake=data[:,0]
S50WBEACupAnemometer_LandLake=data[:,2]
S50WRF_LandLake=data[:,5]
SMeanFB_LandLake=data[:,7]
SMeanNMSE_LandLake=data[:,8]
T50WBEACupAnemometer_LandLake=data[:,10]
T50WRF_LandLake=data[:,13]
TFB_LandLake=data[:,15]
TNMSE_LandLake=data[:,16]
RH50WBEACupAnemometer_LandLake=data[:,18]
RH50WRF_LandLake=data[:,21]
RHFB_LandLake=data[:,23]
RHNMSE_LandLake=data[:,24]
deltaPhi50SurfacelevelWRF_LandLake=data[:,34]-CorrDeltaPhi

fig = plt.figure()
plt.plot(Time.GTOPO,S50WBEACupAnemometer.GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(WBEA_Station)')
plt.plot(Time.GTOPO,S50WRF.GTOPO,color='red',marker='s',markersize=7, markerfacecolor='none'
         ,linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,S50WRF.SRTM,color='green',marker='^',markersize=7, markerfacecolor='none'
         ,linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,S50WRF_Land,color='goldenrod',marker='<',markersize=7, markerfacecolor='none'
         ,linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,S50WRF_LandLake,color='m',marker='>',markersize=7, markerfacecolor='none'
         ,linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (WBEA Station)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(WBEA_Station).png',dpi
          =800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF.GTOPO,color='red',marker='s',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')

```

```

plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_SRTM,color='green',marker='^',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_Land,color='goldenrod',marker='<',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,deltaPhi50SurfacelevelWRF_LandLake,color='m',marker='>',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.axhline(0,color='black',lw=0.1)
plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Bias_of_Wind_Direction_\Delta_phi$[^\circ]$',fontsize=18)
#plt.title('Effect of Topography on Wind Direction (WBEA Station)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(WBEA_Station).png',dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,T50WBEACupAnemometer.GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(WBEA_Station)')
plt.plot(Time.GTOPO,T50WRF.GTOPO,color='red',marker='s',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,T50WRF.SRTM,color='green',marker='^',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,T50WRF.Land,color='goldenrod',marker='<',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,T50WRF.LandLake,color='m',marker='>',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)
plt.xlabel('Mountain_Standard_Time_(MST)_[hr]',fontsize=18)
plt.ylabel('Temperature_[K]',fontsize=18)
#plt.title('Effect of Topography on Temperature (WBEA Station)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Temperature_(WBEA_Station).png',dpi=800)
fig.show()

fig = plt.figure()
plt.plot(Time.GTOPO,RH50WBEACupAnemometer.GTOPO,color='darkblue',marker='o',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Experimental_(WBEA_Station)')
plt.plot(Time.GTOPO,RH50WRF.GTOPO,color='red',marker='s',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(Time.GTOPO,RH50WRF.SRTM,color='green',marker='^',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(Time.GTOPO,RH50WRF.Land,color='goldenrod',marker='<',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(Time.GTOPO,RH50WRF.LandLake,color='m',marker='>',markersize=7,markerfacecolor='none',
         linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim(-0.5,23.5)

```

```

plt.xlabel('Mountain_Standard_Time_(MST)_[hr]', fontsize=18)
plt.ylabel('Relative_Humidity_[\%]', fontsize=18)
#plt.title('Effect of Topography on Relative Humidity(WBEA Station)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Relative_Humidity_(WBEA_Station).png', dpi=800)
fig.show()

```

```
#SODAR
```

```
filename33 ="StatisticalAnalysis/WRFCompareToSODAR_59_60_61.txt"
```

```
#Load all data in a matrix
```

```

data = np.loadtxt(filename33)

Time_GTOPO=data[:,0]
Altitude_GTOPO=data[:,2]
S50SODAR_GTOPO=data[:,3]
S50WRF_GTOPO=data[:,6]
SMeanFB_GTOPO=data[:,9]
SMeanNMSE_GTOPO=data[:,10]
W50SODAR_GTOPO=data[:,11]
W50WRF_GTOPO=data[:,14]
WMeanFB_GTOPO=data[:,17]
WMeanNMSE_GTOPO=data[:,18]
deltaPhi50SODARWRF_GTOPO=data[:,19]-CorrDeltaPhi

```

```
filename34 ="StatisticalAnalysis/WRFCompareToSODAR_62_63_64.txt"
```

```

data = np.loadtxt(filename34)

Time_SRTM=data[:,0]
Altitude_SRTM=data[:,2]
S50SODAR_SRTM=data[:,3]
S50WRF_SRTM=data[:,6]
SMeanFB_SRTM=data[:,9]
SMeanNMSE_SRTM=data[:,10]
W50SODAR_SRTM=data[:,11]
W50WRF_SRTM=data[:,14]
WMeanFB_SRTM=data[:,17]
WMeanNMSE_SRTM=data[:,18]
deltaPhi50SODARWRF_SRTM=data[:,19]-CorrDeltaPhi

```

```
filename35 ="StatisticalAnalysis/WRFCompareToSODAR_65_66_67.txt"
```

```

data = np.loadtxt(filename35)
Time_Land=data[:,0]
Altitude_Land=data[:,2]
S50SODAR_Land=data[:,3]

```



```

S50WRF_Land=data[:,6]
SMeanFB_Land=data[:,9]
SMeanNMSE_Land=data[:,10]
W50SODAR_Land=data[:,11]
W50WRF_Land=data[:,14]
WMeanFB_Land=data[:,17]
WMeanNMSE_Land=data[:,18]
deltaPhi50SODARWRF_Land=data[:,19]-CorrDeltaPhi

filename36="StatisticalAnalysis/WRFCompareToSODAR_68_69_70.txt"

data = np.loadtxt(filename36)
Time_LandLake=data[:,0]
Altitude_LandLake=data[:,2]
S50SODAR_LandLake=data[:,3]
S50WRF_LandLake=data[:,6]
SMeanFB_LandLake=data[:,9]
SMeanNMSE_LandLake=data[:,10]
W50SODAR_LandLake=data[:,11]
W50WRF_LandLake=data[:,14]
WMeanFB_LandLake=data[:,17]
WMeanNMSE_LandLake=data[:,18]
deltaPhi50SODARWRF_LandLake=data[:,19]-CorrDeltaPhi

ind_al_04= [x for x in range(0,len(Altitude_GTOPO)) if (Time_GTOPO[x]==0)]

fig = plt.figure()
plt.plot(S50SODAR_GTOPO[ind_al_04],Altitude_GTOPO[ind_al_04],color='darkblue',marker='o',
         markersize=7, markerfacecolor='none',linewidth=0.6,label='Experimental_(SODAR)')
plt.plot(S50WRF_GTOPO[ind_al_04],Altitude_GTOPO[ind_al_04],color='red',marker='s',markersize
         =7, markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(S50WRF_SRTM[ind_al_04],Altitude_GTOPO[ind_al_04],color='green',marker='^',
         markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(S50WRF_Land[ind_al_04],Altitude_GTOPO[ind_al_04],color='goldenrod',marker='<',
         markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(S50WRF_LandLake[ind_al_04],Altitude_GTOPO[ind_al_04],color='m',marker='>',
         markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes
         )')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m.s$^{-1}$]',fontsize=18)
plt.ylabel('Altitude_[m]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(SODAR)_[0-4].png',dpi
           =800)
fig.show()

fig = plt.figure()
plt.plot(W50SODAR_GTOPO[ind_al_04],Altitude_GTOPO[ind_al_04],color='darkblue',marker='o',

```

```

    markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(W50WRF_GTOPO[ind_al_04], Altitude_GTOPO[ind_al_04], color='red', marker='s', markersize
        =7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(W50WRF_SRTM[ind_al_04], Altitude_GTOPO[ind_al_04], color='green', marker='^',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(W50WRF_Land[ind_al_04], Altitude_GTOPO[ind_al_04], color='goldenrod', marker='<',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(W50WRF_LandLake[ind_al_04], Altitude_GTOPO[ind_al_04], color='m', marker='>',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes
        )')

```

```

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
plt.axvline(0, color='black', lw=0.1)
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Vertical_Wind_Velocity_(SODAR)_[0-4].png', dpi
            =800)
fig.show()

```

```

fig = plt.figure()
plt.plot(deltaPhi50SODARWRF_GTOPO[ind_al_04], Altitude_GTOPO[ind_al_04], color='red', marker='s'
        , markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(deltaPhi50SODARWRF_SRTM[ind_al_04], Altitude_GTOPO[ind_al_04], color='green', marker='^'
        , markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(deltaPhi50SODARWRF_Land[ind_al_04], Altitude_GTOPO[ind_al_04], color='goldenrod',
        marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
        Land_Use)')
plt.plot(deltaPhi50SODARWRF_LandLake[ind_al_04], Altitude_GTOPO[ind_al_04], color='m', marker='>'
        , markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/
        Lakes)')

```

```

plt.xlim()
plt.xlabel('Bias_of_Wind_Direction_$(Delta\phi$)_[^{\circ}]$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction (SODAR)_[0-4].png', dpi=800)
fig.show()

```

```

ind_al_48= [x for x in range(0, len(Altitude_GTOPO)) if (Time_GTOPO[x]==4)]

```

```

fig = plt.figure()
plt.plot(S50SODAR_GTOPO[ind_al_48], Altitude_GTOPO[ind_al_48], color='darkblue', marker='o',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(S50WRF_GTOPO[ind_al_48], Altitude_GTOPO[ind_al_48], color='red', marker='s', markersize
        =7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')

```

```

plt.plot(S50WRF_SRTM[ind_al_48], Altitude_GTOPO[ind_al_48], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(S50WRF_Land[ind_al_48], Altitude_GTOPO[ind_al_48], color='goldenrod', marker='<',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(S50WRF_LandLake[ind_al_48], Altitude_GTOPO[ind_al_48], color='m', marker='>',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes
         )')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(SODAR)_[4-8].png', dpi
           =800)
fig.show()

fig = plt.figure()
plt.plot(W50SODAR_GTOPO[ind_al_48], Altitude_GTOPO[ind_al_48], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(W50WRF_GTOPO[ind_al_48], Altitude_GTOPO[ind_al_48], color='red', marker='s', markersize
         =7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(W50WRF_SRTM[ind_al_48], Altitude_GTOPO[ind_al_48], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(W50WRF_Land[ind_al_48], Altitude_GTOPO[ind_al_48], color='goldenrod', marker='<',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(W50WRF_LandLake[ind_al_48], Altitude_GTOPO[ind_al_48], color='m', marker='>',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes
         )')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
plt.axvline(0, color='black', lw=0.1)
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Vertical_Wind_Velocity_(SODAR)_[4-8].png', dpi
           =800)
fig.show()

fig = plt.figure()
plt.plot(deltaPhi50SODARWRF_GTOPO[ind_al_48], Altitude_GTOPO[ind_al_48], color='red', marker='s',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(deltaPhi50SODARWRF_SRTM[ind_al_48], Altitude_GTOPO[ind_al_48], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(deltaPhi50SODARWRF_Land[ind_al_48], Altitude_GTOPO[ind_al_48], color='goldenrod',
         marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
         Land_Use)')

```

```

plt.plot(deltaPhi50SODARWRF_LandLake[ind_al_48], Altitude_GTOPO[ind_al_48], color='m', marker='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Bias_of_Wind_Direction_\Delta\phi$[{}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(SODAR)_[4-8].png', dpi=800)
fig.show()

ind_al_812= [x for x in range(0, len(Altitude_GTOPO)) if (Time_GTOPO[x]==8)]

fig = plt.figure()
plt.plot(S50SODAR_GTOPO[ind_al_812], Altitude_GTOPO[ind_al_812], color='darkblue', marker='o', markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(S50WRF_GTOPO[ind_al_812], Altitude_GTOPO[ind_al_812], color='red', marker='s', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(S50WRF_SRTM[ind_al_812], Altitude_GTOPO[ind_al_812], color='green', marker='^', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(S50WRF_Land[ind_al_812], Altitude_GTOPO[ind_al_812], color='goldenrod', marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(S50WRF_LandLake[ind_al_812], Altitude_GTOPO[ind_al_812], color='m', marker='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m\ss{-1}]$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(SODAR)_[8-12].png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W50SODAR_GTOPO[ind_al_812], Altitude_GTOPO[ind_al_812], color='darkblue', marker='o', markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(W50WRF_GTOPO[ind_al_812], Altitude_GTOPO[ind_al_812], color='red', marker='s', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(W50WRF_SRTM[ind_al_812], Altitude_GTOPO[ind_al_812], color='green', marker='^', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(W50WRF_Land[ind_al_812], Altitude_GTOPO[ind_al_812], color='goldenrod', marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(W50WRF_LandLake[ind_al_812], Altitude_GTOPO[ind_al_812], color='m', marker='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

```

```

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
plt.axvline(0, color='black', lw=0.1)
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Vertical_Wind_Velocity_(SODAR)_[8-12].png', dpi
            =800)
fig.show()

fig = plt.figure()
plt.plot(deltaPhi50SODARWRF_GTOPO[ind_al_812], Altitude_GTOPO[ind_al_812], color='red', marker=
        's', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(deltaPhi50SODARWRF_SRTM[ind_al_812], Altitude_GTOPO[ind_al_812], color='green', marker
        ='^', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(deltaPhi50SODARWRF_Land[ind_al_812], Altitude_GTOPO[ind_al_812], color='goldenrod',
        marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
        Land_Use)')
plt.plot(deltaPhi50SODARWRF_LandLake[ind_al_812], Altitude_GTOPO[ind_al_812], color='m', marker
        ='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/
        Lakes)')

plt.xlim()
plt.xlabel('Bias_of_Wind_Direction_\Delta\phi$[_{o}]$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(SODAR)_[8-12].png', dpi=800)
fig.show()

ind_al_1216= [x for x in range(0, len(Altitude_GTOPO)) if (Time_GTOPO[x]==12)]

fig = plt.figure()
plt.plot(S50SODAR_GTOPO[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='darkblue', marker='o'
        , markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(S50WRF_GTOPO[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='red', marker='s',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(S50WRF_SRTM[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='green', marker='^',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(S50WRF_Land[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='goldenrod', marker='<',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(S50WRF_LandLake[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='m', marker='>',
        markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes
        )')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m s$^{-1}$]', fontsize=18)

```

```

plt.ylabel('Altitude[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(SODAR)_[12-16].png', dpi
           =800)
fig.show()

```

```

fig = plt.figure()
plt.plot(W50SODAR_GTOPO[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(W50WRF_GTOPO[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='red', marker='s',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(W50WRF_SRTM[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(W50WRF_Land[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='goldenrod', marker='<',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(W50WRF_LandLake[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='m', marker='>',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

```

```

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m.s$^{-1}]$', fontsize=18)
plt.ylabel('Altitude[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
plt.axvline(0, color='black', lw=0.1)
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Vertical_Wind_Velocity_(SODAR)_[12-16].png', dpi
           =800)
fig.show()

```

```

fig = plt.figure()
plt.plot(deltaPhi50SODARWRF_GTOPO[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='red',
         marker='s', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(deltaPhi50SODARWRF_SRTM[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='green',
         marker='^', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(deltaPhi50SODARWRF_Land[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='goldenrod',
         marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
         Land_Use)')
plt.plot(deltaPhi50SODARWRF_LandLake[ind_al_1216], Altitude_GTOPO[ind_al_1216], color='m',
         marker='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
         Land_Use/Lakes)')

```

```

plt.xlim()
plt.xlabel('Bias_of_Wind_Direction_$\Delta\phi$_[^{\circ}]$', fontsize=18)
plt.ylabel('Altitude[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()

```

```

plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(SODAR)_[12-16].png',dpi=800)
fig.show()

ind_al_1620= [x for x in range(0,len(Altitude_GTOPO)) if (Time_GTOPO[x]==16)]

fig = plt.figure()
plt.plot(S50SODAR_GTOPO[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='darkblue',marker='o',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Experimental_(SODAR)')
plt.plot(S50WRF_GTOPO[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='red',marker='s',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(S50WRF_SRTM[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='green',marker='^',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(S50WRF_Land[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='goldenrod',marker='<',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(S50WRF_LandLake[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='m',marker='>',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
plt.ylabel('Altitude_[m]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)',fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(SODAR)_[16-20].png',dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W50SODAR_GTOPO[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='darkblue',marker='o',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Experimental_(SODAR)')
plt.plot(W50WRF_GTOPO[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='red',marker='s',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(GTOPO)')
plt.plot(W50WRF_SRTM[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='green',marker='^',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR)')
plt.plot(W50WRF_Land[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='goldenrod',marker='<',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(W50WRF_LandLake[ind_al_1620],Altitude_GTOPO[ind_al_1620],color='m',marker='>',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
plt.ylabel('Altitude_[m]',fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)',fontsize=18)
plt.legend(fontsize='10.5')
plt.axvline(0, color='black', lw=0.1)
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Vertical_Wind_Velocity_(SODAR)_[16-20].png',dpi=800)

```

```

fig.show()

fig = plt.figure()
plt.plot(deltaPhi50SODARWRF_GTOPO[ind_al_1620], Altitude_GTOPO[ind_al_1620], color='red',
         marker='s', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(deltaPhi50SODARWRF_SRTM[ind_al_1620], Altitude_GTOPO[ind_al_1620], color='green',
         marker='^', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(deltaPhi50SODARWRF_Land[ind_al_1620], Altitude_GTOPO[ind_al_1620], color='goldenrod',
         marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
         Land_Use)')
plt.plot(deltaPhi50SODARWRF_LandLake[ind_al_1620], Altitude_GTOPO[ind_al_1620], color='m',
         marker='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/
         Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Bias_of_Wind_Direction_\\Delta\\phi$\\{^{{o}}\\}$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(SODAR)_[16-20].png', dpi=800)
fig.show()

ind_al_2024 = [x for x in range(0, len(Altitude_GTOPO)) if (Time_GTOPO[x]==20)]

fig = plt.figure()
plt.plot(S50SODAR_GTOPO[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(S50WRF_GTOPO[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='red', marker='s',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(S50WRF_SRTM[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(S50WRF_Land[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='goldenrod', marker='<',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(S50WRF_LandLake[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='m', marker='>',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m\\s$\\{^{-1}\\}\\}$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Horizontal_Wind_Velocity_(SODAR)_[20-24].png', dpi=800)
fig.show()

fig = plt.figure()

```



```

plt.plot(W50SODAR_GTOPO[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Experimental_(SODAR)')
plt.plot(W50WRF_GTOPO[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='red', marker='s',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(W50WRF_SRTM[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(W50WRF_Land[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='goldenrod', marker='<',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(W50WRF_LandLake[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='m', marker='>',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
plt.axvline(0, color='black', lw=0.1)
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Vertical_Wind_Velocity_(SODAR)_[20-24].png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(deltaPhi50SODARWRF_GTOPO[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='red',
         marker='s', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(GTOPO)')
plt.plot(deltaPhi50SODARWRF_SRTM[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='green',
         marker='^', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR)')
plt.plot(deltaPhi50SODARWRF_Land[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='goldenrod',
         marker='<', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use)')
plt.plot(deltaPhi50SODARWRF_LandLake[ind_al_2024], Altitude_GTOPO[ind_al_2024], color='m',
         marker='>', markersize=7, markerfacecolor='none', linewidth=0.6, label='WRF_(SRTM-LIDAR/Land_Use/Lakes)')

plt.xlim()
plt.xlabel('Bias_of_Wind_Direction_\\Delta\\phi$[_{o}]$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Vertical Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='10.5')
fig.tight_layout()
plt.savefig('Plots/Effect_of_Topography_on_Wind_Direction(SODAR)_[20-24].png', dpi=800)
fig.show()

plt.show()

```

## D.3.2 WRF Contour Plotter to Analyze Spatial Distribution

```

import random
import sys

```

```

import os
import matplotlib.pyplot as plt
import numpy as np

#Define text and font

plt.rc('text', usetex=True)
plt.rc('font', family='Times_New_Roman', size='16')

#Define a step for jumping over lat and lon when plotting normalized wind vectors
#The step must be a divisor for both Nx and Ny in WRF
#For example if Nx = Ny = 152 = 2 x 2 x 2 x 19
#The step can be 2, 4, 8, 19, 38, etc ...
latlonstep=8
dlat=0.004
dlon=0.006
latlonTol=0.002

filename ="Mine_68/wrfout_d05_2018-05-18_02-00-00.txt"
#filename ="Mine 68/wrfout_d05_2018-05-18_06-00-00.txt"
#filename ="Mine 68/wrfout_d05_2018-05-18_10-00-00.txt"
#filename ="Mine 68/wrfout_d05_2018-05-18_14-00-00.txt"
#filename ="Mine 68/wrfout_d05_2018-05-18_18-00-00.txt"
#filename ="Mine 68/wrfout_d05_2018-05-18_22-00-00.txt"

#Load all data in a matrix

data = np.loadtxt(filename)

lon=data[:,2]
lat=data[:,3]
height=data[:,4]
land=data[:,5]
ptemp=data[:,6]
smean=data[:,7]
skintemp=data[:,8]
u10=data[:,9]
v10=data[:,10]
RH=data[:,11]
HFX=data[:,12]

lonmin=np.min(lon)
lonmax=np.max(lon)
latmin=np.min(lat)
latmax=np.max(lat)

#Height from Sea Level

fig = plt.figure()
ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]

```

```

hz=plt.tricontourf(lon,lat,height,cmap=plt.get_cmap('jet'),levels=np.linspace(200,520,33))
plt.colorbar(hz,label='Surface_Height_from_Sea_Level(m)')
ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlabel('Distance_from_West_to_East[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North[km]',fontsize=18)
fig.tight_layout()

#plt.savefig('Plots/Surface Height from Sea Level (GTOPO).png',dpi=800)
#plt.savefig('Plots/Surface Height from Sea Level (SRTM).png',dpi=800)
#plt.savefig('Plots/Surface Height from Sea Level (SRTM Land).png',dpi=800)
plt.savefig('Plots/Surface_Height_from_Sea_Level_(SRTM_Land_Lake).png',dpi=800)

fig.show()

#Land Use Index

fig = plt.figure()
ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]
hz=plt.tricontourf(lon,lat,land,cmap=plt.get_cmap('jet'),levels=np.linspace(0,21,22))
#, levels=np.linspace(286,295,10)
plt.colorbar(hz,label='Land_Use_Index')
ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlabel('Distance_from_West_to_East[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North[km]',fontsize=18)
fig.tight_layout()

#plt.savefig('Plots/Land Use Index (GTOPO).png',dpi=800)
#plt.savefig('Plots/Land Use Index (SRTM).png',dpi=800)
#plt.savefig('Plots/Land Use Index (SRTM Land).png',dpi=800)
plt.savefig('Plots/Land_Use_Index_(SRTM_Land_Lake).png',dpi=800)

fig.show()

#Potential temperature at 2 m

fig = plt.figure()

```

```

ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]
hz=plt.tricontourf(lon,lat,ptemp,cmap=plt.get_cmap('jet'),levels=np.linspace(272,300,29))
#,levels=np.linspace(286,295,10)
plt.colorbar(hz,label='Potential_Temperature_(K)')
ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlabel('Distance_from_West_to_East_[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North_[km]',fontsize=18)
fig.tight_layout()

#plt.savefig('Plots/Potential Temperature (GTOPO) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Potential Temperature (SRTM) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Potential Temperature (SRTM Land) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Potential Temperature (SRTM Land Lake) at 0200 MST.png',dpi=800)

fig.show()

#Wind Velocity at 10 m

fig = plt.figure()
ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]
hz=plt.tricontourf(lon,lat,smean,cmap=plt.get_cmap('jet'),levels=np.linspace(0,6,25))
plt.colorbar(hz,label='Horizontal_Wind_Velocity_(m_s$^{-1}$)')
#Loop over lat and lon to plot normalized wind vectors
for i in range(0,np.size(lat),latlonstep):
    if (lon[i]-np.floor(lon[i]*(1/dlon))*dlon < latlonTol) and (lat[i]-np.floor(lat[i]*(1/
        dlat))*dlat < latlonTol):
        plt.quiver(lon[i], lat[i], u10[i]/(np.sqrt(u10[i]**2+v10[i]**2)), v10[i]/(np.sqrt(
            u10[i]**2+v10[i]**2)), pivot='mid',width=0.003)
ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlim(lonmin,lonmax)
plt.ylim(latmin,latmax)
plt.xlabel('Distance_from_West_to_East_[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North_[km]',fontsize=18)
fig.tight_layout()

plt.savefig('Plots/Horizontal_Wind_Velocity_(GTOPO)_at_0200_MST.png',dpi=800)

```

```

plt.savefig('Plots/Horizontal Wind Velocity (SRIM) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Horizontal Wind Velocity (SRIM Land) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Horizontal Wind Velocity (SRIM Land Lake) at 0200 MST.png',dpi=800)

fig.show()

#Skin temperature

fig = plt.figure()
ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]
hz=plt.tricontourf(lon,lat,skintemp,cmap=plt.get_cmap('jet'),levels=np.linspace(270,312,43))
#, levels=np.linspace(286,295,10)
plt.colorbar(hz,label='Skin_Temperature_(K)')
ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlabel('Distance_from_West_to_East_[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North_[km]',fontsize=18)
fig.tight_layout()

plt.savefig('Plots/Skin Temperature (GTOPO) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Skin Temperature (SRIM) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Skin Temperature (SRIM Land) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Skin Temperature (SRIM Land Lake) at 0200 MST.png',dpi=800)

fig.show()

#Relative Humidity at 2 m

fig = plt.figure()
ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]
hz=plt.tricontourf(lon,lat,RH,cmap=plt.get_cmap('jet'),levels=np.linspace(0,100,101))
#, levels=np.linspace(286,295,10)
plt.colorbar(hz,label='Relative_Humidity_($\%$)')

ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlabel('Distance_from_West_to_East_[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North_[km]',fontsize=18)

```

```

fig.tight_layout()

#plt.savefig('Plots/Relative Humidity (GTOPO) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Relative Humidity (SRTM) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Relative Humidity (SRTM Land) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Relative_Humidity_(SRTM_Land_Lake)_at_0200_MST.png',dpi=800)
fig.show()

#Heat flux at 10 m

fig = plt.figure()
ax = plt.gca()
#levels = [0, 5, 10, 15, 20, 25, 30, 35, 40]
hz=plt.tricontourf(lon,lat,HFX,cmap=plt.get_cmap('jet'),levels=np.linspace(-130,470,61))
#,levels=np.linspace(286,295,10)
plt.colorbar(hz,label='Vertical_Turbulent_Heat_Flux_(W_m$^{-2}$)')

ax.set_xticks
    ([-111.98599243,-111.93679810,-111.88760376,-111.83615112,-111.78671265,-111.73519897,-111.68626404])

ax.set_xticklabels(['0','3','6','9','12','15','18'])
ax.set_yticks([57.24551010, 57.27220917, 57.29991913, 57.32666397, 57.35321426, 57.37996674,
    57.40763474])
ax.set_yticklabels(['0','3','6','9','12','15','18'])
plt.xlabel('Distance_from_West_to_East[km]',fontsize=18)
plt.ylabel('Distance_from_South_to_North[km]',fontsize=18)
fig.tight_layout()

#plt.savefig('Plots/Vertical Turbulent Heat Flux (GTOPO) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Vertical Turbulent Heat Flux (SRTM) at 0200 MST.png',dpi=800)
#plt.savefig('Plots/Vertical Turbulent Heat Flux (SRTM Land) at 0200 MST.png',dpi=800)
plt.savefig('Plots/Vertical_Turbulent_Heat_Flux_(SRTM_Land_Lake)_at_0200_MST.png',dpi=800)
fig.show()

plt.show()

```

### D.3.3 WRF Profile Plotter

```

import numpy as np
import matplotlib.pyplot as plt
#plt.rcParams.update({'figure.max_open_warning': 0})

#Define text and font

plt.rc('text', usetex=True)
plt.rc('font', family='Times_New_Roman', size='16')

#SODAR

filename1 ="Mine_68/wrfout_2018-05-18_ForestProfile.txt"

```

```

#Load all data in a matrix

data = np.loadtxt(filename1)

Time_Forest=data[:,0]
Altitude_Forest=data[:,1]
S_Forest=data[:,4]
W_Forest=data[:,5]
D_Forest=data[:,6]
PT_Forest=data[:,9]

filename2 ="Mine_68/wrfout_2018-05-18_PondProfile.txt"

data = np.loadtxt(filename2)

Time_Pond=data[:,0]
Altitude_Pond=data[:,1]
S_Pond=data[:,4]
W_Pond=data[:,5]
D_Pond=data[:,6]
PT_Pond=data[:,9]

filename3 ="Mine_68/wrfout_2018-05-18_BarrenProfile.txt"

data = np.loadtxt(filename3)
Time_Barren=data[:,0]
Altitude_Barren=data[:,1]
S_Barren=data[:,4]
W_Barren=data[:,5]
D_Barren=data[:,6]
PT_Barren=data[:,9]

filename4="Mine_68/wrfout_2018-05-18_MineProfile.txt"

data = np.loadtxt(filename4)
Time_Mine=data[:,0]
Altitude_Mine=data[:,1]
S_Mine=data[:,4]
W_Mine=data[:,5]
D_Mine=data[:,6]
PT_Mine=data[:,9]

ind_time_02= [x for x in range(0,len(Altitude_Forest)) if (Time_Forest[x]==2)]

fig = plt.figure()
plt.plot(S_Forest[ind_time_02],Altitude_Forest[ind_time_02],color='darkblue',marker='o',
         markersize=7, markerfacecolor='none',linewidth=0.6,label='Forest')
plt.plot(S_Pond[ind_time_02],Altitude_Pond[ind_time_02],color='red',marker='s',markersize=7,
         markerfacecolor='none',linewidth=0.6,label='Tailings_Pond')
plt.plot(S_Barren[ind_time_02],Altitude_Barren[ind_time_02],color='green',marker='^',
         markersize=7, markerfacecolor='none',linewidth=0.6,label='Barren')

```

```

plt.plot(S_Mine[ind_time_02], Altitude_Mine[ind_time_02], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[-1]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Horizontal_Wind_Velocity_at_Different_Locations_at_0200_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W_Forest[ind_time_02], Altitude_Forest[ind_time_02], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(W_Pond[ind_time_02], Altitude_Pond[ind_time_02], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(W_Barren[ind_time_02], Altitude_Barren[ind_time_02], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(W_Mine[ind_time_02], Altitude_Mine[ind_time_02], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[-1]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Vertical_Wind_Velocity_at_Different_Locations_at_0200_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(D_Forest[ind_time_02], Altitude_Forest[ind_time_02], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(D_Pond[ind_time_02], Altitude_Pond[ind_time_02], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(D_Barren[ind_time_02], Altitude_Barren[ind_time_02], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(D_Mine[ind_time_02], Altitude_Mine[ind_time_02], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim(0, 360)
plt.xlabel('Wind_Direction_[$^{o}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Wind_Direction_at_Different_Locations_at_0200_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(PT_Forest[ind_time_02], Altitude_Forest[ind_time_02], color='darkblue', marker='o',

```



```

        markersize=7, markerfacecolor='none',linewidth=0.6,label='Forest')
plt.plot(PT_Pond[ind_time_02],Altitude_Pond[ind_time_02],color='red',marker='s',markersize
        =7, markerfacecolor='none',linewidth=0.6,label='Tailings_Pond')
plt.plot(PT_Barren[ind_time_02],Altitude_Barren[ind_time_02],color='green',marker='^',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Barren')
plt.plot(PT_Mine[ind_time_02],Altitude_Mine[ind_time_02],color='m',marker='>',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='Mine')

plt.xlim()
plt.xlabel('Potential_Temperature_[K]',fontsize=18)
plt.ylabel('Altitude_[m]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)',fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Potential_Temperature_at_Different_Locations_at_0200_MST.png',dpi=800)
fig.show()

ind_time_06= [x for x in range(0,len(Altitude_Forest)) if (Time_Forest[x]==6)]

fig = plt.figure()
plt.plot(S_Forest[ind_time_06],Altitude_Forest[ind_time_06],color='darkblue',marker='o',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Forest')
plt.plot(S_Pond[ind_time_06],Altitude_Pond[ind_time_06],color='red',marker='s',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='Tailings_Pond')
plt.plot(S_Barren[ind_time_06],Altitude_Barren[ind_time_06],color='green',marker='^',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Barren')
plt.plot(S_Mine[ind_time_06],Altitude_Mine[ind_time_06],color='m',marker='>',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='Mine')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
plt.ylabel('Altitude_[m]',fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)',fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Horizontal_Wind_Velocity_at_Different_Locations_at_0600_MST.png',dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W_Forest[ind_time_06],Altitude_Forest[ind_time_06],color='darkblue',marker='o',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Forest')
plt.plot(W_Pond[ind_time_06],Altitude_Pond[ind_time_06],color='red',marker='s',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='Tailings_Pond')
plt.plot(W_Barren[ind_time_06],Altitude_Barren[ind_time_06],color='green',marker='^',
        markersize=7, markerfacecolor='none',linewidth=0.6,label='Barren')
plt.plot(W_Mine[ind_time_06],Altitude_Mine[ind_time_06],color='m',marker='>',markersize=7,
        markerfacecolor='none',linewidth=0.6,label='Mine')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[m_s$^{-1}$]',fontsize=18)
plt.ylabel('Altitude_[m]',fontsize=18)

```

```

#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend( fontsize='12' )
fig.tight_layout()
plt.savefig( 'Plots/Vertical_Wind_Velocity_at_Different_Locations_at_0600_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(D.Forest[ind_time_06], Altitude_Forest[ind_time_06], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(D.Pond[ind_time_06], Altitude_Pond[ind_time_06], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(D.Barren[ind_time_06], Altitude_Barren[ind_time_06], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(D.Mine[ind_time_06], Altitude_Mine[ind_time_06], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim(0,360)
plt.xlabel('Wind_Direction_[$\{^o\}]$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend( fontsize='12' )
fig.tight_layout()
plt.savefig( 'Plots/Wind_Direction_at_Different_Locations_at_0600_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(PT.Forest[ind_time_06], Altitude_Forest[ind_time_06], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(PT.Pond[ind_time_06], Altitude_Pond[ind_time_06], color='red', marker='s', markersize=
         =7, markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(PT.Barren[ind_time_06], Altitude_Barren[ind_time_06], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(PT.Mine[ind_time_06], Altitude_Mine[ind_time_06], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Potential_Temperature_[K]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend( fontsize='12' )
fig.tight_layout()
plt.savefig( 'Plots/Potential_Temperature_at_Different_Locations_at_0600_MST.png', dpi=800)
fig.show()

ind_time_10= [x for x in range(0,len(Altitude_Forest)) if (Time_Forest[x]==10)]

fig = plt.figure()
plt.plot(S.Forest[ind_time_10], Altitude_Forest[ind_time_10], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(S.Pond[ind_time_10], Altitude_Pond[ind_time_10], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')

```

```

plt.plot(S_Barren[ind_time_10], Altitude_Barren[ind_time_10], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(S_Mine[ind_time_10], Altitude_Mine[ind_time_10], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[-1]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Horizontal_Wind_Velocity_at_Different_Locations_at_1000_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W_Forest[ind_time_10], Altitude_Forest[ind_time_10], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(W_Pond[ind_time_10], Altitude_Pond[ind_time_10], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(W_Barren[ind_time_10], Altitude_Barren[ind_time_10], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(W_Mine[ind_time_10], Altitude_Mine[ind_time_10], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[-1]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Vertical_Wind_Velocity_at_Different_Locations_at_1000_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(D_Forest[ind_time_10], Altitude_Forest[ind_time_10], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(D_Pond[ind_time_10], Altitude_Pond[ind_time_10], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(D_Barren[ind_time_10], Altitude_Barren[ind_time_10], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(D_Mine[ind_time_10], Altitude_Mine[ind_time_10], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim(0,360)
plt.xlabel('Wind_Direction_[{0}]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Wind_Direction_at_Different_Locations_at_1000_MST.png', dpi=800)
fig.show()

```

```

fig = plt.figure()
plt.plot(PT.Forest[ind_time_10], Altitude_Forest[ind_time_10], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(PT.Pond[ind_time_10], Altitude_Pond[ind_time_10], color='red', marker='s', markersize=
         =7, markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(PT.Barren[ind_time_10], Altitude_Barren[ind_time_10], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(PT.Mine[ind_time_10], Altitude_Mine[ind_time_10], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Potential_Temperature_[K]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Potential_Temperature_at_Different_Locations_at_1000_MST.png', dpi=800)
fig.show()

```

```

ind_time_14= [x for x in range(0, len(Altitude_Forest)) if (Time_Forest[x]==14)]

```

```

fig = plt.figure()
plt.plot(S.Forest[ind_time_14], Altitude_Forest[ind_time_14], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(S.Pond[ind_time_14], Altitude_Pond[ind_time_14], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(S.Barren[ind_time_14], Altitude_Barren[ind_time_14], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(S.Mine[ind_time_14], Altitude_Mine[ind_time_14], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Horizontal_Wind_Velocity_at_Different_Locations_at_1400_MST.png', dpi=800)
fig.show()

```

```

fig = plt.figure()
plt.plot(W.Forest[ind_time_14], Altitude_Forest[ind_time_14], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(W.Pond[ind_time_14], Altitude_Pond[ind_time_14], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(W.Barren[ind_time_14], Altitude_Barren[ind_time_14], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(W.Mine[ind_time_14], Altitude_Mine[ind_time_14], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()

```

```

plt.xlabel('Vertical_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Vertical_Wind_Velocity_at_Different_Locations_at_1400_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(D.Forest[ind_time_14], Altitude_Forest[ind_time_14], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(D.Pond[ind_time_14], Altitude_Pond[ind_time_14], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(D.Barren[ind_time_14], Altitude_Barren[ind_time_14], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(D.Mine[ind_time_14], Altitude_Mine[ind_time_14], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim(0,360)
plt.xlabel('Wind_Direction_[${}^{\circ}]$', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Wind_Direction_at_Different_Locations_at_1400_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(PT.Forest[ind_time_14], Altitude_Forest[ind_time_14], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(PT.Pond[ind_time_14], Altitude_Pond[ind_time_14], color='red', marker='s', markersize=
         =7, markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(PT.Barren[ind_time_14], Altitude_Barren[ind_time_14], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(PT.Mine[ind_time_14], Altitude_Mine[ind_time_14], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Potential_Temperature_[K]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Potential_Temperature_at_Different_Locations_at_1400_MST.png', dpi=800)
fig.show()

ind_time_18= [x for x in range(0, len(Altitude_Forest)) if (Time_Forest[x]==18)]

fig = plt.figure()
plt.plot(S.Forest[ind_time_18], Altitude_Forest[ind_time_18], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')

```

```

plt.plot(S_Pond[ind_time_18], Altitude_Pond[ind_time_18], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(S_Barren[ind_time_18], Altitude_Barren[ind_time_18], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(S_Mine[ind_time_18], Altitude_Mine[ind_time_18], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[' + m_s$^{[-1]}$ + ']', fontsize=18)
plt.ylabel('Altitude_[' + m + ']', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Horizontal_Wind_Velocity_at_Different_Locations_at_1800_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W_Forest[ind_time_18], Altitude_Forest[ind_time_18], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(W_Pond[ind_time_18], Altitude_Pond[ind_time_18], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(W_Barren[ind_time_18], Altitude_Barren[ind_time_18], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(W_Mine[ind_time_18], Altitude_Mine[ind_time_18], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[' + m_s$^{[-1]}$ + ']', fontsize=18)
plt.ylabel('Altitude_[' + m + ']', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Vertical_Wind_Velocity_at_Different_Locations_at_1800_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(D_Forest[ind_time_18], Altitude_Forest[ind_time_18], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(D_Pond[ind_time_18], Altitude_Pond[ind_time_18], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(D_Barren[ind_time_18], Altitude_Barren[ind_time_18], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(D_Mine[ind_time_18], Altitude_Mine[ind_time_18], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim(0, 360)
plt.xlabel('Wind_Direction_[' + $^{[o]}$ + ']', fontsize=18)
plt.ylabel('Altitude_[' + m + ']', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Wind_Direction_at_Different_Locations_at_1800_MST.png', dpi=800)

```

```

fig.show()

fig = plt.figure()
plt.plot(PT.Forest[ind_time_18], Altitude_Forest[ind_time_18], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(PT.Pond[ind_time_18], Altitude_Pond[ind_time_18], color='red', marker='s', markersize=
         =7, markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(PT.Barren[ind_time_18], Altitude_Barren[ind_time_18], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(PT.Mine[ind_time_18], Altitude_Mine[ind_time_18], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Potential_Temperature_[K]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12', loc=9)
fig.tight_layout()
plt.savefig('Plots/Potential_Temperature_at_Different_Locations_at_1800_MST.png', dpi=800)
fig.show()

ind_time_22= [x for x in range(0, len(Altitude_Forest)) if (Time_Forest[x]==22)]

fig = plt.figure()
plt.plot(S.Forest[ind_time_22], Altitude_Forest[ind_time_22], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(S.Pond[ind_time_22], Altitude_Pond[ind_time_22], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(S.Barren[ind_time_22], Altitude_Barren[ind_time_22], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(S.Mine[ind_time_22], Altitude_Mine[ind_time_22], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Horizontal_Wind_Velocity_[m_s$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Horizontal_Wind_Velocity_at_Different_Locations_at_2200_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(W.Forest[ind_time_22], Altitude_Forest[ind_time_22], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(W.Pond[ind_time_22], Altitude_Pond[ind_time_22], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(W.Barren[ind_time_22], Altitude_Barren[ind_time_22], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(W.Mine[ind_time_22], Altitude_Mine[ind_time_22], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

```

```

plt.xlim()
plt.xlabel('Vertical_Wind_Velocity_[-ms$^{-1}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Vertical_Wind_Velocity_at_Different_Locations_at_2200_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(D.Forest[ind_time_22], Altitude_Forest[ind_time_22], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(D.Pond[ind_time_22], Altitude_Pond[ind_time_22], color='red', marker='s', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(D.Barren[ind_time_22], Altitude_Barren[ind_time_22], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(D.Mine[ind_time_22], Altitude_Mine[ind_time_22], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim(0,360)
plt.xlabel('Wind_Direction_[$\{o\}$]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Wind_Direction_at_Different_Locations_at_2200_MST.png', dpi=800)
fig.show()

fig = plt.figure()
plt.plot(PT.Forest[ind_time_22], Altitude_Forest[ind_time_22], color='darkblue', marker='o',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Forest')
plt.plot(PT.Pond[ind_time_22], Altitude_Pond[ind_time_22], color='red', marker='s', markersize=
         =7, markerfacecolor='none', linewidth=0.6, label='Tailings_Pond')
plt.plot(PT.Barren[ind_time_22], Altitude_Barren[ind_time_22], color='green', marker='^',
         markersize=7, markerfacecolor='none', linewidth=0.6, label='Barren')
plt.plot(PT.Mine[ind_time_22], Altitude_Mine[ind_time_22], color='m', marker='>', markersize=7,
         markerfacecolor='none', linewidth=0.6, label='Mine')

plt.xlim()
plt.xlabel('Potential_Temperature_[K]', fontsize=18)
plt.ylabel('Altitude_[m]', fontsize=18)
#plt.title('Effect of Topography on Horizontal Wind Velocity (SODAR)', fontsize=18)
plt.legend(fontsize='12')
fig.tight_layout()
plt.savefig('Plots/Potential_Temperature_at_Different_Locations_at_2200_MST.png', dpi=800)
fig.show()

plt.show()

```



# Appendix E

## Published Work

### E.1 Peer-Reviewed Journal Papers

1. Nahian, M. R., Nazem, A., Nambiar, M. K., Byerlay, R., Mahmud, S., Seguin, M., Robe, F., Ravenhill, J., and Aliabadi, A. A., Complex Meteorology over a Complex Mining Facility: Assessment of Topography, Land Use, Grid Resolution, and PBL Scheme Modifications in WRF. *Applied Meteorology and Climatology*, (under review).
2. Nambiar, M. K., Byerlay, R., Nazem, A., Nahian, M. R., Moradi, M., and Aliabadi, A. A., A Tethered And Navigated Air Blimp (TANAB) for observing the microclimate over a complex terrain. *Geoscientific Instrumentation, Methods and Data Systems*, (under review).
3. Byerlay, R., Nambiar, M. K., Nazem, A., Nahian, M. R., Biglarbegian, M., and Aliabadi, A. A., An Imaging Technique to Identify Land Surface Temperatures Using Oblique Angle Airborne Observations. *International Journal of Remote Sensing and Remote Sensing Letters*, (under review).
4. Moradi, M., Dyer, B., Nazem, A., Nambiar, M. K., Nahian, M. R., Bueno, B., Mackey, C., Vasathakumar, S., Nazarian, N., Krayenhoff, E. S., Norford, L. K., and Aliabadi, A. A., The Vertical City Weather Generator (VCWG). *Geoscientific Model Development*, (under review).

## E.2 Refereed Conferences

1. Nahian, M. R., Nazem, A., Moradi, M., Mahmud, S., Lubitz, W. D., and Aliabadi, A. A., WRF Simulations of Urban Climate: Do We Need Low or High Topographic Resolution? In *Proceedings of The Joint Canadian Society for Mechanical Engineering (CSME) and CFD Society of Canada (CFDSC) International Congress (2019)*, London, Ontario, Canada.
2. Moradi, M., Kizhakkeniyil, M. K., Nazem, A., Nahian, M. R., Lubitz, W. D., and Aliabadi, A. A., A Vertical Diffusion Model to Predict Profiles of Temperature within the Lower Atmospheric Surface Layer: Simple or Complicated? In *Proceedings of The Joint Canadian Society for Mechanical Engineering (CSME) and CFD Society of Canada (CFDSC) International Congress (2019)*, London, Ontario, Canada.

## E.3 Poster Presentations

1. Nazem, A., Nahian, M. R., Nambiar, M. K., Byerlay, R., and Aliabadi, A. A., Complex Meteorology over a Complex Mining Facility. In *Proceedings of The 27th IUGG General Assembly (2019)*, Montréal, Québec, Canada.