2017

Explorations into Machine Learning Techniques for Precipitation Nowcasting

Aditya Nagarajan
University of Massachusetts Amherst

EXPLORATIONS INTO MACHINE LEARNING TECHNIQUES FOR PRECIPITATION NOWCASTING

A Thesis Presented

by

ADITYA NAGARAJAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH

February 2017

Mechanical and Industrial Engineering

EXPLORATIONS INTO MACHINE LEARNING TECHNIQUES FOR PRECIPITATION NOWCASTING

A Thesis Presented

by

ADITYA NAGARAJAN

Approved as to style and content by:

Michael Zink, Chair

David L. Pepyne, Member

Hari Balasubramanian, Member

Sundar Krishnamurty, Department Head
Mechanical and Industrial Engineering

DEDICATION

To my grandfather

Mayuram Viswanathan Krishnamurthy

ACKNOWLEDGMENTS

This work was performed within the UMass Center for Collaborative Adaptive Sensing of the Atmosphere (CASA) under funding provided by the Jerome M. Paros Fund for Measurement and Environmental Science Research. I would like to thank my advisor Dr. David L. Pepyne for his insights and guidance into this research, without which this work would not have been possible. I would also like to thank Professor Michael Zink for giving me the opportunity and the resources to pursue research in machine learning.

ABSTRACT

EXPLORATIONS INTO MACHINE LEARNING TECHNIQUES FOR PRECIPITATION NOWCASTING

FEBRUARY 2017

ADITYA NAGARAJAN

B.E, MADRAS INSTITUTE OF TECHNOLOGY

M.S I.E.O.R, UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Michael Zink

Recent advances in cloud-based big-data technologies now makes data driven solutions feasible for increasing numbers of scientific computing applications. One such data driven solution approach is machine learning where patterns in large data sets are brought to the surface by finding complex mathematical relationships within the data. Nowcasting or short-term prediction of rainfall in a given region is an important problem in meteorology. In this thesis we explore the nowcasting problem through a data driven approach by formulating it as a machine learning problem.

State-of-the-art nowcasting systems today are based on numerical models which describe the physical processes leading to precipitation or on weather radar extrapolation techniques that predict future radar precipitation maps by advecting from a sequence of past maps. These techniques, while they can perform well over very short prediction horizons (minutes) or very long horizons (hours to days), tend not to perform well over medium horizons (1-2 hours) due to lack of input data at the necessary

spatial and temporal scales for the numerical prediction methods or due to the inability of radar extrapolation methods to predict storm growth and decay. Given that water must first concentrate in the atmosphere as water vapor before it can fall to the ground as rain, one goal of this thesis is to understand if water vapor information can improve radar extrapolation techniques by giving the information needed to infer growth and decay. To do so, we use the GPS-Meteorology technique to measure the water vapor in the atmosphere and weather radar reflectivity to measure rainfall. By training a machine learning nowcasting algorithm using both variables and comparing its performance against a nowcasting algorithm trained on reflectivity alone, we draw conclusions as to the predictive power of adding water vapor information.

Another goal of this thesis is to compare different machine learning techniques, viz., the random forest ensemble learning technique, which has shown success on a number of other weather prediction problems, and the current state-of-the-art machine learning technique for images and image sequences, convolutional neural network (CNN). We compare these in terms of problem representation, training complexity, and nowcasting performance.

A final goal is to compare the nowcasting performance of our machine learning techniques against published results for current state-of-the-art model based nowcasting techniques.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

We live in an age where large volumes of data generated each day can provide us with valuable insights to the underlying system the data is representing. Coupled with significant advances in big data technology, we are now able to extract complex patterns from high-dimensional data within reasonable time. Mastering the ability to processes large data sets to extract actionable insights using sophisticated machine learning algorithms has engendered significant interest in Data Science. In this thesis we explore the short-term weather prediction (aka nowcasting) problem from a Data Science perspective. Specifically this thesis will seek to apply data science and machine-learning techniques to the problem of nowcasting precipitation fields 1 hour in the future from time-sequences of past spatial fields of weather radar reflectivity and precipitable water vapor. These two fields are complementary, weather radar reflectivity being a measure of location and intensity of precipitation and precipitable water vapor (otherwise termed as Integrated Precipitable Water or IPW) a measure of the amount of water in the atmosphere that could potentially fall as precipitation. Thus, where weather radar reflectivity tells us where precipitation is currently falling, IPW tells us about the potential for further precipitation. Using the complementary nature of these two measurements we explore the spatiotemporal patterns and correlations between these fields to make 1 hour precipitation nowcasts through the use of various machine learning algorithms. In addition to the challenge of obtaining and pre-processing the input data for machine-learning, we also face the challenge of developing a machine learning algorithm that can handle spatiotemporal input data

– essentially short video streams of radar reflectivity and IPW fields to predict the precipitation field 1 hours in the future.

1.1 Precipitation Nowcasting

Because rain affects so many human activities, predicting rain has a long history. Whereas long-term rainfall forecasts (i.e., beyond 3 hours in the future) are based on models of atmospheric processes (cf. [35]), short-term 0-3 hour nowcasts are frequently based on weather radar data. This is because numerical weather models tend to be too computationally time-consuming to obtain the resolutions desired for short-term predictions. Nowcasting or "very-short-term" forecasting at the resolution of a few kilometers are useful in aviation applications to route arriving and departing air traffic at airports. They also find applications in large sporting events where nowcasts can help to determine when to cover the playing fields. Nowcasts from weather radar data can be "manual" as when a weather radar meteorologist plays a radar reflectivity loop to infer where and how fast a storm is moving.

Nowcasts from weather radar data can also be automated. The Storm Cell Identification and Tracking (SCIT) algorithm [30] uses the history radar data to identify storm cells and estimate their speed and direction. Pixel based approaches have been taken [62] that use Lagrangian models to nowcast rainfall one to three hours into the future. One such algorithm, Dynamic Adaptive Radar Tracking of Storms (DARTS)[45], uses a Lagrangian persistence based nowcast model to project radar reflectivity fields 1-20 minutes into the future. Nowcasts based on weather radar data alone tend to quickly break down, so that a 20-minute DARTS nowcast of the reflectivity will often bear little resemblance to the actual reflectivity field that occurs 20 minutes in the future. This is because nowcast techniques based on weather radar data alone, while they can obtain a good estimate of storm advection, have very little skill at predicting storm growth and decay. They are thus very poor at predicting

that a storm will pop-up at a given location when there is currently no radar data coming from that location, and they are very poor at recognizing that a storm will dissipate 20 minutes from now when it is currently growing in intensity. This has lead many researchers and atmospheric scientists to look for other information with which to augment weather radar data.

It was discovered in the early 1990s that GPS signal propagation delays can be used to infer atmospheric water vapor content [10] [9]. Since then there has been significant advances both in hardware and software to accurately measure atmospheric water vapor more frequently and accurately using the GPS-Met technique. Agencies such as NOAA (National Oceanic and Atmospheric Administration), UCAR (University Collaboration of Atmospheric Research) and SOPAC (Scripps Orbital and Permanent Array Center) in the United States have contributed to the development, operation and maintenance of a nationwide realtime GPS-based water vapor monitoring system [59] [11]. This has lead to the availability of real-time water vapor products to the public and operational forecasters from over 500 GPS-Met stations distributed across the continental United States. In addition, it is now also possible to obtain mature, validated software for GPS-Met calculations (e.g. GAMIT [26]) allowing for research deployments of GPS-Met stations [1] and repurposing of GIS Continuously Operating GPS Reference Stations (CORS http://geodesy.noaa.gov/CORS/) stations for the GPS-Met application.

Motivated by the fact that in the water cycle that describes the movement of water through the atmosphere, water must first exist in the atmosphere as water vapor before it becomes rain, a number of agencies and researchers have explored the potential of real-time observations of atmospheric water vapor content derived from GPS-Met stations for weather forecasting and precipitation nowcasting. Japan thus far has the highest density of GPS stations with an average spacing between stations of 17km [50]. Using this densely spaced GPS network, a number of studies have

looked into the ability of high spatial-temporal resolution GPS-Met derived Integrated Precipitable Water Vapor (IPW) to nowcast thunderstorms and severe rain, e.g., [29], [47]. In a study of how IPW fields relate to the onset of convective weather, it was pointed out that maximum IPW occurred 1-2 hours prior to thunderstorm activity where thunderstorms were measured using cloud-to-ground lightning and convective activity was measured by hourly accumulated rainfall [28]. In another study looking at relationships between spatial variations in IPW and precipitation it was shown that rainfall intensity is related to IPW gradients and in particular that strong convergence (concentration) of water vapor is generally present several hours in advance of convective precipitation [61].

Spatial variations in IPW and their correlations with thunderstorm activity have also been studied in Europe. De Haan [16] devised a method to construct IPW maps from a network of GPS stations using two dimensional variational techniques. The IPW maps were then studied with regard to lightning and thunderstorm events in the Netherlands. The level of convergence of water vapor evident in the IPW fields again correlated well with subsequent precipitation rates and thunderstorm activity. A similar analysis was conducted in Spain [57] using normalized IPW fields to take into account the seasonal variation. The interesting observation made in the paper is that the decay of IPW values coincided with storm direction and intensity.

Generation and validation of IPW spatial fields have also been carried out in the US and the potential use of IPW fields for analyzing long and short term climatological activity have been studied (cf.[40]). These studies have shown that IPW fields generated from point measurements using GPS-Met systems prove to be an excellent tool to visualize convergence and build up of water vapor. They observed a strong build-up of water vapor 1-3 hours in advance of a convective event, leading to the conclusion that IPW holds the potential to accurately predict convective initiation. A recent study conducted in Malaysia evaluated various spatial interpolation techniques

for obtaining water vapor fields from networks of ground based GPS stations. The spatial interpolation techniques were then suggested for applications in nowcasting precipitation [54].

A common indicator of rainfall is a steep increase in IPW seen from individual GPS-Met stations. [49] [2] showed that variations in IPW usually peaks a few hours prior to the onset of precipitation and that the variations in IPW and the variations of rainfall rate measured by rain gauges are also strongly correlated. However both papers also note that IPW alone is not a precise predictor of rainfall and there are other atmospheric parameters that play a role in the onset of rain. This final conclusion is based on observing several cases where high peaks in IPW values did not result in heavy rainfall.

As presented in the previous paragraphs that several research efforts have been made to explore the possibility for using GPS-Met stations for forecasting/nowcasting precipitation and severe weather, no one has taken the step forward to build an automated system which incorporates data from GPS-Met stations. Based on the above findings, we propose a precipitation nowcasting approach that uses both IPW and weather radar reflectivity. Rather than attempting a prediction system that tries to model and explain how these two fields jointly evolve, we propose instead a machine learning approach to learn the joint spatial-temporal patterns and correlations to be able to better predict rainfall.

The use of machine learning for precipitation nowcasting is not new. An early application of machine learning to the precipitation nowcasting problem used artificial neural networks to predict rainfall fields 1 hour in advance based on the current rainfall field [22]. A single layer neural network was trained using a 100×100 km^2, 4 km resolution simulated precipitation field to predict the corresponding precipitation field 1 hour in the future. The predictions were evaluated against mean areal index (MAI) and PAC (Percent Areal Coverage) and performed as well as the standard

nowcasting algorithm of the time. A neural network approach was also applied to forecasting rain gauge readings with 0-6 hr lead times using moisture and updraft data from the NCEP (National Center for Environmental prediction) Nested Grid Point Model. A feature selection method was used to reduce the initial 528 feature input space to the 25 most important features [32].

A machine learning approach using random forests and logistic regression was used in [38] to make probabilistic predictions 1-hour ahead of convective storms. The inputs were GOES satellite and numerical weather prediction (NWP) data and the prediction goal was to determine if a particular cloud would turn into a convective storm. As part of the analysis, this paper used the feature importance attribute of the random forest method to determine relative importance of each feature in predicting convective initiation. In an another study, a modified version of the random forest called the Spatiotemporal Relational Random Forest (SRRF) [37] was used to detect turbulence areas for aircraft. This algorithm used archived NWP estimates of the weather and meteorological observations. This algorithm achieved a 0.80 area under the receiver operating characteristic (ROC) curve detection rate of turbulence with 100 trees in the forest.

The state-of-the-art machine learning algorithms of today fall under the category of "Deep Learning" [4]. The word "Deep" was coined to emphasize the fact that this type of learning algorithm tries to learn multiple layers of representations of the input data space or representation learning as it is called in the literature. These multiple layers of representations take advantage of the spatial correlations in an input image or the temporal correlations of time series data. Some of the well known deep learning architectures include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and LSTM (Long Short Term Memory). A very recent application of Deep Learning and one of the first in the meteorological domain to tackle the nowcasting problem used a CNN-LSTM, a combination of CNN and LSTM, to predict

rainfall fields based on a sequence of radar reflectivity products [60]. Specifically, the goal was to predict the next 15 frames of reflectivity from the previous 5. The algorithm was trained and tested on a dataset of reflectivity echoes from a radar in Hong-Kong for rainfall days which occurred over a period of three years. The results showed that the CNN-LSTM performs better than the current model based state-of-the-art radar reflectivity extrapolation nowcasting algorithm called the Real-time Optical flow by Variational methods for Echoes of Radar (ROVER).

In this thesis we attempt to go a step beyond the works cited above (which use reflectivity only, IPW only, or use NWP data), to develop a machine-learning precipitation nowcasting system based on direct observations of IPW (water vapor) and reflectivity (precipitation). We develop a data pipeline which ingests data from a "network of networks" of remote sensing systems in real time. We make our software system an open source artifact for future researchers to use. We provide the data used to train our open source software and one of the goal of this thesis is to establish various bench marks on this data set against which future researchers can compare the results of the techniques they develop.

1.2 Thesis Goals

Precipitation nowcasting systems are generally based on the underlying physics model or radar reflectivity extrapolation approaches which make the persistence assumption (i.e. the system does not account for the growth and decay of storms). Our goal in this thesis is to build a nowcasting system by complementing the radar reflectivity data with water vapor data. Because water vapor convergence is correlated with rainfall, we believe that it should improve the nowcasting performance. By comparing nowcast systems that use water vapor information against ones that do not, we can evaluate this conjecture. The second goal is the general exploration of machine learning as an approach to the precipitation nowcasting problem. Given

the large amount of training data one can generate and recent developments in machine learning, e.g., the new deep learning techniques, we think machine learning is well suited to the precipitation nowcasting problem. By comparing a few machine learning algorithms against each other and against other nowcasting algorithms from the literature, we can evaluate the applicability of using machine learning algorithms to the precipitation nowcasting problem.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 will discuss the hydrologic cycle of how water moves into and out of the atmosphere in the various states and the theory behind the two remote-sensing systems (GPS-Met and Weather Radars) that we will use for our studies. Chapter 2 also gives a brief description of the interpolation technique we use to build IPW fields from IPW point measurements. Chapter 3 will discuss the region of Texas that will be the focus of our studies and how we collected, organized, and processed the data set we use for our experiments. We also analyze several storm cases using the reflectivity fields overlapped over the IPW fields visualizations, to visually understand if IPW can account for the growth and decay of a storm. Chapter 4 will give our solution approach and a brief introduction machine learning and background on the various algorithms we use in this thesis. Chapter 5 will present the results of the machine learning experiments and Chapter 6 will give the conclusions, recommendations for future work and discuss about some of the issues we encountered.

Readers only interested in the machine learning related work that has gone into this thesis can skip directly to chapter 4.

CHAPTER 2

PRECIPITATION BACKGROUND

In the hydrologic cycle that describes the movement of water into and out of the atmosphere, water must first exist as water vapor before it precipitates back to the earth as rain. After a simplified explanation of how water vapor becomes precipitation, this chapter describes the instruments we will use to measure atmospheric water vapor and rainfall.

2.1 Mechanisms of Precipitation

In the hydrologic cycle shown in Figure 2.1, water enters the atmosphere as vapor through evaporation and transpiration. Humidity, measured in mass of water per volume of atmosphere, is a measure of the amount of water vapor present in the atmosphere. The total amount of water vapor the atmosphere can hold is mainly a function of temperature. Relative humidity gives the saturation percentage - 0% implies no water vapor at all in the atmosphere, 100% implies the atmosphere is fully saturated and can hold no more. The dew point is the temperature at which the relative humidity becomes 100%.

Precipitation forms when there is uplift that forces warm moist air into increasingly colder air aloft. As the moist air is lifted, the relative humidity increases to 100% at which point the water vapor begins to condense into water droplets. Very small droplets (0.01 mm in size) remain suspended in the atmosphere to become clouds. The height of the cloud base being roughly the height where the temperature is equal to the dew point (a "cloud base" at ground level gives fog). For small

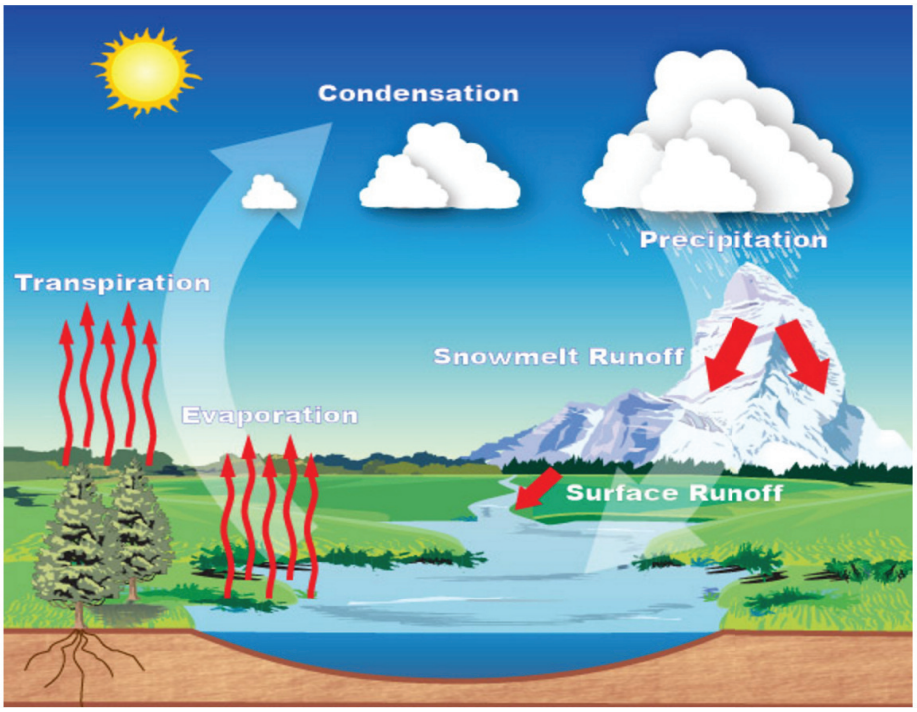

Figure 2.1: The basic hydrological cycle (www.srh.noaa.gov/jetstream/atmos/hydro.htm).

cloud droplets to become larger, they need a surface (condensation nuclei), in the form of dust, pollen or frozen ice crystals to coalesce onto. Once the water droplets become sufficiently large (e.g., greater than 0.1 mm in size), the vertical motion of the atmosphere can no longer hold them aloft and they begin to fall to earth as precipitation.

Figure 2.2 shows the two basic mechanisms leading to precipitation. The left of the figure shows a cold front, where convective uplift is caused by cold air being forced into warm moist air. This is the mechanism of thunderstorms and the super-cells that can spawn tornadoes. The right of the figure shows a warm front where warm moist air is dynamically forced into cold air. This is the mechanism of less violent and more widespread stratiform rain. For more detailed explanations of the hydrologic cycle, atmospheric water vapor, and the mechanisms of precipitation, the reader is referred to [6] and the summaries in [48] and [18].

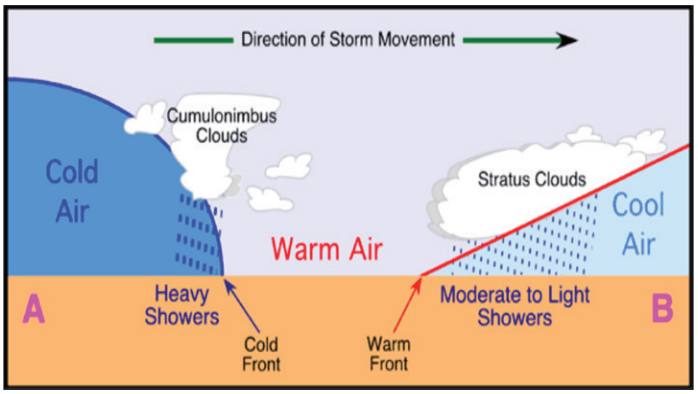

Figure 2.2: Basic mechanisms of precipitation - warm moist air uplifted by or into colder air. On the left is a cold front leading to convective lift and thunderstorms. On the right is a warm front leading to dynamic lift and stratiform rain. Figure from www.physicalgeography.net.

2.2 Measuring Atmospheric Water Vapor

There are a number of instruments for measuring the amount of water vapor in the atmosphere. These include:

1. Radiosondes: A radiosonde is a battery powered telemetry instrument package with sensors for sampling various atmospheric variables as it is carried up by a balloon from ground launch to between 20-30 km altitude[1]. While radiosondes have the advantage that they can measure the vertical distribution of water vapor, they have the distinct disadvantages that they are typically only launched two times a day (at 0000 and 1200 UTC) and from only a handful of locations (the entire CONUS is covered by a mere 90 radiosonde launch sites).

2. Radiometers. Ground-based water vapor radiometers measure the background microwave radiation emitted by atmospheric water vapor along a given line of site [7]. An advantage of these instruments is their ability to make continuous measurements of water vapor. Disadvantages are cost, calibration, and sparse

[1] http://www.wrh.noaa.gov/rev/tour/UA/introduction.php

spatial deployment. They are also limited in that they do not work when it is raining.

3. Satellites. The GOES (Geostationary Operational Environmental Satellite) system provides two sources of information about the water vapor [21]: imagery through its water vapor channel (at 4km spatial resolution, 15 min temporal resolution), and sounder retrievals (at 20km spatial resolution, 1hr temporal resolution). Both of these observations, however, are negatively impacted by cloud cover.

4. GPS-Meteorology. GPS-meteorology (GPS-Met) is a technique that allows GPS receivers to simultaneously perform the multiple functions of position estimation and precipitable water vapor estimation [10]. For a given GPS receiver, precipitable water vapor estimates can be made with 30-minute temporal resolution. In regions, such as the middle and western U.S., where there is a high density of Continuously Operated GPS Reference Stations (CORS), techniques have been developed to combine the water vapor measurements from multiple stations into 2D and 3D water vapor fields. The spatial resolution of the field depends on the density of GPS stations (spatial Nyquist). While GPS-Met currently cannot provide the spatial and temporal resolution of that of GOES satellite, it has the advantage that it is accurate in all weather conditions and not impacted by clouds or precipitation.

Based on our previous work [1], where we developed low-cost GPS-Met systems for near real-time Integrated Precipitable Water Vapor (IPW) estimation and an infrastructure for disseminating the IPW data on-line, we will use the GPS-Met technique as our source of atmospheric water vapor information.[2].

[2]UMass operates two low-cost GPS-Met stations in Dallas-Fort-Worth, Texas metroplex region, one site designation CNVL at the Univ. of Texas at Arlington and another site designation NWSD

2.2.1 GPS Meteorology Technique

The Global Positioning System (GPS) is a system of satellites operated by the U.S. Department of Defense (DoD). First launched in the 1970s for the purpose of military navigation, the system was later opened up for civilian use. The GPS system consists of a core of 24 satellites flying at 22,200 ft AGL orbiting in 6 different orbital planes inclined at 55° to each other[3]. For a GPS receiver located in the CONUS the number of satellites in view at any one time ranges between 8 and 12, though only 4 are required for an estimate of horizontal and vertical position.

The signal path from a given satellite to a GPS receiver is called a slant path. Since GPS satellites are not in geosynchronous orbit, the azimuth and elevation angles of the slant paths to the satellites in view change with time, as does the particular set of satellites in view[4]. Along each slant path a GPS receiver receives carrier signals at two distinct frequencies L_1 ($f_1 = 1575.42MHz$) and L_2 ($f_2 = 1227.60MHz$). These carrier signals are modulated as a sequence of bits called Pseudo Random Noise (PRN) and each satellite is identified by a unique PRN code [27]. From the carrier signals (code and carrier phase) the GPS receiver obtains measurements of the distance (pseudo-range) between the satellite and the receiver. As the GPS carrier signals travel from a satellite to a receiver, they accumulate delays that cause the pseudo-ranges to accumulate errors,

$$P_r^s(t_r) = \rho_r^s - (\delta t_r - \delta t^s) * c + \delta_{r,ion}^s + \delta_{r,trop}^s + \xi \tag{2.1}$$

Here P_r^s is the code pseudorange measurement from satellite s to receiver r, ρ_r^s is the geometric distance as a function of the receiver and satellite coordinates, and

---

at the NWS Dallas-Fort-Worth Weather Forecast Office (WFO). IPW observations from these two sites is published on-line at http://emmy9.casa.umass.edu/gpsmet/2015/)

[3]http://www.gps.gov/systems/gps/space/

[4]see the animation at wiki page for GPS https://en.wikipedia.org/wiki/Global_Positioning_System

the rest of the terms are range corrections - δt^s and δt_r the clock corrections for the satellite and receiver respectively, c is the speed of light in vacuum, $\delta^s_{r,ion}$ is the correction for the signal delay through the electrically charged ionosphere, $\delta^s_{r,trop}$ is the correction for the refractivity induced delays through the troposphere, and ξ are residual corrections for things like multipath delay and receiver and satellite hardware biases (cf. [48]).

For the geodest, position accuracy is limited by the accuracy of the clocks and the accuracies of the ionospheric and tropospheric corrections. For the meteorologist, its the tropospheric correction that is of interest, since the tropospheric delay is the term that varies with atmospheric water vapor content [10] [42] [20]. Estimating the tropospheric delay from the observed code range and carrier phase requires estimating and subtracting the other correction terms. For high accuracy, such as required by the GPS-Met application, the so-called double differencing technique is often used [41], [3]. This involves taking the differences of the pseudorange equations between two receivers and two different satellites and then taking the difference of these differences. If the baseline distance between the two receivers is sufficiently large ($> 500km$) that the observables are uncorrelated, then the result of double differencing is the elimination of both satellite and receiver clock errors (the $(\delta t_r - \delta t^s) * c$ term in eq. 2.1). For the electrically charged ionosphere, which is the region of the atmosphere between 60 and 1000 km altitude, the ionospheric delay (the $\delta^s_{r,ion}$ term in eq. 2.1) is frequency dependent (dispersive) [52]. This delay, which amounts to between 1 and 15 meters of pseudorange error, can be estimated to millimeter precision via linear combinations of the GPS dual frequency observables [42].[5] Residual errors (ξ in eq.

[5]The lowest cost GPS receivers, such as those in cell-phones, are single frequency receivers that use L_1 to estimate position. It is because these receivers cannot correct for the large ionospheric delay as accurately as dual-frequency receivers that they are not generally used for the GPS-Met application.

2.1), such as due to multipath, are avoided by careful selection of GPS site to avoid obstructions such as cell phone towers and buildings.

What remains after applying the above corrections is the tropospheric delay. The troposphere is the lowest portion of the atmosphere from the earth's surface to about 17 km and is the site of all weather on earth. The excess path length that GPS signals travel in the troposphere is due to refraction and can reach up to 2.5 meters at sea-level. This excess path length is given by [10], [48],

$$\delta^s_{r,trop} = 10^{-6} \int N ds + (S - G) \tag{2.2}$$

where N is the refractivity, S is the actual signal path and G is the geometric signal path respectively along the slant path between satellite s and receiver r, and the integral is along the slant path. The refractivity, N, can be split into a dry part, N_h, (refractivity of dry air) and a wet part, N_w, (refractivity due to water vapor) [15], [46],

$$\delta^s_{r,trop} = 10^{-6} \int (N_h + N_w) ds + (S - G) \tag{2.3}$$

The refractivity's depend on pressure, temperature, and humidity according to,

$$N_h = 77.6(\frac{P_d}{T}), N_w = 64.8(\frac{P_w}{T}) + 3.776 \cdot 10^5(\frac{P_w}{T^2}) \tag{2.4}$$

where P_d and P_w are the partial pressures (in millibars) of dry air and water vapor respectively, and T is the surface temperature (in degrees Kelvin).

A GPS-Met estimation of the right hand side of eq. 2.3 proceeds as follows. The tropospheric delay along a slant path is called the Slant Total Delay (STD). This is broken into a Slant Hydrostatic Delay (SHD) term and a Slant Wet Delay (SWD) term,

$$STD = SHD + SWD \tag{2.5}$$

of these, the SHD accounts for the majority of the excess path length, or about 2 meters, while the SWD accounts for only about 1-2 meters of excess path length. The STD is commonly written in terms of the hydrostatic and wet delays in the zenith direction as (cf.[20]),

$$STD = m_h(\theta)ZHD + m_w(\theta)ZWD \tag{2.6}$$

where $m_h(\theta)$ and $m_w(\theta)$ are mapping functions (inversely proportional to the sine of the slant path elevation angle θ) for the hydrostatic and wet components respectively. For elevation angles above 15° the hydrostatic and wet mapping functions are essentially equal allowing us to write,

$$STD = m_n(\theta)ZTD \tag{2.7}$$

where,

$$ZTD = ZHD + ZWD \tag{2.8}$$

is the Zenith Total Delay. The ZTD is determined from the measured STDs [58] and the ZWD is determined as,

$$ZWD = ZTD - ZHD \tag{2.9}$$

where the ZHD is a slowly varying quantity that can be estimated to a fraction of a millimeter via the so-called Saastamoinen model [46],

$$ZHD = \frac{0.00227768P_0}{1 - 0.00266cos(2\phi) - 0.00028h_{ref}} \tag{2.10}$$

where P_0 is the surface pressure (in millibars), h_{ref} is the geodic height of the station (in meters) and ϕ is the station latitude. Given the ZWD, the Integrated Precipitable Water (IPW), which represents the depth of water in mm per square meter that the column of atmosphere directly over the GPS receiver is holding in the vapor state, is given by [9],

$$IPW = \Pi ZWD \tag{2.11}$$

where,

$$\Pi = \frac{10^6}{461,525(\frac{373,900}{T_m} + 22.1)} \tag{2.12}$$

and

$$T_m = 70.2 + 0.72T_0 \tag{2.13}$$

with T_0 the surface temperature at the GPS receiver location.

2.2.2 GAMIT Software

There are a number of available software packages that one can use for IPW estimation. The one we use in this work is the GAMIT software package developed at MIT [26]. In addition to providing high-precision position analysis, GAMIT has routines for GPS-Met IPW estimation. The routines use the double differencing technique and tropospheric delay models described previously.

Inputs to GAMIT are RINEX (Receiver Independent Exchange Format [25]) navigation files containing the receiver and satellite clock offsets, observation files containing the code and carrier phase measurements for each slant path at 30 second sample rate, and for the GPS-Met application, meteorological files (from a collocated or nearby weather station) containing the surface pressure, temperature, and relative humidity data mapped to the GPS site location. The satellite orbital parameters giving precise orbit information for each satellite are also an input. These are generated

by the IGS (International GNSS services) [19] and are automatically downloaded by GAMIT for the analysis time period in order to correct for orbit errors. Reference stations with baselines of more than 500 km from the GPS stations at which IPW is desired are chosen to satisfy the double-differencing condition that the reference site is uncorrelated from the IPW sites [42]. GPS-Met software, such as GAMIT, has been validated to produce precipitable water vapor measurements of better than 2 mm RMS [20].

2.2.3 IPW Normalization

The GAMIT software produces point estimates of IPW. To obtain the spatial distribution of IPW, we need to combine IPW values from multiple GPS receivers. Before we can do so, however, we first need to normalize the IPW values from the different GPS stations. IPW is the amount of water vapor in a vertical column over the site. Because the amount of water vapor the atmosphere can hold is a function of temperature and temperature generally decreases with altitude, IPW will consequently also depend on the altitude of the station. Since IPW depends on temperature, IPW will also change with season so that an IPW value corresponding to low humidity at summer temperatures might be saturated at winter temperatures. Moreover, IPW by itself does not tell the level of saturation, since again that depends on the daily temperature. To account for station height differences, seasonal (monthly) variations, and to identify anomalously high or low IPW values, we normalize the IPW values by the monthly mean and standard deviation before we combine them as follows,

$$NIPW = \frac{IPW - \mu_{ij}}{\sigma_{ij}} \; \forall \; i, j \tag{2.14}$$

here the subscripts i and j are for the station and month respectively and μ_{ij} and σ_{ij} are the mean and standard deviations respectively. In the literature NIPW is

termed the standardized anomaly of precipitable water vapor and is a common way of presenting precipitable water information [24].

2.2.4 Multiquadric Interpolation

To obtain the spatial distribution of NIPW we interpolate from a set of point measurements. The paper [56] describes a number of methods for interpolating geophysical data. The method we chose is the multiquadric method. This method was chosen because it has been shown to perform nearly as well as the more common Kringing technique [40] but without the need for historical data.

Similar to Kringing, the multiquadric method is a weighted linear interpolation method where the estimate h_0 for any grid point (x_0, y_0) is given by,

$$h_0 = \sum_{j=1}^{n} w_j \cdot h_j \tag{2.15}$$

where h_j is the observed NIPW at point (x_j, y_j) and w_j is the weight giving the influence of h_j in determining h_0.

In the multiquadric method the weights w_j are calculated from the matrix of distances between the observed points (x_j, y_j) and the distances between the interpolated point and each observed point as follows. We start by expressing the observed points as a weighted linear combination of the distances between the observation points,

$$h_j = \sum_{i=1}^{n} c_i \cdot d_{ji} \ \forall j = 1, 2..n \tag{2.16}$$

where d_{ij} is the distance between GPS site i and GPS site j and h_j is the observed NIPW measurement at that GPS station. The coefficients c_i are then determined by,

$$c_i = \sum_{j=1}^{n} \delta_{ij} \cdot h_j \ \forall i = 1..n \tag{2.17}$$

where δ_{ij} is an element of the inverse of the $n \times n$ distance matrix $d_{ij}, j = 1..n \; and \; i = 1..n$. Given the c_i we can thus write,

$$\begin{aligned} h_0 &= \sum_{i=1}^{n} c_i \cdot d_{0i} \\ &= \sum_{i=1}^{n} [\sum_{j=1}^{n} \delta_{ij} \cdot h_j] \cdot d_{0i} \\ &= \sum_{j=1}^{n} [\sum_{i=1}^{n} \delta_{ij} \cdot d_{0i}] \cdot h_j \end{aligned}$$

or

$$w_j = \sum_{i=1}^{n} \delta_{ij} \cdot d_{0i} \tag{2.18}$$

2.2.5 Time complexity of Multiquadric interpolation

The multiquadric interpolation works in two steps. The first step involves computing the weights for each point on the grid that we are interpolating onto as a function of the influence of the N points where the measurements are made. The second step involves computing the interpolated estimate at that grid point as a function of the N grid points. Thus for a given number of stations N where the measurements are made the weights or influence of each one of these points on each point in the grid needs to be computed only once and can be stored.

To analyze the time complexity of the multiquadric interpolation algorithm, let us start by defining a spatial grid of size $m \times m$ with M number of points and measurements made at N points. To compute the distance matrix would require $O(N^2)$ and to invert this matrix would be $O(N^3)$.

To interpolate each of these points to the $m \times m$ domain where each point is a weighted linear combination of each element in the inverse distance matrix we obtain

the time complexity of $O(N^2)$. Thus the total time complexity to compute the weights is $O(N^2 + N^3 + MN^2)$.

The storage required for the weights is $O(MN)$. In our case we have $N = 44$ stations or measured points and we interpolate to a $M = 100 \times 100$ grid. The time complexity to compute the weights can be distributed in a way that the M points can be subdivided threads, and each thread can concurrently compute the weights of the subdivided points. Finally the points can be concatenated to obtain the weight matrix of the entire spatial grid.

Once we compute and store the matrix, the measurement must be estimated at each grid point. As the estimate at each grid point is a linear combination of the measurements the time complexity to compute the entire field is $O(MN)$ given that the weights are calculated.

2.2.6 IPW Field Generation

In light of the above discussion, we can summarize our method for obtaining fields of NIPW as follows. Every 30-minutes we do the following,

1. Obtain GPS navigation and observation files from N GPS sites distributed throughout the geographical region of interest;

2. Obtain Pressure(P), Temperature(T), Relative Humidity(RH) for each GPS station;

3. Put the P, T, RH data in the required RINEX format;

4. Feed the GPS and meteorological RINEX files into GAMIT to get IPW values for each station for the current 30-minute interval;

5. Normalize the IPW values based on the average and standard deviation for the given station and given month;

6. Apply multiquadric interpolation to obtain the NIPW field for the current 30-minute interval.

2.3 Measuring Rainfall

The primary instrument for obtaining spatial-temporal fields of precipitation is the ground-based weather radar. There are a number of different weather radar systems in operation in the U.S. These include the 160 long-range WSR-88D, Next Generation Weather Radars (NEXRADs) operated by the U.S. National Weather Service (NWS) for real-time weather monitoring and short-term forecasting and the 45 Terminal Doppler Weather Radars (TDWR) operated by the Federal Aviation Administration (FAA) to provide high-update rate, high-resolution weather and wind data at key major airports. In addition to these, increasing numbers of television stations have their own weather radars, and there are a variety of research weather radars, such as the network of small X-band radars operated by the University of Massachusetts, Center for Collaborative Adaptive Sensing of the Atmosphere (CASA) in the Dallas-Fort-Worth metroplex region.

As a weather radar scans in azimuth, it sends out a narrow, pencil beam of microwave pulse energy and then samples the return echo. This partitions the space around the radar into resolution volumes or voxels (volume elements). Voxels have the shape of a disk on its side: the diameter of the disk determined by the radar's beam-width; the thickness by the radar's gate spacing. The size V of a voxel in cubic meters thus varies with the square of the range from the radar according to,

$$V = G \cdot \pi \cdot tan^2(\theta/2) \cdot R^2 \tag{2.19}$$

where G is the gate spacing in meters, θ is the beam width in radians, and R is the range from the radar to the voxel in meters. For NEXRAD, with its 1 degree beam and 250 meter gate spacing, this leads to voxel with volumes that are roughly 6

million cubic meters at 10 km from the radar to 3 billion cubic meters at the radar's maximum (Doppler) range of 230 km.

Under assumptions that the voxels are completely filled with liquid hydrometeors (raindrops) and that the hydrometeors are small relative to the radar's wavelength (Rayleigh scattering), weather radars measure the echo from each voxel to infer properties of the hydrometers in the voxel (their density, size, radial (Doppler) motion, (Polarimetric) shape asymmetry, and so on) [13] [18]. To sample the complete volume around the radar, multiple 360 degree sweeps in azimuth are performed, each at a different elevation tilt angle. Each such 360 degree sweep is termed a Plan Position Indication (PPI) scan, and the sequence PPIs starting at the radar's lowest elevation tilt angle and working up to the radar's highest tilt angle that are performed to cover the volume is called the Volume Coverage Pattern (VCP). The NEXRAD lowest elevation tilt angle is 0.5 degrees, and the standard convective weather VCP has a temporal update rate of approximately 5-minutes between revisits of the 0.5 degree lowest elevation PPI.

The main weather radar product, reflectivity (Z in mm^6/m^3), is defined as the 6-th moment of the drop size distribution [18].

$$Z = \int_0^\infty N(D) \cdot D^6 dD \tag{2.20}$$

where D is the (volume equivalent spherical) drop size (diameter in mm) and $N(D)$ is the drop size distribution given by (Marshall-Palmer model) [18].

$$N(D) = 8000 \exp(-4.1 R^{-0.21} \Lambda D) \tag{2.21}$$

where R is the rain rate (in mm/hr).

Plugging (2.21), into (2.20) one can obtain an expressions relating reflectivity to median drop size [18],

$$Z = 642D_0^7 \tag{2.22}$$

and rain rate,

$$Z = 297R^{1.47} \tag{2.23}$$

Using the above relationships, Table 2.1 relates reflectivity (in dBZ = $10log_{10}Z$) to median drop size D_0 (in mm) and rain rate R (in mm/hr)[6]. Noting that cloud droplets

Table 2.1: Reflectivity as a function of median drop size diameter and rainfall rate.

dBZ	D_0 (mm)	R (mm/hr)
-42	0.1	-
0	0.4	-
10	0.6	0.1
20	0.8	0.5
30	1.1	2.3
40	1.5	11.0
50	2.1	52.0

average $12\mu m = 0.012mm$, and that even the high-powered NEXRAD only has -21dBZ sensitivity at 10km (estimated from the WSR-88D specification of -7.5 dBZ at 50 km[7] and the fact that minimum detectable reflectivity is proportional to range squared), we see that weather radars generally cannot detect clouds or uncondensed water vapor, but can only detect active precipitation. We also remark that although dual-polarimetric weather radar, such as the upgraded NEXRAD, do provide a rainfall rate product[8], in this thesis we will use reflectivity as a proxy for rainfall rate with 24dBZ and above indicating rainfall and the dBZ value reflecting rainfall intensity.

[6]We remark that the Z-R relationship in equation 2.23 is only one of many such relationships. Others include the famous 1948 $Z = 200R^{1.6}$ Marshall-Palmer relationship [36] and the $Z = 300R^{1.4}$ relationship used by the NWS as the default Z-R relationship for the WSR-88D radar network http://www.srh.noaa.gov/tlh/?n=research-zrpaper

[7]https://www.roc.noaa.gov/WSR88D/PublicDocs/NTR96.pdf

[8]https://www.ncdc.noaa.gov/data-access/radar-data/nexrad-products

Specifically, we will use the NEXRAD 0.5 degree elevation reflectivity PPI as an indicator of precipitation activity and intensity at ground level.[9]

[9]To say that the 0.5 degree NEXRAD reflectivity product gives the rainfall at ground level ignores the fact that the earth is curved while radar beams travel in essentially straight lines leading to an increase in radar beam height above ground level with distance from the radar. In particular, we are ignoring the fact that even at its lowest tilt of 0.5 degrees, the bottom of a NEXRAD beam is some 3000 meters (10,000 feet) above ground level at the radar's maximum (Doppler) range of 230 km.

CHAPTER 3

DEVELOPING A BENCHMARK DATA SET

The domain of study considered in this thesis is the Dallas Fort-Worth area in Texas. Part of the infamous U.S. "tornado alley", DFW spring and summer weather is dominated by convective thunderstorms that move in lines generally from west to east through the region. We choose the DFW region because we understand its climatology (through CASA's more than 15 years of operating networks of weather radars, first in Oklahoma and now in DFW) and because the DFW region has a high density of GPS receivers and weather stations whose data are publicly available on-line for us to repurpose as a GPS-Met network. We also select this region as our domain of study because it has a fairly flat topography with no lakes, rivers or oceans surrounding it. This makes it easier for us to study the effects of water vapor on precipitation. The flat topography also allows us to make the assumption that one particular region in the domain behaves like an another one in the domain. We use this assumption in training our machine learning algorithm by uniformly sampling points in the domain and applying the trained model to the rest of the points.

Three years of storm data are analyzed (2014 - 2016) during the spring summer storm season in the DFW region which consists of days between May to August. We take as the center of our region the NWS KFWS NEXRAD radar in Fort-Worth Texas[1]. Within the 230 km coverage range of the radar we identified 44 Regional Reference Points, i.e., high performance dual-frequency GPS receivers. These GPS

[1]http://radar.weather.gov/radar.php?rid=fws

receivers, which are operated by the Texas Dept. of Transportation (TxDOT)[2], were deployed to provide precise position information for Geodetic studies. As such these GPS receivers do not have collocated weather stations. For the weather data (surface temperature, pressure, and relative humidity) required for IPW estimation, we used data from the network of Automated Surface Observation Stations (ASOS) operated by NOAA NWS[3]. The ASOS network provides surface meteorological variables at 5 minute resolutions. Figure 3.1 shows the relative locations of the GPS receivers and ASOS stations within the 230 km coverage range of the KFWS radar. Table A.1 in the Appendix gives the locations and heights of the the GPS receivers as well as the location and height of the ASOS that is closest to the GPS receiver.

GPS RINEX files of code range and carrier phase for the storm periods of 2014 – 2016 are downloaded for each of the 44 stations from databases maintained by SOPAC[4] [11] and CORS[5][51]. The meteorological data from the ASOS stations are obtained at 30 minute intervals for the both years 2014 and 2015 from Teresa Vanhove of UCAR[6] and the data for 2016 was downloaded from NCDC. The KFWS 0.5 degree reflectivity product is downloaded from NCDC[7]. For the long baseline stations needed for double differencing, we chose the four stations: AC20 in Girdwood Alaska, CONZ in Concepcion Chile, P019 in Fairfield, Idaho, and UNBJ at the University of New Brunswick, Canada. The closest of these stations, P019 is approximately 1500 km from the KFWS radar that defines the center of our DFW GPS network. These

[2]http://www.txdot.gov/inside-txdot/division/information-technology/gps.html

[3]http://www.nws.noaa.gov/asos/

[4]http://sopac.ucsd.edu/

[5]http://www.ngs.noaa.gov/

[6]http://www.suominet.ucar.edu/

[7]https://www.ncdc.noaa.gov/data-access/radar-data

stations were chosen to ensure that there was always at least one satellite that had a view of all of the TxDOT GPS stations and one of the baseline stations.

To get the meteorological data for a particular TxDOT GPS station, we found the closest ASOS site and used the equations from [5] to interpolate the surface temperature (T), pressure (P), and relative humidity (RH) data from the MSL height of ASOS site to the MSL height of the GPS station,

$$P_{SL} = P_{MSL} \cdot (1 - 2.26 \cdot 10^{-5} \cdot H)^{5.225} \tag{3.1}$$

$$T_{SL} = T_{MSL} - 0.0065 \cdot H \tag{3.2}$$

$$RH_{SL} = \frac{RH_{MSL}}{e^{-0.0006396 \cdot H}} \tag{3.3}$$

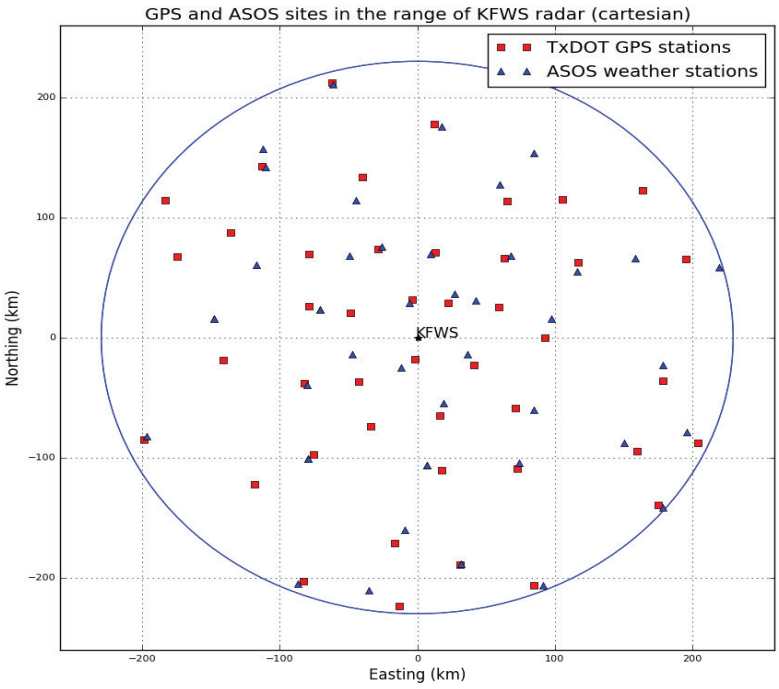

Figure 3.1: GPS and ASOS stations within the KFWS 230 km coverage range.

In the above equation, P_{SL} is the pressure at the station level, P_{MSL} is the pressure at mean sea level, and H is the height of the station (in meters MSL). We developed a Python script to convert the meteorological values from the ASOS file format to the met RINEX format required for GAMIT.

Processing for IPW is done using the GAMIT software using the GPS and met RINEX to obtain measurements of 30 minute intervals for each GPS station for the duration of our study (spring summer storm season 2014 – 2016). The mean and standard deviation of IPW for each GPS station are calculated for each month, and the IPW values normalized to obtain the standardized anomaly of precipitable water, which we denote NIPW. The NIPW values were then mapped to a 300 km by 300 km, 3km resolution grid centered on the KFWS radar using the multiquadric interpolation method as previously described in Chapter 2.

The data processing was done on a Ubuntu server where GAMIT is installed. The server has 16 CPUs with 4 cores each. The data processing was performed in a manner that optimized the computation resources of the server thus allowing many days to be processed at the same time. Specifically, the 44 GPS stations were broken down into 4 different sub networks where the data processing for the networks could run in parallel. CSV files containing the IPW estimates for the spring-summer storm season from the 44 GPS stations are available online [8].

Regarding the NEXRAD reflectivity data, this was first down-sampled from its native 5-minute update rate to the 30-minute NIPW update interval. We then performed a polar to Cartesian coordinate conversion to map the reflectivity field to the same 3 km grid using the WGS84 mapping equations[9] to the same resolution as used for NIPW. The result is two fields of 10,000 pixels each updated every 30-

[8] http://emmy9.casa.umass.edu/gpsmet/DFWnet/

[9] http://williams.best.vwh.net/avform.htm

minutes, one of NIPW (a measure of atmospheric water vapor) the other of reflectivity (a measure of rainfall).

3.1 Experimental Data Set

To understand the role of NIPW in predicting precipitation we chose a set of storm dates and visually inspect the relationship between NIPW and reflectivity. We denote these storm dates as "weather anomaly" days. The criteria for selecting these dates are to see if 30 or more NIPW values from a particular day had a standard deviation of greater than 2 from the mean amongst all of the NIPW values from that day. In other words consider a matrix of 44×48 values of NIPW (44 stations where IPW is sampled at 30 minute intervals for the entire day) values, a day is termed as a weather anomaly day if there are more than 30 values which have a standard deviation of greater than 2. The list of days were further padded by taking into consideration one day before and one day after the weather anomaly days to account for storm build up and decay. Table 3.1 lists the weather anomaly days for the years 2014 – 2016. Interestingly almost all of the weather anomaly dates were days during which severe weather occurred and all severe storm reports reported by NOAA[10] for the DFW region are included in the "weather anomaly" days.

3.1.1 Data Preprocessing

One issue we encountered is that we would occasionally get "bad" IPW estimates from GAMIT. For instance we found numerous instances where there is an abrupt increase or decrease in the IPW value at a station followed by smooth measurements of IPW values thereafter. This did not make any sense as the atmosphere can not lose so much moisture (energy) in such a short period of time. We believe these bad IPW estimates were due to missing or corrupt carrier phase and pseudo range mea-

[10]http://www.spc.noaa.gov/climo/reports/today.html

surements files and/or missing or corrupt orbit files. The bad values may also be due to the fact that the zenith measurements are averaged from only 3-4 satellites rather than the 8-12 satellites. Some of these bad values, for example, had a greater level of uncertainty associated with them as reported by GAMIT (i.e standard deviation of the estimate was greater than 2 mm for some of these bad values).

To account for bad IPW values, we wrote an algorithm to search the IPW sequences for times where the IPW value dropped by 50% or increased by 100% between consecutive values. The algorithm would then replace the bad value by linear interpolation of the IPW values to either side. This algorithm was run before finding the monthly mean and standard deviations of the stations and determining the weather anomaly days. Alternatively, these "bad values" could have been determined directly from GAMIT itself by considering a value bad when the uncertainty in the IPW estimate is greater than 2 mm. We did not use this technique as this did not account for all apparent bad values in our experiment.

Table 3.1: Weather anomaly days

Month	2014	2015	2016
May	8,12,23,25,31	9,10,19,20,23,24,29	26,31
June	13,18,19,22,25	15,16,17,18,21	1,12,13,28
July	15,16,17,18,24,28,29,30,31	3,7,8,22,31	3,4,5,25,26,27,28
August	11,16,17,18,19,29	1,19,20,21,25	-

3.2 Preliminary Data Analysis

To give a sense of typical IPW values in the DFW region and how these IPW values vary with station altitude, Figure 3.2 plots the monthly means for the 4 highest and 4 lowest GPS sites for the year of 2014. GPS station altitudes range from a low of 83.6 meters above sea level to 498.3 meters above sea level. Clear in the plot is the altitude dependence, with mean IPW inversely proportional to station altitude.

To see how IPW varies by season in the DFW region, Figure 3.3 plots histograms of the IPW values observed at the lowest (TXBX) and highest (TXC3) GPS sites. From the figure we can see the seasonal dependence with higher mean IPW values during the warmer months (April-September) and lower mean IPW values during the cooler months (October-March). This is due to the fact that during the summer of the warmer months the atmosphere can hold more water vapor than the colder months. Again we see the height dependence with the IPW of the highest station having generally lower IPW values than the lowest station.

These two plots illustrate why IPW values are typically normalized before they are combined to produce a field. Our approach of normalizing to obtain the number of standard deviations from the mean removes height and seasonal dependancies and also allows us to infer something about saturation level as many standard deviations above (below) the mean will typically indicate a saturated (dry) atmosphere respectively without a need to estimate relative humidities or dew points. This normalization technique also renders unnecessary the temperature measurement to determine the saturation levels of water vapor.

3.3 Visualizing the joint evolution of NIPW and radar reflectivity fields

As a first step in visualizing the correlation between the NIPW and the reflectivity fields, we plot the reflectivity fields on top of the NIPW fields as shown by Figure 3.5 – 3.7. We only show the reflectivity measurements with values greater than 30dBz (light rain and above). Browsing through these figures we see that there is some correlation between the NIPW field and the reflectivity field, where in the patches where there is a large peak of NIPW values, there is also high levels of precipitation. This suggests that the NIPW indicates where the storm will grow into and where it will decay and die out. It thus seems true that NIPW should help with the "Growth

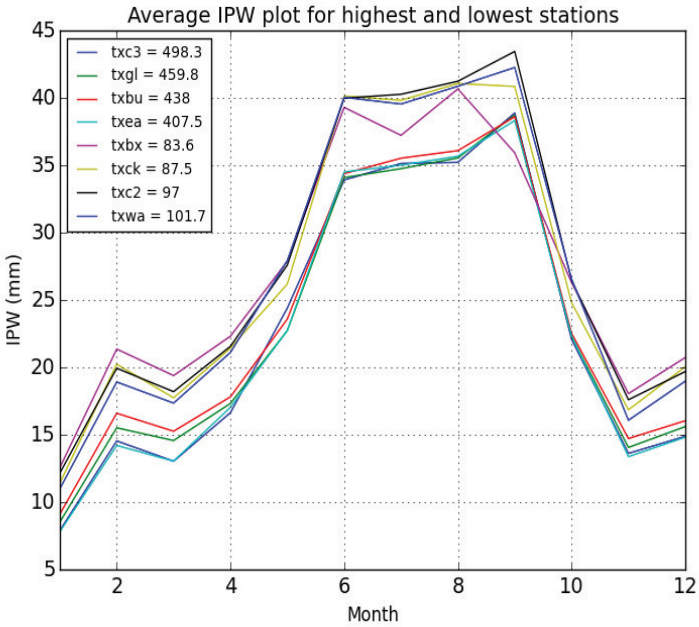

Figure 3.2: Mean IPW for each month for 4 highest and 4 lowest stations. The height is measured as the Geodic height in meters.

and Decay" problem that plagues radar extrapolation based nowcasting algorithms. In some cases we see that these patches of peak NIPW is leading the precipitation fields. We further analyze several of these cases in the following sections.

3.3.1 May 8th 2014

A severe storm occurred on May 8th 2014 which caused devastating floods and tornadoes in the DFW area. Figure 3.4 shows the reflectivity overlaid on the NIPW plot. The storm approaches the area as a front which originates from the North-West and moves across the domain in lines. The rain seems to follow the decreasing edge of the NIPW.

3.3.2 July 17th 2014

As discussed in Chapter 2 a stratiform cloud formation is one where the clouds are mostly stationary causing a heavy downpour while circulating in the area. July

17 2014 was one such case where there was no front but the clouds seemed to have been circulating in a single location. From Figure 3.5 we can see that the rain seems to be pulled along by the NIPW fields. We can also notice the higher than normal NIPW values with larger white patches.

3.3.3 August 29 2014

Similar to the storm on May 8th 2014 a storm on August 29 presented with fronts which originated from the North-West. From Figure 3.6 we see that the reflectivity fields rides on top of the NIPW fields. The large convergence of NIPW coincides well with the storm front. This convergence can be detected 1 - 2 hours before the storm appears in the region. We rely on this convergence of water vapor to be indicative of severe storms.

3.3.4 May 17th 2015

A similar pattern can also be seen on a storm on May 17th 2015 in Figure 3.7. It is interesting to note from this figure that there is a large build up of NIPW in the South-West corner of the domain but there is no indication of rain. This indicates that a large convergence of water vapor does not always lead to severe front across the region.

3.3.5 Concluding Summary

As we have seen from all of the different cases presented above, the common trend seen is that precipitation moves into the regions of growing NIPW and moves away from decreasing NIPW. This correlation seems to be obvious in some cases and not so obvious in other cases. We believe these correlations although present in only a few cases will be enough for a machine learning algorithm to infer the growth-and-decay component of a storm.

We explore this conjecture that since radar reflectivity can not determine growth and decay of a storm and convergence of NIPW indicates growth and decay which can aid in precipitation nowcasting.

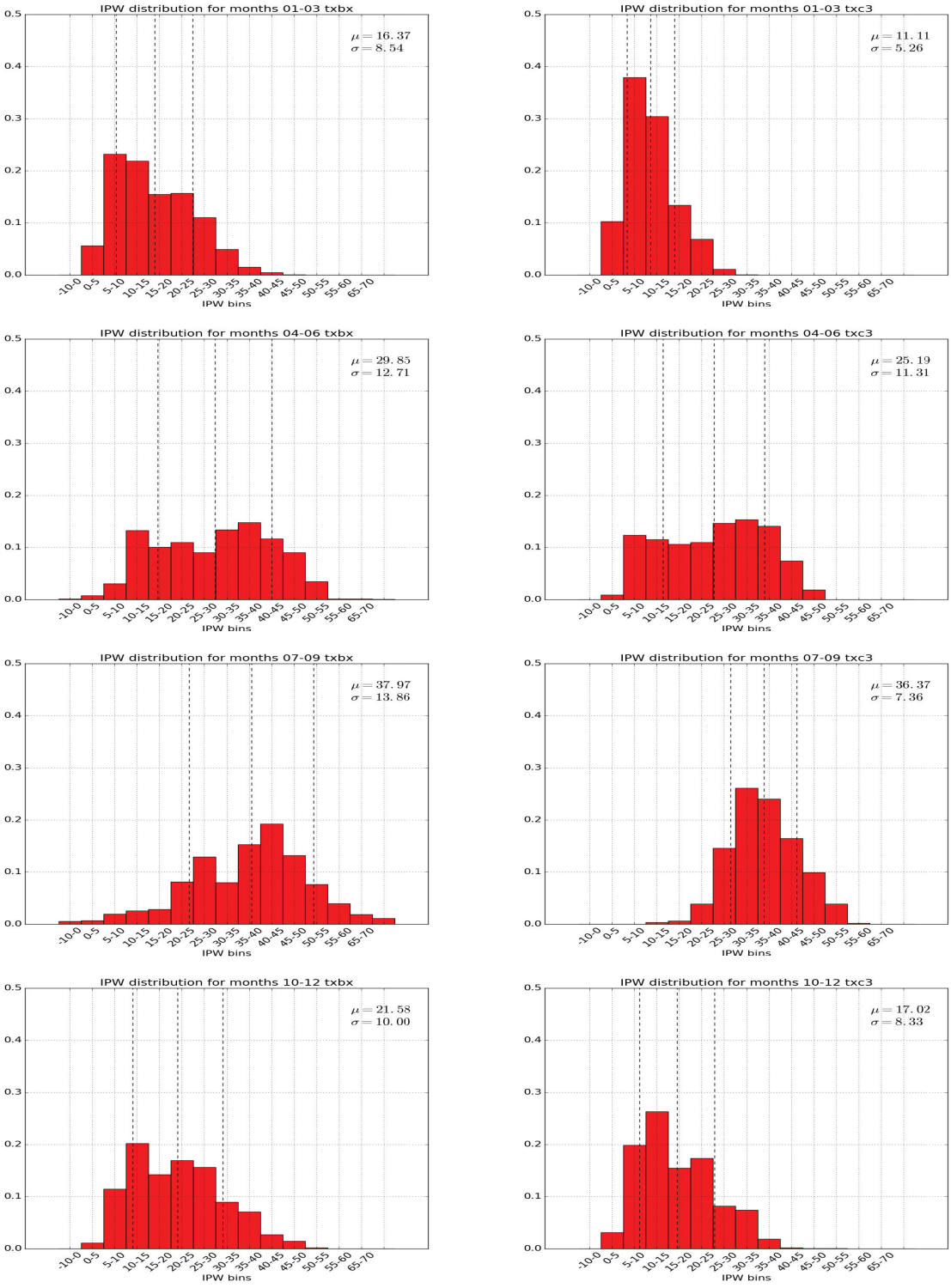

Figure 3.3: IPW histograms for each season left: lowest station TXBX at 110m MSL and right: highest station TXC3 at 524m MSL. The vertical bars indicate the mean and +/- standard deviations from the mean

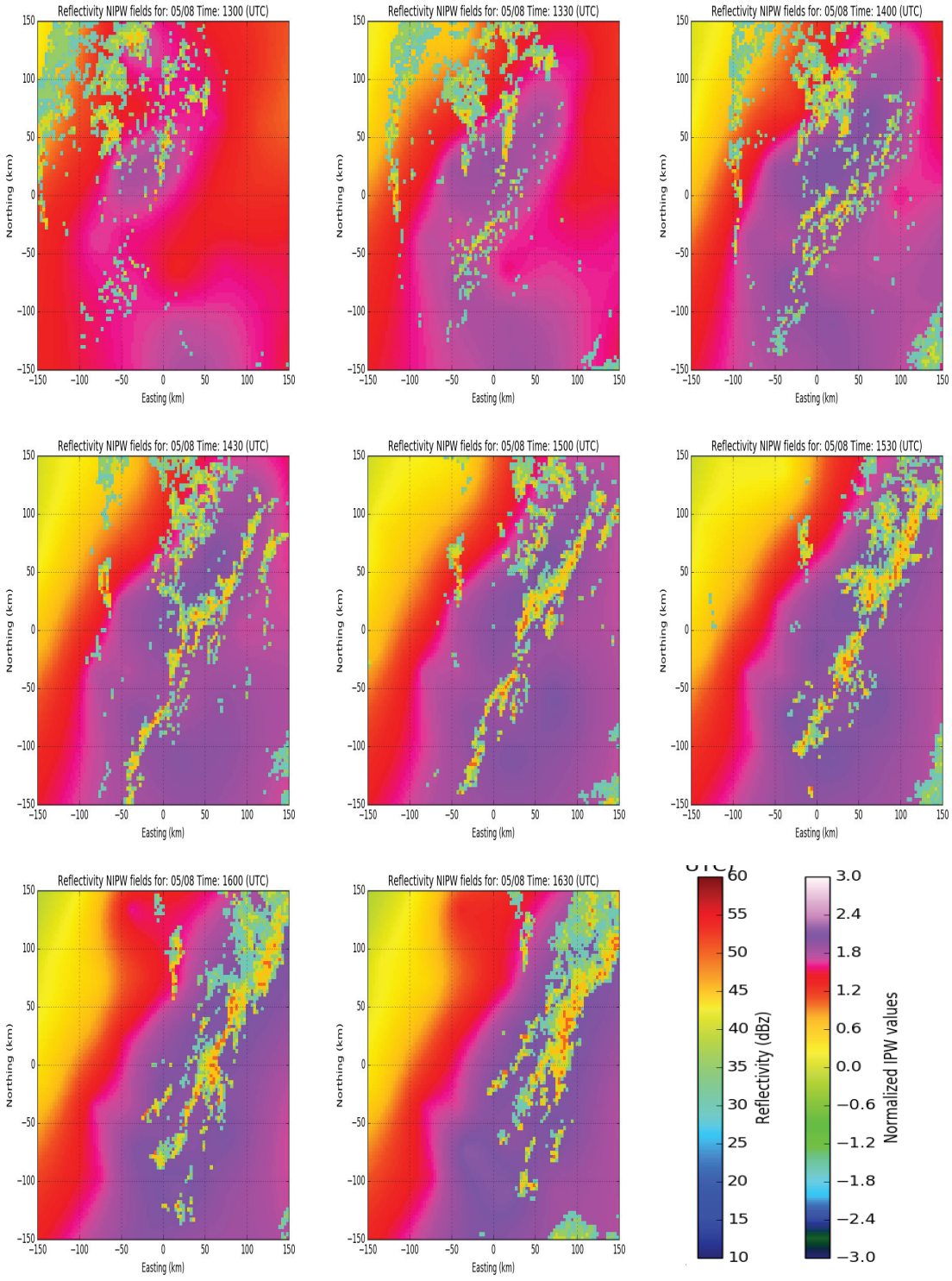

Figure 3.4: Reflectivity IPW fields for a storm on May 8th 2016. The correlation between NIPW and precipitation is clear in this case where we see precipitation is increasing at regions of high NIPW and decreasing or stopping at regions of decreasing NIPW.

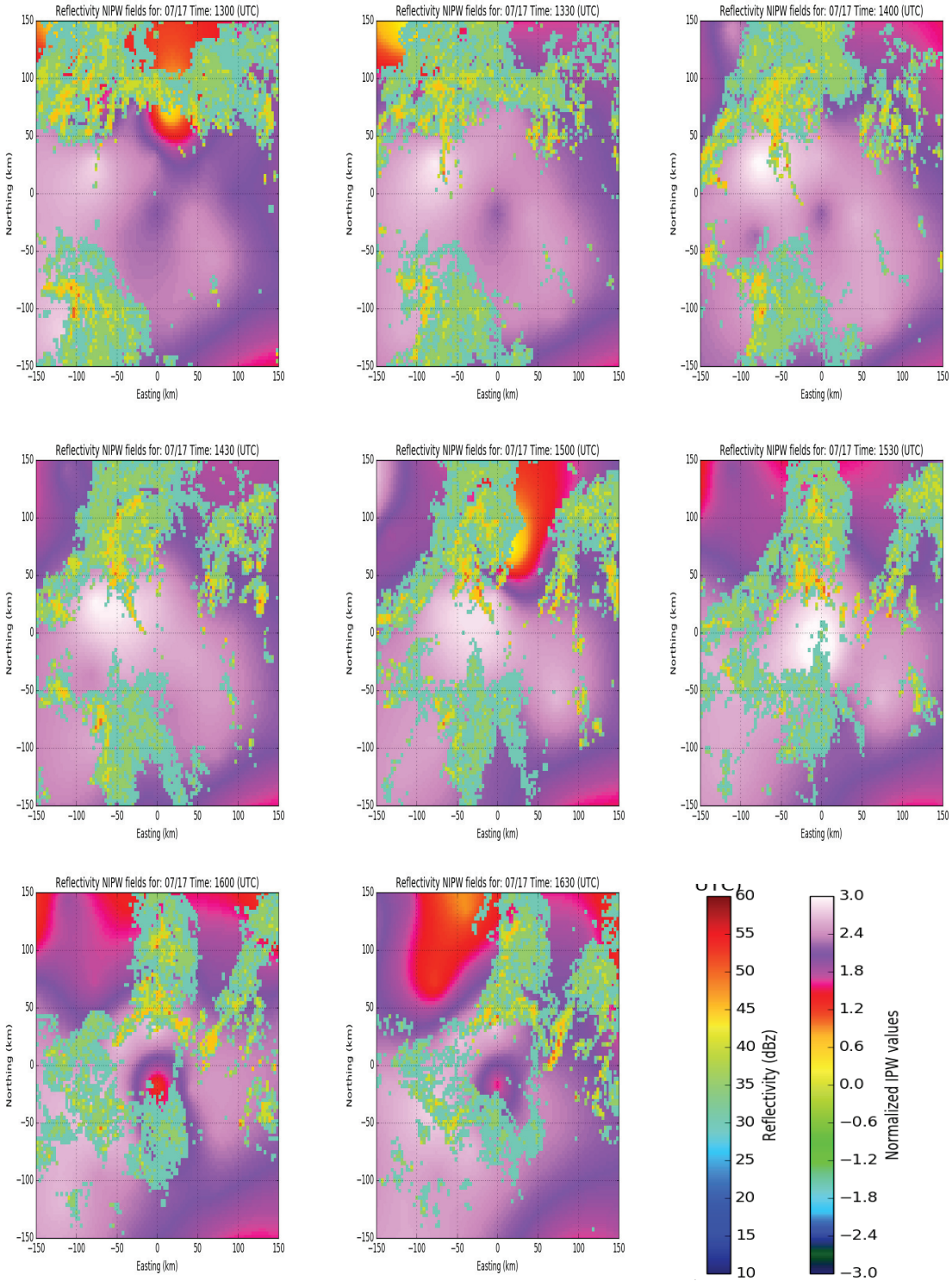

Figure 3.5: A strataform storm on July 17th 2014. The correlations seem to be less clear however we see that where there is gradients of NIPW there tends to be precipitation.

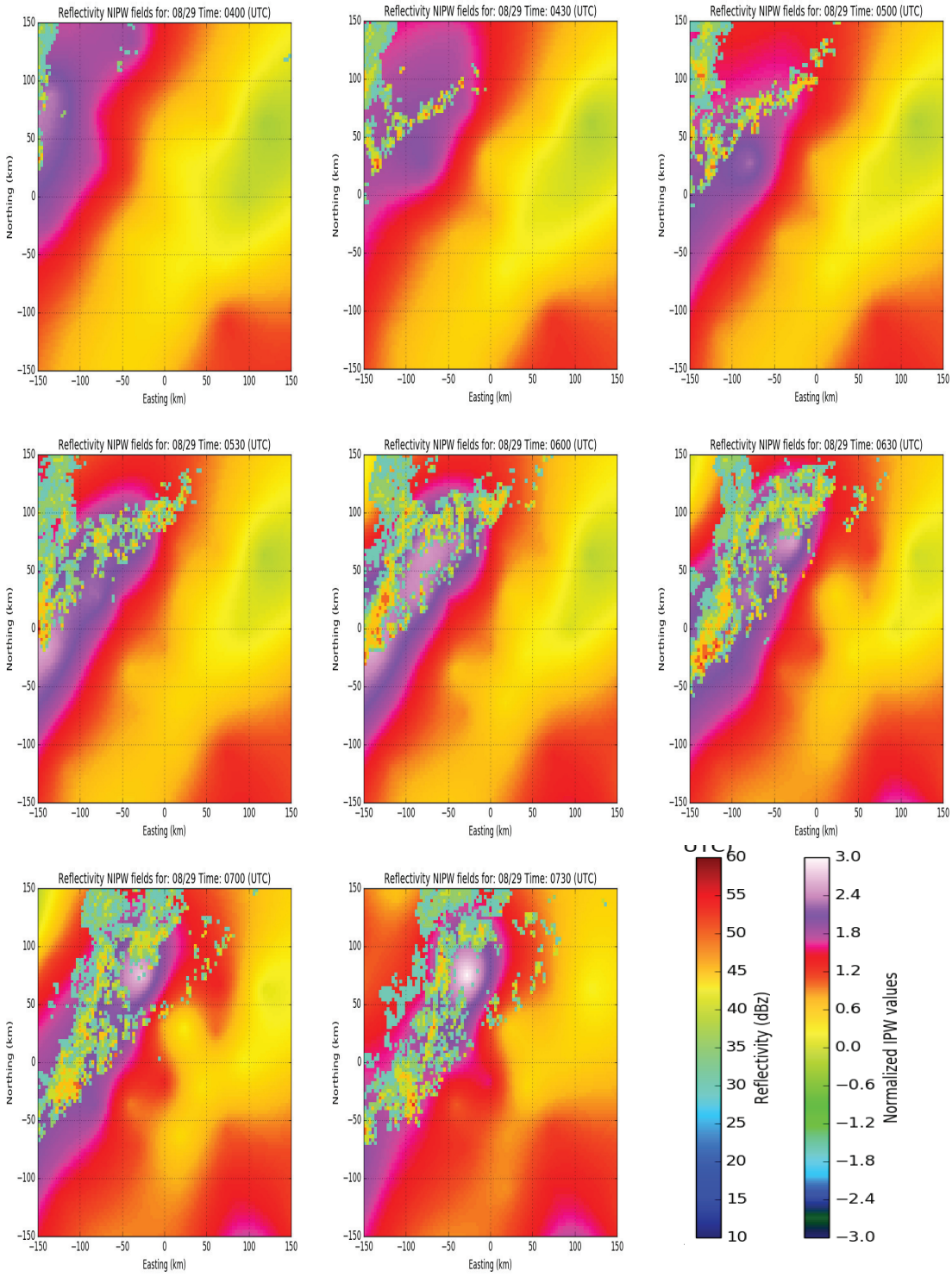

Figure 3.6: A convective front on August 29th 2014. Similar to May 8th, there is a correlation between the NIPW and precipitation where precipitation is riding atop or behind the convergence of NIPW

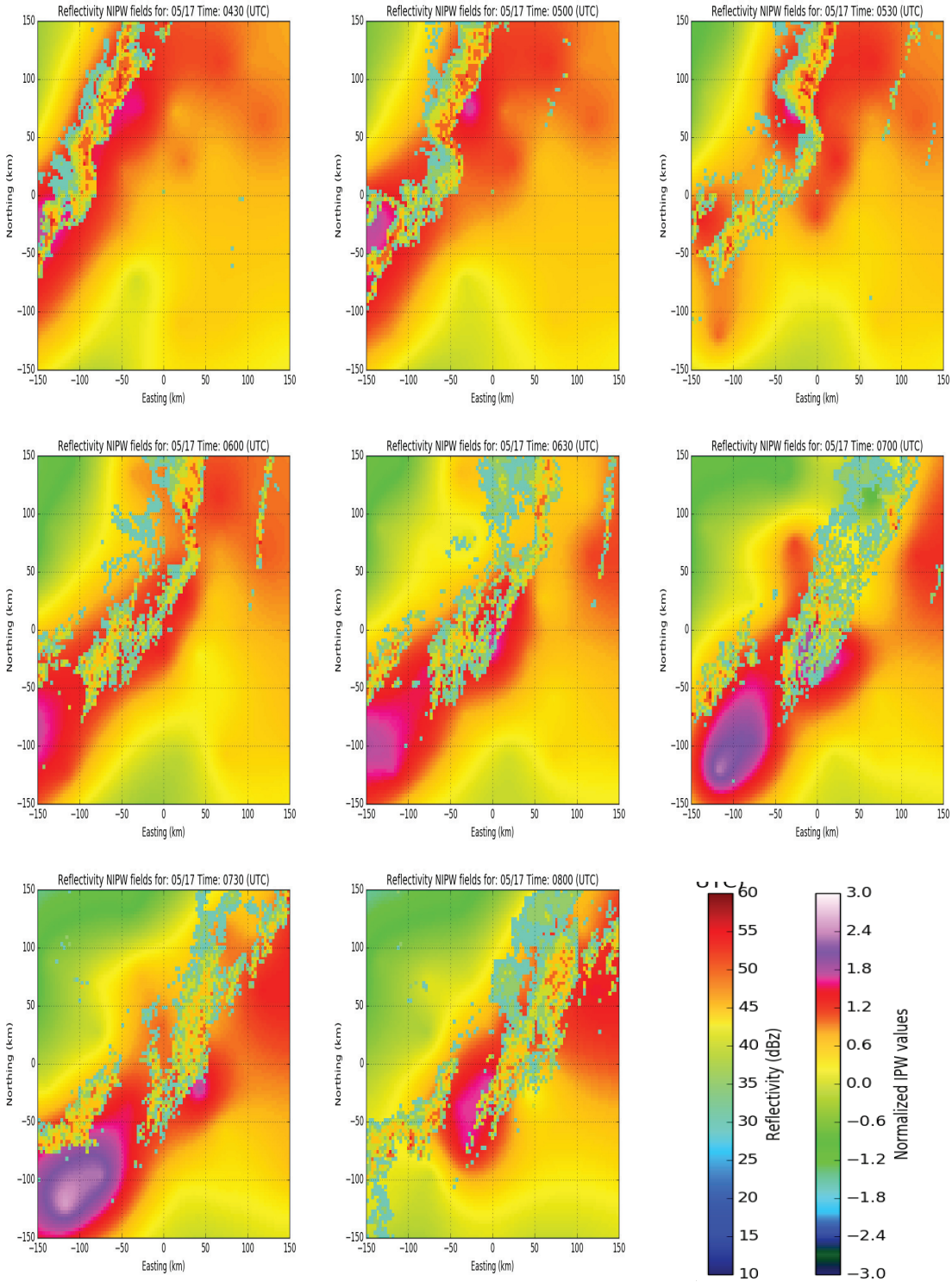

Figure 3.7: Storm on June 17th 2015. Here's another case where the correlations are not as immediately obvious - the general trend of rain riding slightly behind the peak in NIPW is there, but the NIPW does not trace as clear a frontal boundary as in the previous cases in Figure 3.4 and 3.6

CHAPTER 4

MACHINE LEARNING SOLUTION APPROACH

As seen from the previous chapter, visually there appears to be a spatiotemporal correlation between atmospheric water vapor and precipitation in that a concentration of the former tends to lead to the latter. To investigate this correlation and see if there is any precipitation nowcasting benefit in complementing radar reflectivity fields with NIPW fields we take a machine learning approach.

In this chapter we formulate our machine learning nowcasting problem and then describe the machine learning algorithms that we will explore for its solution.

4.1 Machine Learning Nowcasting Problem

For a machine learning approach, we can phrase our problem as follows: We have two evolving images or videos, one of the reflectivity field showing where it is raining now and another of the NIPW field showing the concentrations of precipitable water vapor that could potentially become rain, and our goal is to predict the reflectivity field 1-hour in the future. The two fields are sampled at the same spatial and temporal resolution (30 minute temporal resolution, 3 km spatial resolution over a 300×300 sq-km domain) to give, at each sample time t, two 100×100 matrices, one for each field, where each element in the matrix represents the intensity value of the corresponding variable at a particular grid location on the domain. Let us denote the NIPW field and the reflectivity field at a given time step t as X_t^{ipw} and $X_t^{refl} \in \mathbb{R}^{100 \times 100}$, respectively.

In order to find the pattern between a history of reflectivity and NIPW fields up to time t and the reflectivity field $t + 2$ time steps (1-hour) in the future, we need,

in principle, to find the conditional probability distribution $P(X_{t+2}^{refl}|X_k^{refl}, X_k^{ipw}, k = t-T, \ldots, t)$, where T is the amount of past history used for the prediction. Rather than attempting to develop a physics-based model for the conditional probability distribution, as traditional nowcasting algorithms usually do, we will attempt to use supervised machine to learn the distribution function.

4.2 Supervised Machine Learning

Supervised machine learning involves a labeled data set where each instance or example is a pair $(\mathbf{x}_i, y_i)$ where the vector $\mathbf{x}_i = (x_1, x_2, x_3, ...x_D)$ consists of reals $x_j \in \mathbb{R}$ $(j = 1, \ldots, D)$ or categorical values $x_j \in X_k$ $(j = 1, \ldots, D)$ and y can either belong to a finite set $c_1, c_2, c_3, ..., c_n$ as in case of classification or can take a real value $y \in \mathbb{R}$ as in the case of a regression problem. If we thus have N labeled examples of the form $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$, $i = 1, \ldots, N$ we can write the input matrix as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and the output as a column vector $\mathbf{Y} \in \mathbb{R}^{N \times 1}$. From this labeled data, supervised machine learning is a family of techniques for finding a functional mapping $f : \mathbf{x} \rightarrow y$.

In machine learning, the function f is obtained by adjusting a set of adaptable parameters $\mathbf{w}$, where the parameters are adjusted to minimize the discrepancy between the predicted value $f_i(\mathbf{x})$ and the associated truth value y_i. For a regression problems, this usually involves minimizing the mean squared error (MSE) given by $1/N \sum_{i=1}^{N} (y_i - f(\mathbf{x_i}; \mathbf{w}))^2$. In practice, the labeled data set is divided into two subsets – a *training set* used to learn the functional mapping f and a *testing or validation* set used to evaluate the ability of the learned function to generalize to new examples not seen during training.

4.3 Ensemble Methods

Ensemble methods rely on the average prediction of many different models rather than single model [17]. The different models can be thought of as a panel of experts each with different knowledge about the data and the principle is that one can obtain a better decision by combining the ensemble of decisions from the individual experts.The philosophy behind ensemble methods is a desire to balance the so-called bias/variance tradeoff. To explain, let the true mapping from $\mathbf{x}$ to y be f and our machine learning approximation be $\hat{f}$. Given an arbitrary input output pair $(\mathbf{x}, y)$ we want to understand the mean-squared prediction error,

$$E[(y - \hat{f}(\mathbf{x}))^2] \tag{4.1}$$

where the expectation is over the dataset $\Omega = \{(\mathbf{x}_i, y_i), i = 1, 2, \ldots\}$. Through some manipulation we can write the above as (c.f [34]),

$$E[(y - \hat{f}(\mathbf{x}))^2] = Bias[\hat{f}(\mathbf{x})]^2 + Variance[\hat{f}(\mathbf{x})] \tag{4.2}$$

where,

$$Bias[\hat{f}(\mathbf{x})] = E[\hat{f}(\mathbf{x} - f(\mathbf{x})]$$
$$Variance[\hat{f}(\mathbf{x})] = E[\hat{f}(\mathbf{x})^2] - E[\hat{f}(\mathbf{x})]^2$$

A good result is a small mean-squared error. This implies we want a system that has both small bias and small variance. We can get small bias by fitting the training data very accurately, i.e., we get small bias when we "overfit" the training data. On the other hand, we get small variance when we generalize well to data points not seen during training, which is generally not the case for a model that overfits its training

data. This difficulty in getting a system that gives good training performance (has low bias) and good generalization (has low variance) is the essence of the bias-variance tradeoff. The power of ensemble methods is that it can be shown that if we train many models $\hat{f}$ on different randomly selected (with replacement) subsets of Ω such that each model has low bias but high variance.(i.e., overfits its training set), then by averaging the model outputs we can reduce the variance while retaining the individual models' low bias [34].

4.3.1 Random Forests

One of the more popular ensemble algorithms is the random forest algorithm [12]. Each model in a random forest is a decision tree otherwise known as a Classification and Regression Tree (CART). Decision trees are ideal candidates for the individual models in the random forest as they are able to capture non-linear patterns in the data. Trees are also good candidates because they are "interpretable" in the sense that one can determine the importance of each feature in the input vector $\mathbf{x}$ as it relates to the output of the tree.

Given a set of training examples $(\mathbf{x}, y)$, a decision tree learns by recursively finding the optimal variable $x_d \in \mathbf{x}$ to split (branch) on and optimal splitting threshold t_d using a greedy heuristic that tries to maximize the training accuracy. In a geometrical sense the decision tree partitions the D dimensional input space into smaller regions such that each region contains the maximum number of instances that belong to a single class as shown in Figure 4.1 (left). An example is thus classified by traversing through the tree as shown by 4.1 (right).

Let us define p_{km} to be the proportion of data instances from the k^{th} class that are assigned to the m^{th} region. In terms of these definitions, CART seeks the variable x_d and split threshold t_d that maximizes the so-called Genie index,

$$C_{GI} = \sum_{k=1}^{K} p_{km}(1 - p_{km}) \tag{4.3}$$

CART maximizes the Genie index criteria by exhaustive search over all variables and

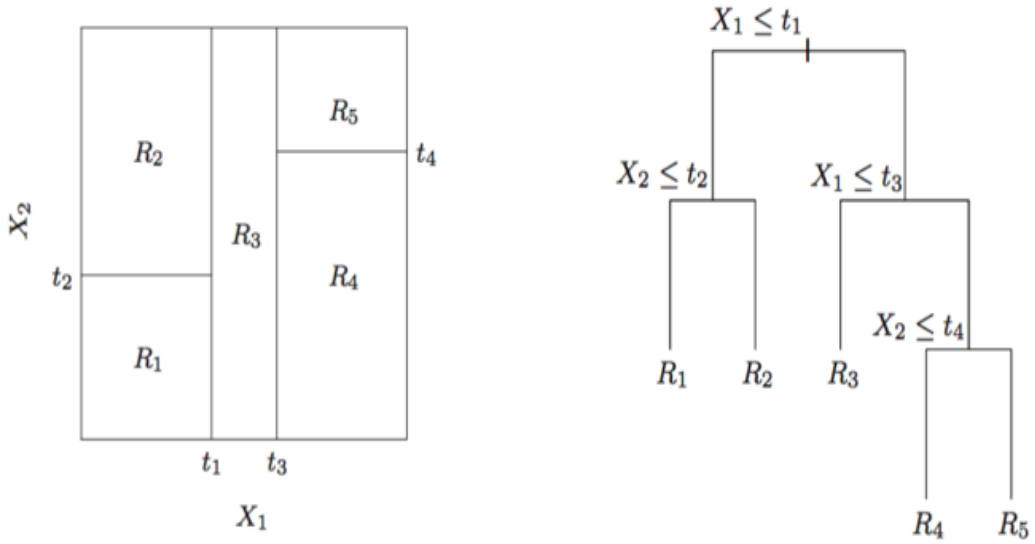

Figure 4.1: Decision Tree. Left: variables and thresholds for two dimensional variable **x**. Right: Representation of the decision tree corresponding to the split and thresholds made by the figure on the left. Figure from [23].

all different possible threshold values. The search process continues until all terminal nodes or leaves are pure[1] or until a pre-determined maximum tree depth is reached. It must be noted that the greater the maximum tree depth the more likely it is the model will overfit the training data. Each decision tree is grown on a randomly bootstrapped subset of the entire data set Ω. Moreover, each tree is further randomized in that each one is grown using a random subset of the features from the input vector **x**. This use of randomly selected features from randomly selected data sets results in a "forest" of decorrelated trees. In this fashion an ensemble of low bias, high variance models is obtained. The ensemble prediction of the random forest is obtained for the

[1]A leaf is said to be pure when all the data cases represented by that leaf are of the same class

regression case by taking the average prediction of each of the individual models and in the classification case by taking the average of the probabilities for each class (c.f [34]), i.e,

$$f_{RF}(\mathbf{x}) = \frac{1}{B}\sum_{i=1}^{B} T_b(\mathbf{x}) \tag{4.4}$$

for the regression case, where B is the number of decision trees and $T_b(\mathbf{x})$ is the prediction for the b^{th} tree, and,

$$f_{RF}(x) = \arg\max_{c \in y} \ \frac{1}{B}\sum_{b=1}^{B} P_b(Y = c|X = \mathbf{x}) \tag{4.5}$$

for the classification case, where c is a category in y and $P_b(Y = c|X = \mathbf{x})$ is the probability of class c given the input $\mathbf{x}$ for tree b.

As mentioned, one of the key advantages of the random forest algorithm is its interpretability through variable importance. Variable importance measures the relative importance of each variable. This is done by measuring the prediction performance on so-called OOB (Out Of Bag) samples. When the b^{th} tree is being constructed, OOB samples, which are samples separate from the bootstrapped samples used to train the model, are used to measure its predictive performance at a particular split. Different permutations of the variable split are tried and the performance is measured on the OOB samples. The results are accumulated over all trees and the variables are ranked with respect to their importance in determining the overall ensemble prediction.

4.3.2 Time complexity analysis of Random Forests

The random forest can be summarized as follows [23],

1. Initialize the number of trees B in the forest.
2. for $b = 1 : B$
   (a) Draw a bootstrap sample Z of size N examples from the training set Ω

(b) Grow a random forest tree T_b to the bootstraped data by recursively repeating the following steps

   i. select m variables at random from D-dimensional input vector $\mathbf{x}$ (ideally $m = \sqrt{D}$)

   ii. pick the best variable split among the m variables based on the Genie criteria

   iii. split the node to two daughter nodes

(c) The classification is based on the majority votes from B trees.

To analyze the time complexity of the random forest algorithm as a whole, we find the worst case time complexity for training the random forest and then for predicting a data case. We reproduce the analysis from [34] which gives a rigorous derivation of the best, average and worst case time complexities of the random forest algorithm.

As the random forest algorithm consists of an ensemble of decision trees, we begin by analyzing the worst case time complexity of growing each tree. Given a data set Ω with N labeled samples of the form $(\mathbf{x}_i, y_i)$ where $|\mathbf{x}| = D$, the optimal split is found by recursively searching each variable and splitting on each value to find the impurity measure (Genie index). As a first step each variable D in $\mathbf{x}$ needs to be sorted from N different values. Using the fastest sorting algorithm, quick sort, this can be achieved with a worst case of $O(Nlog(N))$. With D different variables this becomes $O(DNlog(N))$ and for the randomized trees where a subset $m \leq D$ variables are searched, this becomes $O(mNlog(N))$. For growing M trees in the forest the worst case complexity becomes $O(MmN^2log(N))$.

For prediction, each data case must traverse through all M trees in the forest. The worst case time complexity for N examples is thus $O(MN)$.

4.4 Convolutional Neural Networks

The Deep Convolutional Neural Network(CNN) was first introduced in the 1990s [33] to classify hand written digits with near human level accuracy. Since then CNNs have been used for many different tasks such as image and speech recognition. For image recognition, CNNs have come to dominate the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).[2] Given its success in the field of computer vision, CNNs have very recently started to surface in the solutions to precipitation nowcasting problem. In [31] CNNs were used to predict the next reflectivity image from a set of previous images sampled at 10 minute intervals. To exploit the temporal correlations inherent in sequences of reflectivity images, [60] used a type of sequence-to-sequence learning deep neural network called the Long Short-Term Memory (LSTM) network for precipitation prediction. Our work in this thesis on one hand duplicates the above in that we try to predict the next reflectivity image in a sequence of reflectivity images. The difference lies in that we base our prediction on two measured atmospheric variables, one a sequence of reflectivity fields, the other a sequence of precipitable water vapor fields.

4.4.1 Multi-Layered Perceptron

The CNN has its roots in the multi-layered perceptron, the classical artificial neural network (ANN). An ANN generally has three layers – an input layer, a hidden layer and an output layer. The input layer consists of the D dimensional input vector $\mathbf{x} = (x_1, x_2, x_3, ..x_D)$. The hidden layer consists of some number of hidden units, each of which generates an output according to,

$$z_j = h(a_j) = h(\sum_{i=0}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}) \tag{4.6}$$

[2] http://www.image-net.org/challenges/LSVRC/

where w_{ji} is the "weight" between input i and hidden node j and h is the so-called "activation function". In practice, h is usually one of three non-linear functions, the sigmoid,

$$h(a) = \frac{1}{1 - e^{-a}} \tag{4.7}$$

the hyperbolic tangent,

$$h(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-1}} \tag{4.8}$$

or the rectifier function,

$$h(a) = max(0, a) \tag{4.9}$$

Similar to the hidden layer, the output layer is either the weighted sum of the outputs of the hidden layer nodes or some function (sigmoid, hyperbolic tangent, or rectifier) of the weighted sum of the outputs of the hidden layer nodes.

An ANN is trained by adjusting the hidden and output-layer weights to minimize the mean-squared error (MSE) in case of regression problem,

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{4.10}$$

or the cross-entropy in the case of classification problem,

$$L(w) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=c}^{C} y_{ik} \; log \; f_c(\mathbf{x_i}) \tag{4.11}$$

A revolution in the use of ANNs, including rainfall prediction[22], occurred in the 1990's following the discovery of the back-propagation algorithm, an iterative gradient descent algorithm based on the chain rule of differentiation, for adjusting the ANN weights to minimize the objective function over a training set [44].

4.4.2 Convolutional Neural Networks

Since it was proven in the 1980's that an ANN with a single hidden layer can, in theory, represent any arbitrary function to an arbitrary degree of accuracy [39],

research on neural networks with more than one hidden layer seemed unnecessary. This was until the introduction of the CNN. Rather than viewing machine learning as a function approximation problem, the CNN sought to mimic in some sense the functioning of the human visual cortex. Here we discuss the properties of CNNs, – local receptive fields, shared weights, and pooling – that differentiate CNNs from ANNs and give them their power for problems in image recognition and prediction.

4.4.3 Convolution Layer

Unlike traditional ANNs where every input is connected to every hidden unit in a layer, CNNs have "sparse connectivity" in that only a local set of inputs termed as "local receptive fields" is connected to each hidden unit. Figure 4.2 illustrates this where (top) each unit in the hidden layer of the CNN is connected to three variables in the input or its local receptive field, and (bottom) shows a fully connected ANN, where each input unit is connected to all hidden units performing matrix multiplication. The sparse (local) connectivity of the CNN reduces the the number of weights the network has to learn, which in turn reduces the memory and computational burden, thereby allowing for the construction of deeper networks by stacking multiple convolution operations.

To better understand the operation of convolution over images consider in Figure 4.3 an $N \times N$ input image labeled the input neurons to a CNN. The output of the convolution layer is labeled the first hidden layer in the figure. A neuron in the hidden layer of a CNN is connected to a local region (5×5 neurons) in the input called the receptive field. This filter of size 5×5 is slid across the image with a stride S ($S = 1$ in the figure). The same weights of the filters are used as the filter is slid across the image with stride 1 connecting the various 5×5 regions in the inputs to the different neurons in the hidden layer, thus giving us the property of "shared weights". Shared weights is derived from the property of "translation invariance" of an image where

a feature detected in one location of the image is considered as the same in another location of the image. In Figure 4.3 the CNN layer has $K = 1$ filter. The number of filters can be varied which would then increase the depth of the first hidden layer where each neuron in each depth is connected to the same same receptive field through a different set of filter weights there by learning to extract different features from the input image.

The sliding can be thought of as a convolution operation with a 3D input image tensor $V_{i,j,k}$– where i is the axis representing the different channels (red, green and blue for images) or depth, and j, k represents the spatial location of the pixel intensity in the image – is convolved with a 4D kernel tensor $K_{i,j,k,l}$ which represents the weight connecting an input unit at depth j to an output unit at depth i with an offset of k, l. The convolution operation performed by a convolution layer can be summarized by the following equation (c.f [8]),

$$Z_{i,j,k} = c(\mathbf{K}, \mathbf{V}, s) = \sum_{l,m,n} V_{l,(j-1)\times s+m,(k-1)\times s+n} K_{i,l,m,n} \tag{4.12}$$

where s is the so-called stride and $Z_{i,j,k}$ is the output neurons of the convolution layer. This can be seen as a 2D convolution with the tensor $\mathbf{K}$ representing the filters, and learning a CNN can be thought of as constructing different filters, each responding to the presence of a different feature in the input image[3]. To deal with boundary cases in convolution we can pad the inputs. Three types of padding strategies are employed in practice. These are 'valid' padding where no padding is added; 'same' padding where padding is added to give the output the same shape as as the input; and 'full' padding such that the convolution operation visits every input unit once (i.e. filter size - 1 zeros are added to each side of the input).

[3]Noting that the filters are not flipped when convolving across the image, the operation performed by a CNN is more technically a cross-correlation than a convolution.

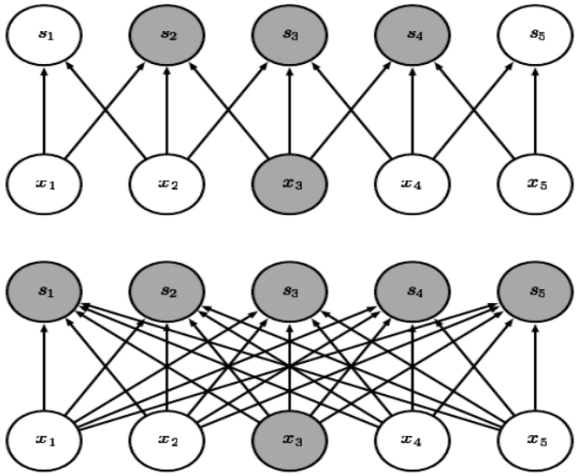

Figure 4.2: Connections for an ANN (bottom) and a CNN (top). Whereas an ANN is fully-connected, a CNN is sparsely(locally), in this case with a kernel width of 3. Figure from [8]

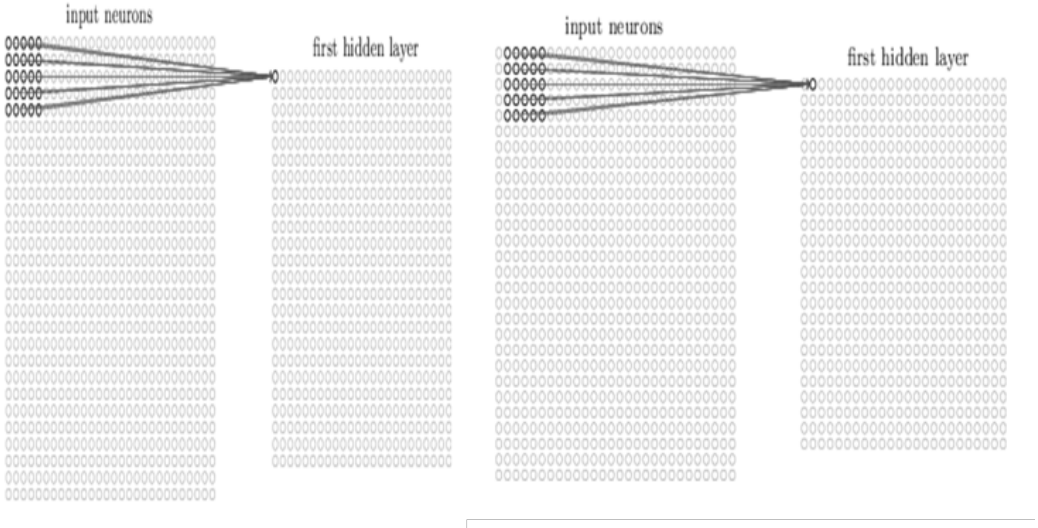

Figure 4.3: Input and output of a convolution layer with 1 filter, a 5×5 receptive field and a stride of 1. Image from http://neuralnetworksanddeeplearning.com/chap6.html

The number of output units or neurons $\mathbf{Z}$ are given by the following equations[4],

[4]http://cs231n.github.io/convolutional-networks/

$$W_2 = (W_1 - F + 2P)/S + 1$$

$$H_2 = (H_1 - F + 2P)/S + 1$$

$$D_2 = K$$

where W_2, H_2, D_2 is the volume (width, height and depth) of the output/hidden layer and W_1, H_1, D_1 is the volume (width, height and depth) of the input to a CNN layer with K filters, a receptive field of F, stride by S and padded with P zeros on each side of the image.

Multiple convolution layers can be stacked together where the output of one convolution layer is the input to an another convolution layer creating a deep network. The output of a convolution layer can also be the input to a pooling layer as explained in the next section.

4.4.4 Pooling Layer

It is common practice to add a pooling layer after a convolution layer. A pooling layer is defined by an $M \times M$ receptive field and stride s. The pooling layer differs from a convolution layer in that the receptive field has no weights but rather performs an operation such as average, max, or square root of the sum of squares over the receptive field.

Figure 4.4 below puts it all together. Adapted from a hand writing recognition problem where the input consists of hand written digits digitized to a 28×28 pixel resolution. A convolutional layer obtained by sliding three different 5×5 receptive field windows over the input image – each with different weights and hence each looking for different features. A pooling layer consisting of three pooling 12×12 maps is obtained by sliding 2×2 windows over the different feature maps.

The pooling layer is then connected to a fully connected layer (not pictured in the figure) with ten units and a "softmax" activation, giving a probability score for each one of the 10 digits. Like ANNs, CNNs are trained using back-propagation, stochastic gradient descent.

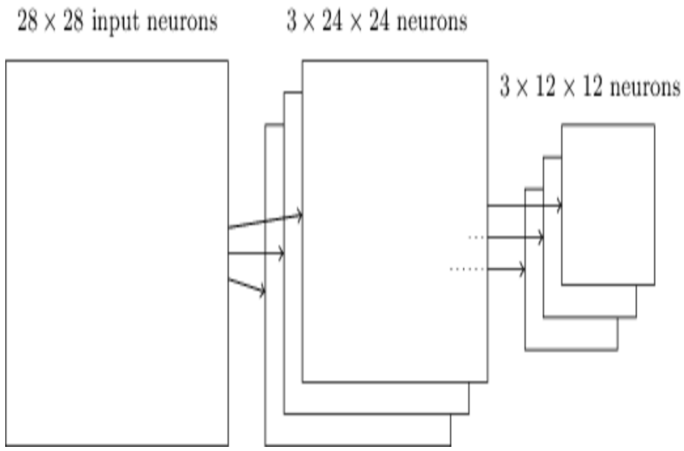

Figure 4.4: 1 convolution and 1 pooling layer. Image from http://neuralnetworksanddeeplearning.com/chap6.html

4.4.5 Regularization and Dropout

A problem facing deep neural networks is overfitting. Two ways to overcome this problem are – regularization and dropout. Regularization is a family of techniques that includes early stopping to stop the training before overfitting occurs, or constraining the parameters of the network in various ways, e.g., by limiting the number of hidden units or penalizing the weight vectors, to impair the network's performance. Dropout is a very recently developed technique [53] that is becoming the standard for CNN's because it greatly speeds the training process. At each stage of training, dropout involves removing a given network node with probability p and training only those nodes that remain. The "dropped out" nodes are then reinserted with their

original weights for the next stage of training and the process repeats. By avoiding training all nodes on all training data, dropout decreases overfitting and speeds training.

4.4.6 Analysis of number of parameters, operations and memory

Let us consider our input tensor of shape $(8 \times 33 \times 33)$[5]. We pass this through a convolution layer with $K = 16$ filters, a receptive field of $F = 5$ and a stride of $S = 2$ with 'full' padding or $P = 4$ on each side. Using the equation from the Section 4.4.3 this produces an output tensor (hidden neurons) of shape $16 \times 19 \times 19$ or 5776 hidden units/neurons. A receptive field of 5 means that each unit in the hidden layer is connected with $5 \times 5 \times 8 = 200$ weights and 1 bias term to the input layer for which 201 parameters per hidden unit. The total number of parameters the convolution layer needs to learn would thus appear to be $5776 \times 201 = 1,160,976$. Recalling the notion of parameter sharing, separate parameters are not used at each location. Replication the $K = 16$ filters across the layer we thus obtain a number of parameters equal to $16 \times 5 \times 5 \times 8 = 3200$ plus 16 bias terms for a total of 3,216. Contrast this to a fully connected layer where each input is connected to each hidden unit, in which case we would obtain $5776 \times 8712 = 50,320,512$ parameters, which may not even fit in memory, making implementing such a network infeasible if not impossible.

[5]As will be discussed, this will be the size of our "tensors" we use for the precipitation nowcasting problem

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter we apply the machine learning algorithms – random forest (RF) ensemble algorithm and convolutional neural network (CNN) algorithm – discussed in the previous chapter to the precipitation nowcasting problem. By comparing the nowcasting performance of the two machine learning algorithms when trained using only radar reflectivity fields and when trained using both radar reflectivity and NIPW fields, we draw conclusions relative to our thesis that water vapor information in the form of NIPW might be expected to improve the performance of a reflectivity-based precipitation nowcaster. By comparing the two machine learning algorithms in terms of problem representation, implementation, and performance, we draw conclusions related to the suitability of each algorithm to the nowcasting problem. And by comparing the nowcasting performance of our algorithms against published results for current state-of-the-art nowcast algorithms, we are able to draw conclusions regarding potential for improvement.

5.1 Problem Representation

The success of machine learning algorithms generally depends on how the problem is "represented" to the algorithm. Representations are generally done manually based on prior knowledge about the problem.

As described in previous chapters, our broad goal is to take a sequence reflectivity images and a sequence of water vapor images (in the form of normalized integrated precipitable water vapor, NIPW) recorded at spatial resolution of 3 km over a 300

sq-km domain and a temporal resolution of 30 minutes and predict from them the reflectivity image 1-hour in the future.

As a first simplification we will formulate the above problem as a classification problem – raining or not raining – where we define rain at a pixel reflectivity value $\geq$ 24 dBZ (corresponding to approximately 0.9 mm/hr of rainfall). Given the sequence of reflectivity and water vapor images, $X_{t_i}^{refl}$, $X_{t_i}^{ipw}$, $t_i = t - T, \ldots, t$ our goal is thus to predict the binary image Y_{t+2} where element i, j is a 1 if the reflectivity at i, j is predicted to exceed 24 dBZ and 0 otherwise. Here T is the amount of past history of reflectivity and NIPW images that we use for the prediction. In what follows we choose $T = 4$, or 2 hours of past images. This should be adequate to infer reflectivity (storm) advection rates and rates of water vapor convergence.

As a second simplification we will not try to predict the entire 100×100 binary output field Y_{t+2} at once, but rather we will only try to predict the value at a single pixel. Then to obtain the 100×100 output field, we separately predict the values for each pixel. The idea is that where we would likely never have sufficient training data to predict entire 100×100 output fields (there are simply too many permutations of input patterns of reflectivity and NIPW to output patterns of binary rainfall), we should have sufficient training data to learn the input patterns that lead to rain at a single pixel.

Given that weather has limited spatial influence, i.e., the weather at a given pixel location 1-hour from now is most strongly influenced by the weather that can reach that pixel in 1 hour, we will only look at the evolving reflectivity and NIPW within a 50 km radius of the pixel we are trying to predict. Since 50 km/hr far exceeds the typical advection rate of storms in the DFW region, the weather in the 100 sq-km region surrounding the pixel should be ample to see storms moving towards the pixel from any direction. This reduces the size of the input from the full 100×100 pixels

that comprise the DFW domain to the 33×33 pixels surrounding the pixel under consideration.

Finally, under the assumption that the relative flatness of the DFW region makes it homogeneous in terms of how weather forms and evolves, we will only seek to learn a single model mapping reflectivity and NIPW to binary rainfall. We will train and validate the model on 500 points randomly selected from an 66×66 pixel region inscribed within the 100×100 domain (the smaller 66×66 region ensures a full 33×33 region around each pixel). We will test the resulting nowcast algorithm by applying the same model to every pixel in the domain.

5.2 Feature Extraction

Classification is generally based on the identification of certain key features in the input. For machine learning classifiers, these features can be manually or automatically extracted/learned.

Four 33×33 fields of NIPW and four 33×33 fields of reflectivity from the last two hours are the raw inputs to the machine learning algorithms. For the random forest classifier we manually extract the following features from these raw inputs:

1. Average NIPW value in each NIPW frame (4 features): This gives a measure for the amount of water vapor surrounding the pixel to be predicted.

2. Average reflectivity value in each reflectivity frame (4 features): This gives a measure for rainfall rate (using reflectivity as a surrogate) surrounding the pixel to be predicted.

3. Standard deviation of the NIPW values in each NIPW frame (4 features): This gives a measure of the spread of water vapor surrounding the pixel to be predicted.

4. Standard deviation of reflectivity values in each reflectivity frame (4 features): This gives a measure of the spread of values about the mean – uniformly or not uniformly distributed – of rainfall intensity surrounding the pixel to be predicted.

5. Difference between successive NIPW averages (3 features): Gives information about the rate of moisture convergence, which has been shown to play a role in storm growth and decay [16].

6. Difference between successive reflectivity averages (3 features): Gives information about the rate at which a storm is advecting into or out of the region surrounding the pixel to be predicted.

7. Difference between alternating NIPW features (2 features): Since we predict 1 hour in advance leaving out the intermediate 30 minute prediction, these 1-hour differences may be important.

8. Difference between alternating reflectivity features (2 features): Same as above.

The input to the random forest classifier is thus a vector of real numbers containing the 26 features described above. Through these features, we have manually identified moisture concentration and convergence as well as rainfall intensity, advection, growth and decay. All of these are features that the physics of the problem (recall Chapter 2) have suggested are important to the onset and cessation of rainfall.

Regarding the CNN, deep learning architectures have been shown able to automatically extract multiple layers of representation from raw inputs such as images or text. For instance in image classification the first few layers of deep neural networks extract features such as edges while deeper layers extracts more complex features such as specific shapes. In the same way we would like to see if a CNN is able to extract similar features in the context of rainfall prediction such as storm advection

rate and growth and decay. To test this we provide the CNN an input tensor with shape $8 \times 33 \times 33$ consisting of the 4 NIPW frames and 4 reflectivity frames from the last two hours.

Note that not only do the features provided to the random forest explicitly capture the temporal aspects related to storm advection, growth, and decay, they do so in a way that does not depend on storm direction. This is not the case for the CNN. By making nothing explicit, the CNN will have to learn not only the spatial features in the input frames that predict rain, they will also have to learn, if possible, the predictive value of changes in features over time. Regarding storm direction, we believe that the translational invariance property of the CNN would allow it to recognize patterns leading to rain regardless of the direction they are coming from.

5.3 Training and Validation

The generalization performance of a prediction system is important because it must be able to make a good prediction given any kind of event and not only perform well on specific types of events. This is particularly important in the meteorological domain where storms can be convective or stratiform, can come from any direction, and can travel at a wide range of speeds. To evaluate our prediction systems, we use the K-fold cross validation technique. This technique splits the dataset to K folds or sub-datasets. A model is trained on K - 1 folds and validated on the K^{th} fold. The process is repeated until all K folds have been validated. The prediction score for the system is then taken as the average of the performance over the K folds.

In our problem we divide our dataset into 9 folds, where each fold is data from a different month. The 9 months of data were obtained as follows: 4 months from 2014 (May - August), 2 months from 2015 (May, June) and 3 months from 2016 (May - July). For each month we use the reflectivity and NIPW data from the weather anomaly days listed earlier in Table 3.1. In this thesis however we only use

the 4 months from 2014 as our validation set to evaluate the random forest and the convolutional neural network algorithms as each training and validation block takes a long time to train explained as follows.

Our data set consists of 49 weather anomaly days selectively chosen from Table 3.1[1]. NIPW and reflectivity fields are sampled at 30 minute intervals each day thus obtaining 48 (some time steps are dropped because we don't have 2 hours worth of prior data at the beginning of the day) examples per day. We thus obtain a total of 2361 time steps to train on for each pixel point. Randomly sampling 500 points from our domain gives a total of 1,180,500 training examples.

For the random forest experiments where we extract the 26 features as our training data obtaining a input matrix size of $1,180,500 \times 26$ and an output column vector of size $1,180,500 \times 1$. If we use double precision floating point to represent the two matrices, this occupies a total of $1180500 \times 27 \times 64 = 2,039,904,000$ bits or 250MB.

On the other hand for the CNN experiments where the entire frames are inputs to the algorithm we have $1180500 \times 8 \times 33 \times 33 \times 64 = 658,209,024,000$ bits or 82 GB[2]. The data is read from disk in batches and processed by a Tesla K-80 GPU. The workhorse of deep learning techniques, the batch stochastic gradient descent method allows us to evaluate the gradient in batches and sum them together in the end [39].

5.4 Measuring Nowcaster Performance

The output of our prediction systems are real numbers between 0 and 1. These can be interpreted as probability scores, or beliefs, that it is raining or not raining at a selected pixel location. The actual rain, no-rain decision is determined by a decision

[1]Some weather anomaly days were dropped due to bad NIPW fields and some were relatively small storms

[2]Theano uses single precision floating point thus the tensors are converter to 32 bit floating point when training

Table 5.1: Contingency Matrix

	Predicted 1	Predicted 0
Actual 1	T_p	F_n
Actual 0	F_p	T_n

threshold - rain if the output exceeds the decision threshold, no-rain otherwise (the probability decision threshold is set at 0.5 but can be varied to determine rain or no rain).

The contingency matrix shown in Table 5.1 defines the quantities, T_p or the number of cases predicted correctly(hits), F_n the number of false negatives (misses), F_p the number of false alarms and T_n the number of negative cases predicted correctly. The metric most commonly used to evaluate nowcasting algorithms is the critical success index ($CSI = \frac{T_p}{T_p+F_n+F_p}$) otherwise known as the threat score [43]. CSI ranges from 0 to 1, with a value of 1 indicating a perfect nowcast. Compared to measures like Probability of Detection ($POD = \frac{T_p}{T_p+F_n}$) and False Alarm Rate ($FAR = \frac{F_P}{F_P+T_p}$), CSI takes into account both false alarms and missed events, making it a more balanced score.

For our machine learning nowcaster we calculate the CSI on the predictions made on 500 randomly chosen points for the storm dates from validation set.

5.5 Random Forest Experiments

For our random forest experiments we used the random forest implementation in the Scikit-learn Python API[3] trained on a server with 32 cores.

[3]http://scikit-learn.org/stable/

The first step in learning a random forest nowcast system is to determine the hyper-parameters of the random forest. These are the maximum depth that each tree can be grown, the total number of trees in the forest and, the maximum number of variables to sample at each split. To choose the hyper-parameters, we use a grid search approach involving arbitrarily selecting set of values for each hyper-parameter and choosing the values hat obtain highest average CSI score on the 4 validation sets. For the total number of trees we choose from 400, 500, and 600, for maximum tree depth we choose 6, 9 and 12. For the maximum number of variables we sample randomly from $log(D)$ variables or D variables.

Figure 5.1 shows the CSI scores averaged over the four training and validation blocks evaluated for both the NIPW + reflectivity (I + R) and reflectivity(R) only experiments using the training samples and validation samples. The bottom row of the figure shows the training and validation CSI scores for the random forest which samples all D variables and the top row for $log(D)$ variables. From both plots on the left representing the training CSI scores as a function of max tree depth, we see that CSI scores increase with increasing tree depth. This is a case of overfitting and is expected as deeper trees give better fit. From evaluating the training CSI scores the I + R model obtains a better CSI than the R model for almost all max depths. We thus choose the hyper-parameters which yield the best CSI score on the validation set. This is observed for the random forest which samples all D variables at each split, uses a max tree depth of 6 and 600 trees in the forest for both the I + R and the R models.

5.6 Convolutional Neural Network Experiments

Unlike the random forests where features are manually extracted, the inputs to the CNN are the raw NIPW and reflectivity images represented as $8 \times 33 \times 33$ tensors. Using the various hyper-parameters of the CNN described in the Chapter 4.4, we

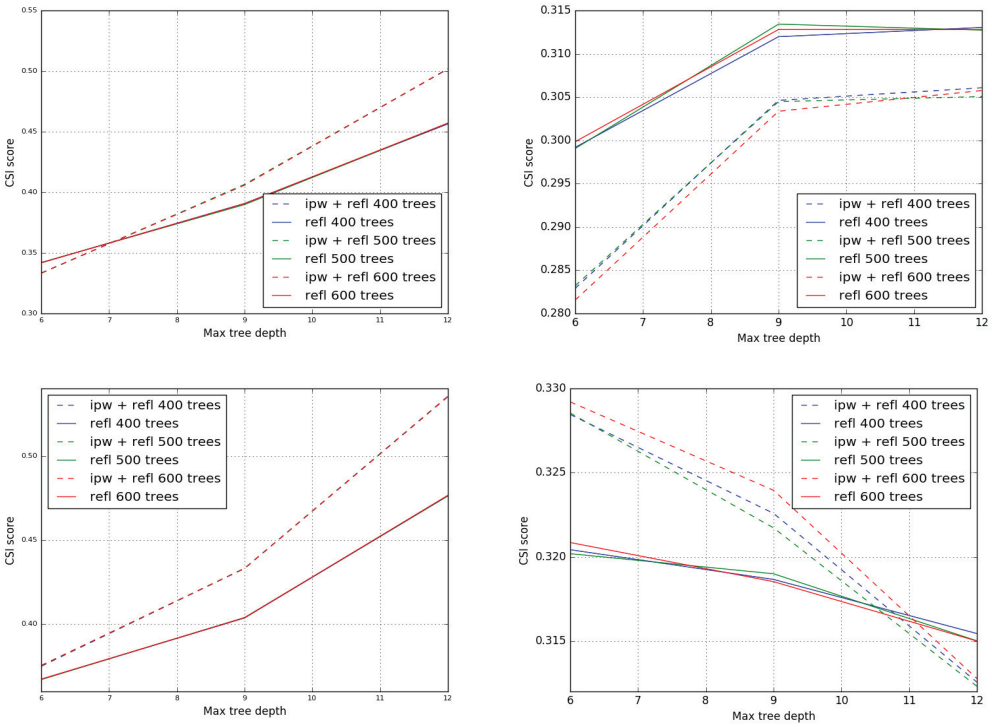

Figure 5.1: CSI scores for RF1 and RF2 evaluated on the training set and the validation set. Top left: average CSI scores evaluated on training set using RF2. Top right: average CSI scores evaluated on validation set using RF2. Bottom left: average CSI scores evaluated on training set using RF1. Bottom right: average CSI scores evaluated on validation set using RF1

explore 2 CNN architectures each for the I + R model and the R model. Figure 5.2 shows the schematic of the single layer convolutional neural network CNN1 and a two layer convolutional neural network CNN2 along with the hyper-parameters for each layer. From the figure we see that the connections from the reflectivity fields and the NIPW fields to the convolution layer are separate and features learnt from the individual convolution layer(s) are later concatenated before passed through a fully connected (FC) layer with a dropout rate of 20%. This is done in the concatenation layer where the neurons of each of the incoming layers are stretched out as vectors from tensors and joined together to form a large vector. The output of the FC layer

is passed through a softmax layer with two units which outputs a probability score for rain or no-rain at the center of the 33×33 one hour into the future.

The motivation for connecting the two types of inputs (i.e. reflectivity and NIPW images) separately this way is we can use different convolution layer parameters (i.e. receptive field, stride and number of filters) for each input type, based on the spatial sampling and the way each input evolves over time. However in this thesis we use the same convolution parameters for both types of inputs shown in Figure 5.2. This type of connections can be generalized for any number of weather variables representing the weather at a particular location. The variables can also be of different spatial resolution making a single connection to the convolution layer infeasible.

Both CNN1 and CNN2 are trained using the batch stochastic gradient descent technique with a batch size of 250. Since all the training data does not fit in memory each point is read from disk separately. Before training the averages of each feature (i.e. each of the $8 \times 33 \times 33$) is computed over all storm dates for that point and is subtracted from the respective feature (i.e. we center each feature around 0 which should help the optimizer to find the global minima faster). The networks are trained using the Nestrov momentum [55] with a training rate of 0.0001 and a momentum of 0.9 for 200 epochs.

Our implementation of the CNN is using the Python Theano library[4]. The CNN is trained using a Tesla K-80 GPU instance available to us from Massachusetts Green High Performance Computing Cluster (MGHPCC)[5].

---

[4]http://deeplearning.net/software/theano/

[5]http://www.mghpcc.org/

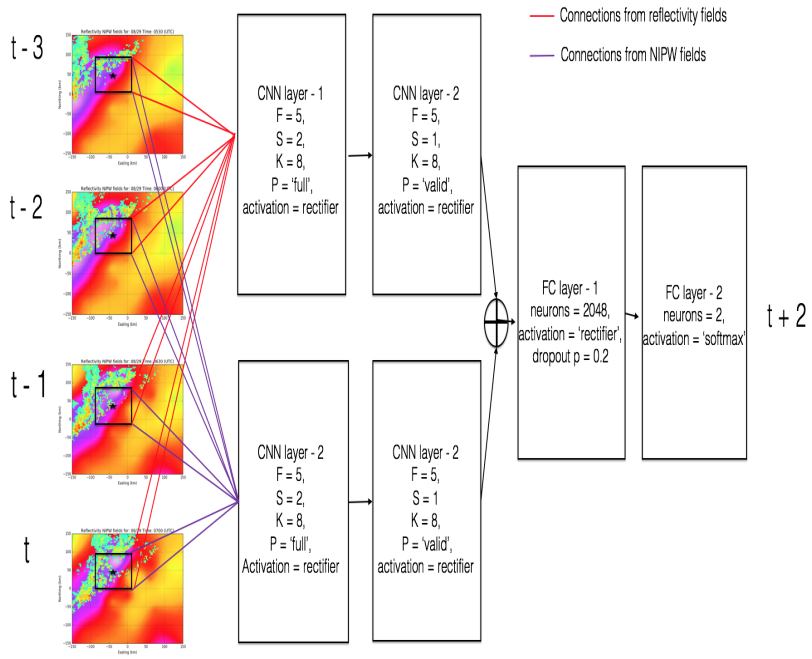

Figure 5.2: CNN 1 and 2 architectures, where F is the receptive field size, S is the stride, K is the number of filters and P is padding strategy. CNN1 uses only the first convolution layer followed by the concatenation layer. CNN2 uses both convolution layers followed by the concatenation layer

5.7 Results

Both RF and CNN are trained and evaluated when IPW and reflectivity (I + R) are inputs and when reflectivity only (R) are inputs to the algorithm.

5.7.1 Random Forest Results

Tables 5.2 - 5.5 shows the POD, FAR and CSI for the 4 validation months and Table 5.6 summarizes the averages of these metrics taken over the 4 validation months evaluated over the randomly chosen 500 points. The RF using the R model seems to perform as well or better for the months of May, June and July but in the month of August the I + R model seems to perform better based on the CSI scores. The RF I + R obtains an average CSI of 0.33 compared with an average CSI of 0.32 for the

R model. This small difference in the CSI scores is not enough to conclude that the RF has learnt the growth and decay information in the I + R model which allows the model to make a better prediction.

5.7.2 Random Forest Feature Importance

From Section 1.1 and chapter 2 we have seen that water vapor should predict the onset of rainfall. We thus based our conjecture that water vapor information must help with nowcasting precipitation. To see how much water vapor helped the I + R RF model we evaluate the feature importance of the I + R and R models of the RF. Figure 5.3 shows the variable importance of the I + R model. From this we see that the average and standard deviation of reflectivity and the rate of average reflectivity change are dominant variables in making a prediction. It is also interesting to note that there is a feeble contributions from the NIPW averages which could have contribute to the slight performance gain of the I + R model. However as a whole the top three variables are ones derived from the reflectivity fields and the RF seems to have ignored the NIPW data. From Figure 5.4 we again see the same top three variables as the I + R model for the R model.

5.7.3 Convolutional Neural Network Results

In the CNN family of models the inputs to the algorithm are the raw tensors of NIPW and reflectivity fields and no information about advection rate and water vapor convergence are explicitly given to the model. Our hope is that the convolution layer will learn the feature representation to make a good prediction. We use the same training and validation set strategy as the random forest experiments.

Figure 5.5 shows the training and validation loss (cross entropy objective) for CNN1 measured at each epoch for 200 epochs for both I + R and R models. From evaluating the training and validation loss on the 4 folds we see that the training loss converges converges to the same value for both I + R and R models. Although the

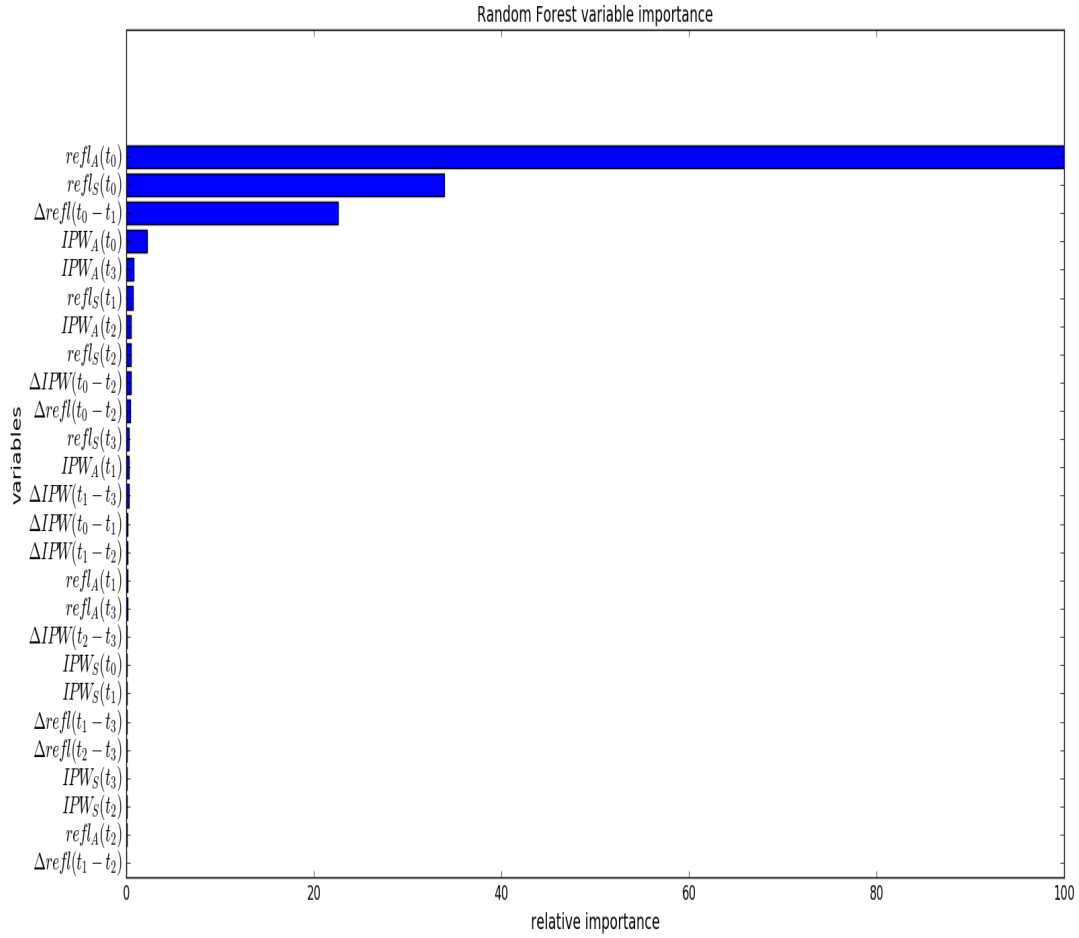

Figure 5.3: Relative variable importance for trees grown by sampling all features at each split for the I + R model

training loss seems to have converged quite well, we see that the validation loss has not converged quite as well for both models (I + R, R). This is likely because our training set is too small to expose the CNN to the huge number of different input pattens that the NIPW and reflectivity fields can take and there by attaining a good generalization performance on the validation set is difficult. This is especially true for the month of June where we see a general trend that the validation loss is decreasing but there are still large fluctuations in the validation loss. However we can also see that the R

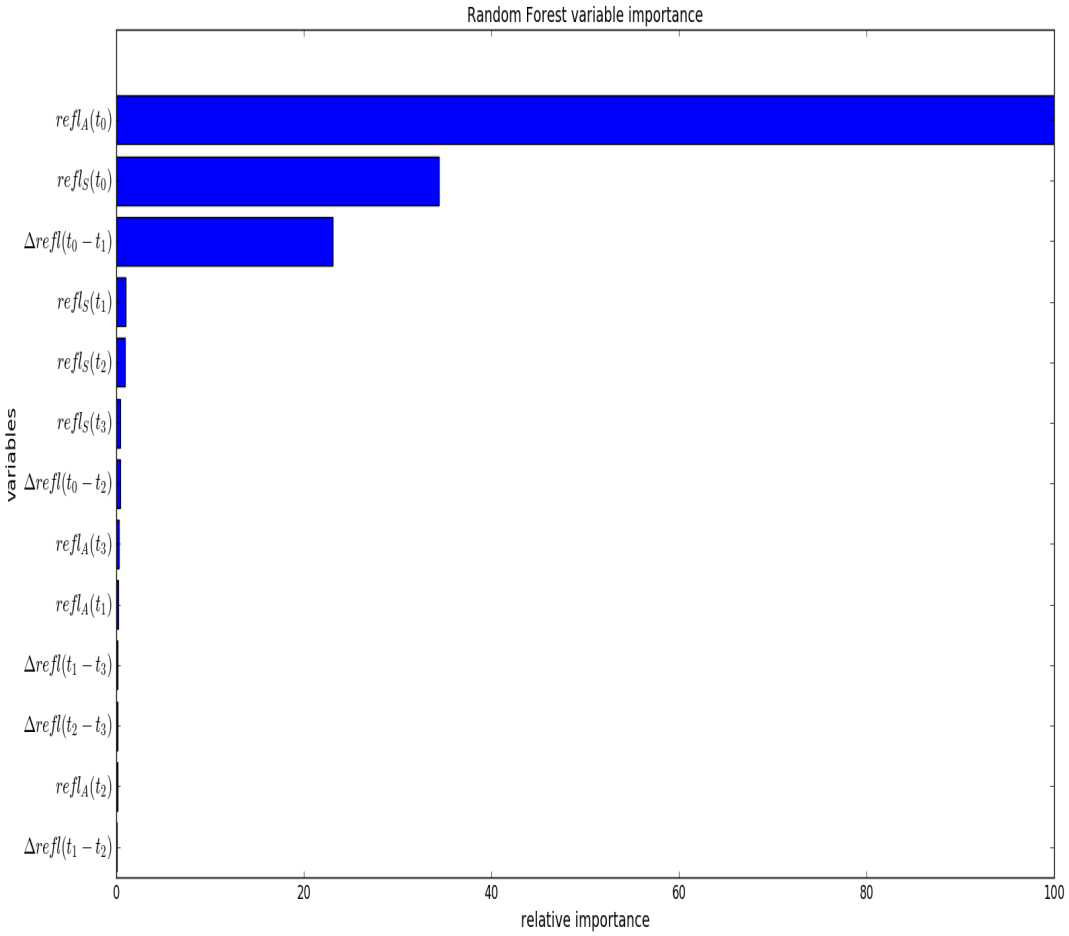

Figure 5.4: Relative variable importance for trees grown by sampling all features at each split for the R model

model for June also finds it difficult to obtain a good generalization performance on the validation set. Clear from the May, June and August validation sets, the I + R model performs better than the R model. However from the validation loss in July 2014 we see that R model converges to a lower value than the I + R model. This may be due to the stratiform storm event encountered during the month of July.

We determine the epoch at which the model attains the best CSI score on each of the validation sets and use the network parameters at this epoch to make predictions

on storms from the validation set. The results for the 4 validation months are shown in Tables 5.2 - 5.5 along with the averages in Table 5.6. Comparing the performance metrics of the I + R and R models of the CNN1, we see that the CSI is higher for I + R model for the months May and June and the inverse is true for the months of July and August. On average however the CSI of the I + R and R models are the same. From Table 5.6 it is also clear that the average CSI of CNN1 with a score of 0.38 is much higher than that of the RF1 model.

Figure 5.6 shows the training and validation loss of the CNN2 model evaluated on the 4 cross validation sets. We see that by increasing the capacity of the model i.e. by allowing more layers in the model to capture higher-order features from the data we are able to obtain a lower training loss than the CNN1 model and the validation loss seems to have stabilized better compared to the CNN1 model. This suggests that CNN2 has learnt a better feature representation of the storm advection rate and water vapor convergence that it can now generalize better. The validation losses for the month of May and August are marginally lower for the I + R model compared to the R model and the inverse is true for the months of June and July.

We determine the epoch where we obtain the best CSI score on the validation set for each model and measure the nowcasting performance made by the network using parameters at this epoch. The nowcasting metrics evaluated on the 4 validation sets for CNN2 are summarized in Tables 5.2 - 5.5 and the summaries in Table 5.6. From Table 5.5 we see the largest difference in CSI scores between the I + R and R model using the CNN2 architecture with CSI score of 0.38 and 0.35 respectively. From Table 5.6 we see that CNN2 achieves the best CSI overall from the I + R model with a value of 0.39.

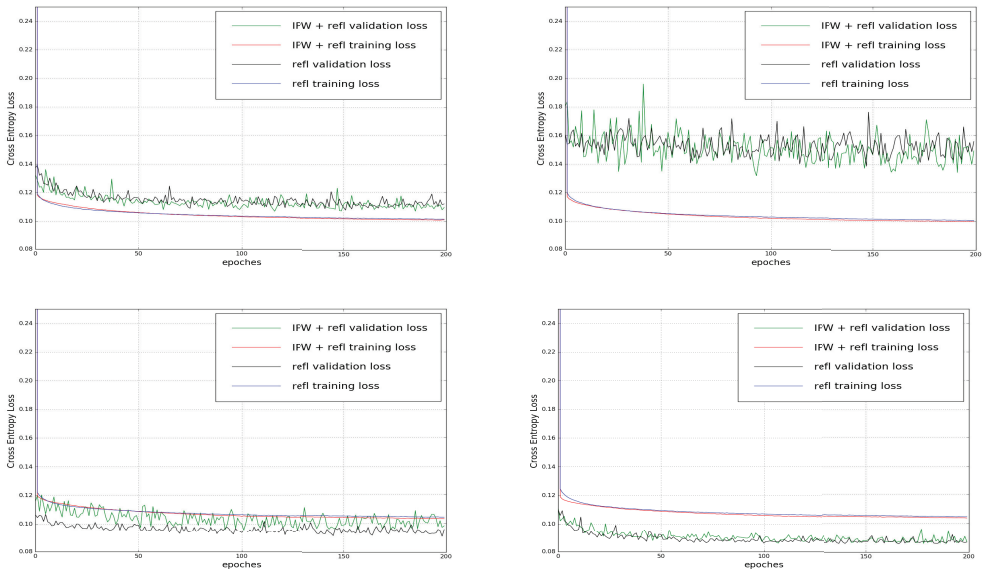

Figure 5.5: Training and validation loss for the 4 cross validation blocks using I + R and R models. Top-left: May 2014 Top-right: June 2014 Bottom-left: July 2014 Bottom-right: August 2014

5.7.4 Evaluating Nowcasting Performance on Individual Storm Cases

Tables 5.7 and 5.8 show the performance metrics of the CNN1 and CNN2 evaluated on individual storms for all points in our prediction domain. The storm dates column represents the UTC day in which the nowcasting algorithm started prediction and was run through the number of days represented in the table. All storms in May 2014 are predicted better by the I + R model than the R model. Figure 5.8 shows the inputs and predictions made by CNN2 I + R and R models from 17:30 - 20:30 UTC on May 12th. From Table 5.8 we see that the I + R scores a higher CSI of 0.54 compared to the R model with a CSI of 0.50. It is also clear from the Figure where we see the water vapor convergence in the North-West of the prediction domain from 17:30 - 19:00 in the inputs results is a better prediction at by the I + R model at 20:00. From this figure we can see that the NIPW fields are smooth and clearly show

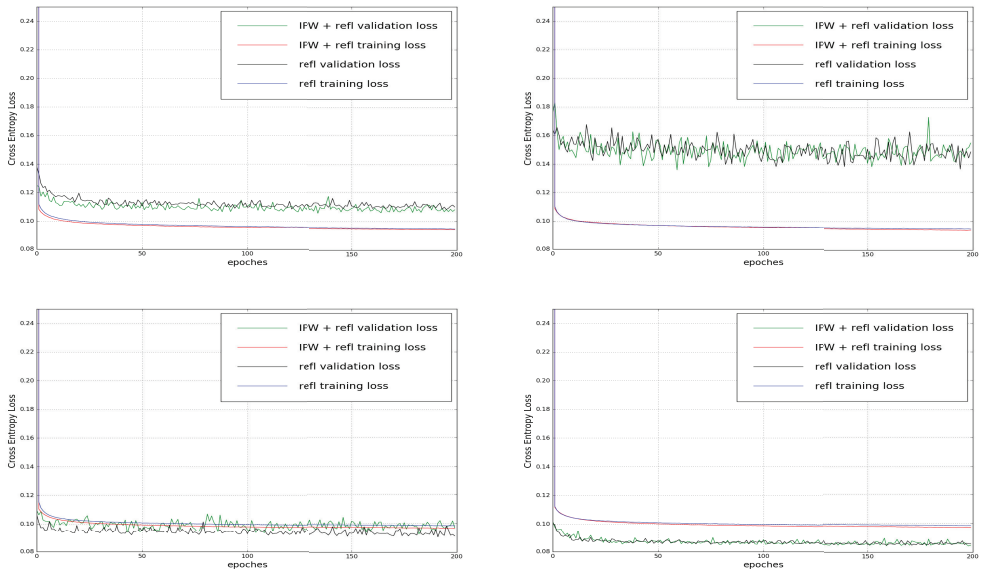

Figure 5.6: Training and validation loss for the 4 cross validation blocks using IPW + reflectivity variables and reflectivity variables only trained on the 2CNN model (top-left)May -2014 (top-right) June 2014 (bottom-left) July 2014 (bottom-right) August 2014

the water vapor convergence that allow the I + R model to perform better. However this is not always the case.

Figure 5.9 shows the inputs and predictions made by the CNN2 I + R and R models for a storm on June 9th 2014 from 12:30 - 15:00 UTC. The prediction made at 14:30 and 15:00 seem to be discontinuous in that there are holes in the predictions themselves. This is due to the fact that the NIPW fields are noisy where there are large fluctuations in IPW seen from individual stations compared to the previous case and a strong water vapor convergence is not present like the one seen earlier. From Table 5.8 the R model performs better than the I + R model with CSI scores of 0.46 and 0.38 respectively. From this figure we can also generally see that the pattern of water vapor convergence followed by rainfall is not followed like the May 12th case.

We thus note that for days where the 44 stations measure accurate IPW values, the I + R model tends to perform better than the R model. One of the reasons the

R model performs better than the I + R model contrary to our conjecture is that the IPW values measured by the stations add more noise than signal and degrade system performance.

Table 5.2: Performance metrics of the best models validated for storms during May 2014

	CNN1		CNN2		RF	
	I + R	R	I + R	R	I + R	R
POD	0.57	0.55	0.53	0.54	0.44	0.45
FAR	0.35	0.36	0.33	0.36	0.32	0.33
CSI	0.43	0.42	0.42	0.41	0.37	0.37

Table 5.3: Performance metrics of the best models validated for storms during June 2014

	CNN1		CNN2		RF	
	I + R	R	I + R	R	I + R	R
POD	0.58	0.55	0.53	0.53	0.53	0.56
FAR	0.40	0.41	0.41	0.41	0.41	0.42
CSI	0.42	0.40	0.39	0.39	0.38	0.40

Table 5.4: Performance metrics of the best models validated for storms during July 2014

	CNN1		CNN2		RF	
	I + R	R	I + R	R	I + R	R
POD	0.43	0.44	0.46	0.48	0.34	0.32
FAR	0.42	0.36	0.38	0.40	0.31	0.27
CSI	0.33	0.35	0.36	0.36	0.29	0.29

5.7.5 Comparison With State-Of-The-Art Nowcasting Algorithms

Current state-of-the are nowcasting systems such as Conv-LSTM [60] and ROVER [14] predict a sequence of reflectivity frames for the next two hours at a 10 minute resolution. The reflectivity fields predicted are also the intensity fields and not the

Table 5.5: Performance metrics of the best models validated for storms during August 2014

	CNN1		CNN2		RF	
	I + R	R	I + R	R	I + R	R
POD	0.46	0.47	0.54	0.46	0.30	0.25
FAR	0.42	0.39	0.43	0.41	0.30	0.28
CSI	0.34	0.35	0.38	0.35	0.26	0.23

Table 5.6: Average performance metrics

	CNN1		CNN2		RF	
	I + R	R	I + R	R	I + R	R
POD	0.51	0.50	0.51	0.50	0.40	0.40
FAR	0.40	0.38	0.39	0.39	0.34	0.33
CSI	0.38	0.38	0.39	0.38	0.33	0.32

binary fields like the predictions made in this thesis. CSI values for these two algorithms were determined by evaluating their ability to predict that rainfall intensity would be above or below 0.5 mm/hr (contrast this to our algorithm, which predicts if rainfall intensities will be above or below 0.9 mm/hr). The CSI scores are provided for Conv-LSTM and ROVER in [60].

Figure 5.7 shows the nowcasting performance curve. From this curve we see that our best model CNN2 with an average CSI score of 0.39 predicting 0.9mm/hr rainfall is not far from ROVER with a CSI of 0.44 and Conv-LSTM with a CSI of 0.49. We also note that predicting a higher rainfall rate is tougher than predicting a lower rainfall rate because there are more number of cases of the lower rainfall rate. We can thus conclude that CNN2 is on par with the 1 hour ahead predictions made by ROVER.

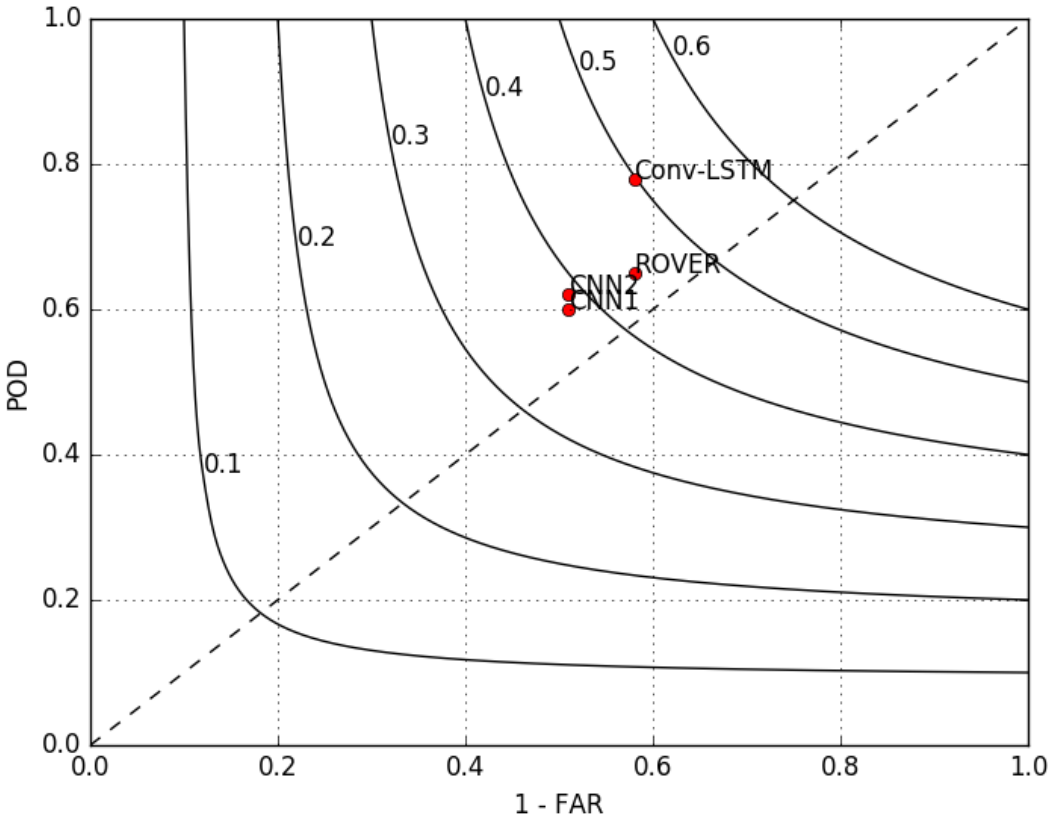

Figure 5.7: Performance curves of our CNN1 and CNN2 model evaluated at a prediction made for 0.9 mm/hr rainfall compared with Conv-LSTM and ROVER prediction made at 0.5mm/hr rainfall for a 1 hour nowcast. The solid contour lines represent CSI values

Table 5.7: CNN1 performance metrics for individual storms. The storm date is recorded in UTC and performance metrics are measured for a number of days for all points. As seen, the algorithm is very good on some days, very poor on others. The general thing we notice is the poor days correspond to days with a lot of "bad" NIPW data.

Storm dates	number of days	I + R			R		
		POD	FAR	CSI	POD	FAR	CSI
05/08/14	2	0.30	0.41	0.25	0.20	0.48	0.17
05/12/14	2	0.59	0.21	0.51	0.52	0.23	0.45
05/24/14	3	0.34	0.29	0.30	0.32	0.32	0.28
06/09/14	1	0.42	0.29	0.36	0.56	0.36	0.42
06/13/14	1	0.25	0.43	0.21	0.30	0.53	0.22
06/23/14	1	0.27	0.60	0.19	0.24	0.61	0.17
07/03/14	1	0.39	0.14	0.36	0.43	0.17	0.39
07/14/14	4	0.34	0.29	0.30	0.34	0.26	0.30
07/28/14	4	0.25	0.40	0.21	0.18	0.35	0.16
08/11/14	1	0.24	0.49	0.20	0.28	0.35	0.24
08/16/14	4	0.35	0.43	0.28	0.25	0.34	0.22
08/29/14	1	0.56	0.35	0.43	0.62	0.37	0.45

Table 5.8: CNN2 performance metrics for individual storms. The storm start date is recorded in UTC and measured for a number of days for all points.

Storm dates	number of days	I + R			R		
		POD	FAR	CSI	POD	FAR	CSI
05/08/14	2	0.40	0.49	0.29	0.24	0.48	0.20
05/12/14	2	0.66	0.25	0.54	0.59	0.23	0.50
05/24/14	3	0.44	0.36	0.35	0.35	0.36	0.29
06/09/14	1	0.47	0.35	0.38	0.60	0.35	0.46
06/13/14	1	0.30	0.48	0.24	0.41	0.54	0.28
06/23/14	1	0.20	0.66	0.15	0.32	0.56	0.23
07/03/14	1	0.42	0.15	0.39	0.54	0.27	0.45
07/14/14	4	0.44	0.37	0.35	0.42	0.31	0.35
07/28/14	4	0.28	0.43	0.24	0.28	0.42	0.23
08/11/14	1	0.34	0.40	0.28	0.27	0.34	0.24
08/16/14	4	0.33	0.39	0.27	0.21	0.33	0.19
08/29/14	1	0.63	0.35	0.47	0.56	0.32	0.44

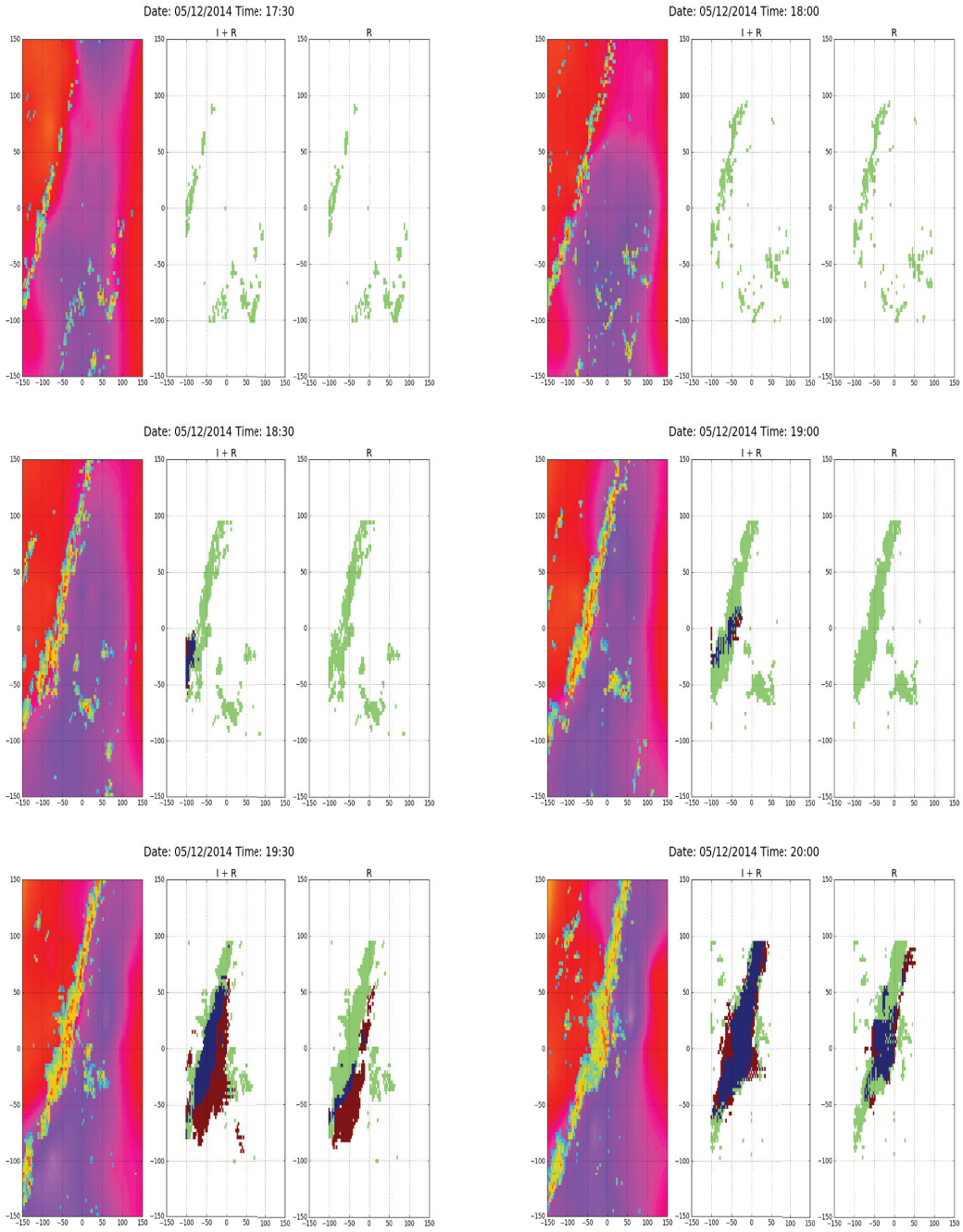

Figure 5.8: Comparison of I + R and R CNN2 models for storm on May 12th UTC where the I + R model did better than the R model. We can see clear water vapor convergence starting at 17:30 to 19:00 in the North-West of the prediction domain which eventually leads to a better prediction at 20:00 by the I + R model compared with the R model. Blue = hits, green = misses, red = false alarms.

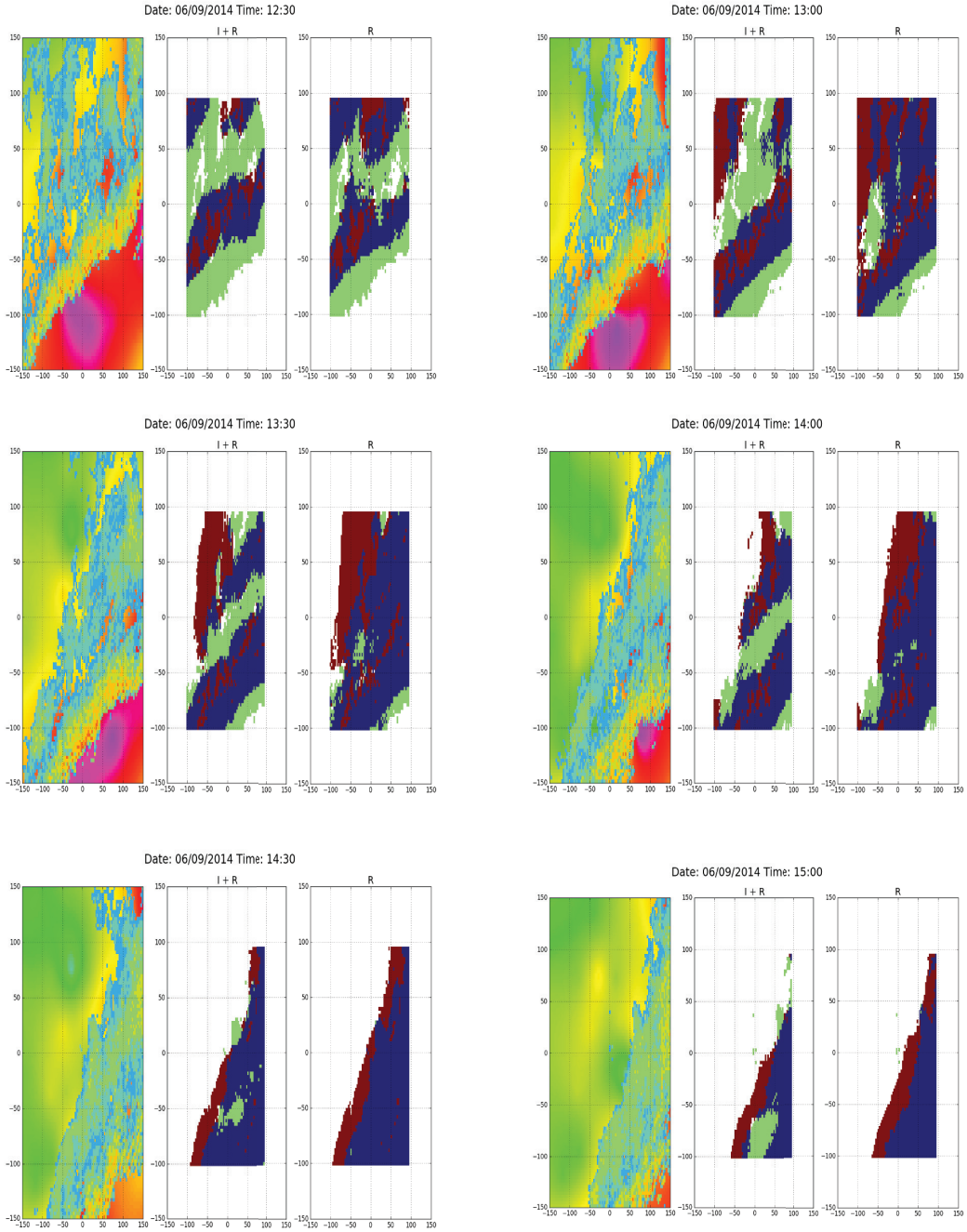

Figure 5.9: Comparison of I + R and R CNN2 models for storm on June 9th UTC where the R model did better than the I + R model. Large fluctuations in the IPW values viewed at each station creates holes in the NIPW maps which leads to discontinuous predictions made by the I + R model seen at 14:00 and 14:30 UTC by the holes in the prediction. This is due to the noise added by the NIPW fields to the I + R model which degrades the prediction. Blue = hits, green = misses, red = false alarms.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

This thesis looked at the 1-hour ahead precipitation nowcasting problem. This is a difficult problem for nowcasting algorithms based on the extrapolation of weather radar data because weather radars can only detect where it is actively raining and cannot predict the atmospheric variables that predict rain-cell formation, growth, and decay. As a result, the forecast skill of such algorithms quickly degrades after a few 10 minutes. A 1-hour nowcast is also difficult for nowcasting algorithms based on numerical weather prediction, largely due to the lack of input data at the spatial resolution needed to achieve the high-resolutions expected from precipitation nowcasting algorithms.

Different from traditional extrapolation and numerical weather prediction based techniques, in this thesis we adopted a machine learning approach. As inputs to our machine learning algorithm, we used histories of weather radar reflectivity and normalized integrated precipitable atmospheric water vapor (NIPW) fields. Radar reflectivity tells where it is currently raining. Sequences of radar reflectivity images give indications of advection rate and direction. NIPW fields, which we obtain by interpolating point NIPW measurements generated using the GPS-Meteorology technique from a network of GPS and ASOS (weather stations), gives the location and concentration of the atmospheric water vapor that might become rain. Sequences of NIPW fields give the flow of water vapor as well as the rate at which water vapor is concentrating. Since water must first exist in the atmosphere as vapor before it

becomes rain, our thesis was that the combination of NIPW and reflectivity could on one hand give the same high-resolution prediction as reflectivity extrapolation nowcasters, while extending its forecast horizon by providing the water vapor information indicative of storm initiation, growth, and decay.

For our domain, we used the 300 sq-km region centered on the Dallas Fort Worth (DFW) KFWS NEXRAD radar. Our reflectivity data came from the KFWS archives. Our NIPW data was generated from a local network of 44 GPS CORS stations and ASOS weather stations using the GPS-Met technique. The reflectivity and NIPW data was gridded onto the 300 sq-km domain at a spatial resolution of 3 km and temporal resolution of 30 minutes. Our data set, which covered the years 2014-2016 contained a number of severe convective rainfall events as well as a number of stratiform rainfall events.

6.1 Summary of Results

Two machine learning techniques were explored in this thesis, the Random Forest (RF) technique and the Convolutional Neural Network (CNN) technique. The RF technique was chosen as a baseline as it has been shown to work well on a number of other weather forecasting problems, e.g., aviation turbulence forecasting. The CNN technique was chosen, not only because it represents the current cutting edge machine learning technique, but also because it is explicitly designed for learning relationships in image based data.

For the RF the input consisted of a hand tailored vector of 26 features related to reflectivity and NIPW intensity, spatial variation, and rate of change. For the CNN the input consisted of the reflectivity and NIPW fields observed over the past 2 hours. The learning goal in each case was to predict rain vs. no rain at each grid point in the DFW domain. Thus, whereas the RF was explicitly given the variables

that seem important in predicting precipitation, the CNN was expected to learn both the temporal and spatial relationships between reflectivity, NIPW and rain.

To test the thesis that water vapor information can improve 1-hour precipitation nowcasts, we conducted what might be referred to as "data denial" studies. In these studies we trained and tested our machine learning algorithms first using reflectivity alone (R) and then using both NIPW and reflectivity (I + R).

The best random forest model with reflectivity alone gives an average CSI score of 0.32 and the NIPW and reflectivity gives an average CSI score of 0.33. The best convolutional neural network CNN2 gives an average CSI of 0.38 for the reflectivity only model and an average CSI of 0.39 for the reflectivity and NIPW model.

From this we are unable to conclude that NIPW improves nowcaster performance. From our experiments relating the performance improvement due to NIPW and the quality of the NIPW fields, we believe that our inconclusive results may be a result of poor NIPW data quality.

As for machine learning techniques, the deep learning CNN technique outperformed the RF technique. This was expected, as the deep learning CNN techniques are designed explicitly to extract predictive features from image-based data. Also expected was that training a CNN would be much more complicated than training a RF. The RF learns on a 26 dimensional feature vector which takes up 250MB of space and can be trained using a 32 core machine in 30 minutes. The data set for the CNN however occupies 80GB of space and takes 12 hours to train using the powerful state-of-the-art Tesla k-80 GPU.

Finally, comparing the CSI scores of our CNN model with state-of-the-art nowcasters from the literature indicates that we are on par with the best 1-hour ahead nowcasters such as ROVER.

6.2 Summary of Contributions

A significant portion of our work is highly experimental in that we developed a software system which ingests GPS derived Integrated Precipitable Water vapor and radar reflectivity data to nowcast precipitation in the DFW area 1 hour into the future. The software system was built to answer the question if realtime water vapor data from a network of GPS stations can provide additional information to improve radar reflectivity based nowcast algorithms. We showed that there is a minor performance benefit from our "data denial" experiments when NIPW is added as input to the machine learning based nowcaster. We also pointed out that improvements can be made to the water vapor and reflectivity based nowcast algorithm by improving on the quality of IPW estimates alone. The following summarizes the contributions of thesis.

1. A hardware/software design for a low-cost GPS-Met station capable of near-real-time IPW estimates and use in a high density GPS network. This work is documented in the publication [1].

2. Deployment of two low-cost GPS-Met stations in the DFW metroplex that include high-resolution barometers for the DFW Weather Forecast Office and other researchers to use for IPW and fine resolution barometric pressure analysis during events such the passing of nearby tornados and other weather anomalies.

3. Python based suite of tools for near-real-time estimates of IPW fields from the network of GPS-receivers and NWS ASOS weather stations in the DFW region. This set of tools is capable of automatically downloading the necessary GPS and ASOS data from on-line databases, putting the data in appropriate RINEX formats and directories, executing GAMIT for IPW estimation, and then calling a multiquadric interpolation algorithm to map the results to a field.

4. An improved understanding of the relationships between IPW and precipitation, including the ability to make movies to show visually the joint spatial-temporal evolution of IPW and weather radar reflectivity.

5. Application and analysis of random forest and convolution neural networks to the precipitation nowcasting problem using multiple weather variables (i.e. water vapor and radar reflectivity).

6. Development of Python-based tools to analyze and conduct machine-learning experiments with spatial-temporal fields of weather data that others will be able to use for future weather forecasting and analysis studies.

7. Development, analysis, and comparison of different machine learning approaches for precipitation nowcasting from spatial-temporal sequences of normalized IPW and weather radar reflectivity, including random forests and convolutional neural networks.

8. Develop and benchmarking a dataset of NIPW and reflectivity fields from several storms in the DFW area from 2014-2016.

9. We open-source our code which provides a modular framework to try different things such as different models etc. Since the bulk of the work went into the preprocessing future researchers can concentrate on building and evaluating different models. The code can be found in my github repository[1]. All of our results are reproducible with code and instructions provided in this repository.

10. Experimenting with a novel framework of the CNN for nowcasting applications. Our contribution is the idea and results obtained by using separate convolutional

[1]https://github.com/adityanagara/deep_nowcaster

for the two weather variables and concatenating the features learnt by these layers before passing them to the fully connected layer.

6.3 Recommendations for Future Work

We took a systems approach in building the nowcasting algorithm, where the entire nowcasting system was broken down into various subsystems. These subsystems can be improved independently to increase the performance of the systems as a whole. In this section we enumerate on the improvements that can be made to the individual subsystems.

6.3.1 Estimating IPW from GAMIT

Our results that IPW should improve precipitation nowcasting were largely inconclusive. We feel this is because of poor quality IPW data and that with better quality IPW data our results would have been different. This improvement can be made in GAMIT software package to generate the IPW values. To infer the water vapor from equation 2.8 GAMIT first needs to compute the ZTD from raw GPS pseudo-ranges. The ZTD is either computed as a constant for each station per session or is computed as a piece-wise linear function over a session [26]. The ZTD parameters computed for this thesis is for every 2 hours. The ZHD are then estimated from the pressure observations every 30 minutes. To get the water vapor observations as a true function of the GPS signal delays, we can estimate the ZTD values themselves every 30 minutes or even every 5 minutes. In GAMIT this can be done by increasing the "Number Zen" parameter in the tables/sestbl. session file which is currently set to a value of 25. Obtaining an estimate of ZTD at a higher resolution comes at the expense of additional computational cost.

Strong a-priori station coordinates is key to better estimates of IPW. Since we processed 3 years worth of data for the 44 GPS stations in DFW, we can compute

a stronger a-priori station coordinates for each station by averaging the location estimate from the three years. This would increase the accuracy of the IPW estimate at each station as a result of more precise station coordinates. There is a simple script in GAMIT which would allow us to estimate the position from all the GAMIT files produced from the three years.

GAMIT outputs the value of IPW at a given sampling interval along with the uncertainty in that measurement. The uncertainty or standard deviation usually ranges between the values of 0.05 mm to 2.0 mm. This uncertainty can be used as a criteria to determine the bad values of IPW which abruptly increase or decrease causing noise in the data set.

6.3.2 IPW Normalization

The normalization technique we tried only accounted for height and seasonal variation. Other normalization can take into account temperature, which we have seen affects the saturation point of IPW in the atmosphere. Other techniques for normalizations such as by pressure can also be explored as well such as the CAMR proposed by [40].

Normalization techniques using seasonal mean and standard deviations may yield better results, rather than monthly means and standard deviations as in this thesis. The seasonal means are now available for the storm seasons from 3 years (2014 - 2016). Normalizing by the seasonal mean and standard deviation can improve the statistical efficiency, where the NIPW representing all storm dates are varying at the same scale.

6.3.3 Interpolation techniques

In this thesis we used an interpolation technique to bring the point measurements of IPW to the same spatial sampling interval of the reflectivity fields. Specifically we used the Multiquadric interpolation technique which is a deterministic approach.

Several other stochastic based approach such as the Kringing interpolation can be used which rely on historical data at each point to get a better interpolation. There has been a recent study of evaluating various interpolation technique to build fields from point IPW estimates [54]. The aim of the study was to determine a good interpolation technique to use for an IPW based nowcasting system.

6.3.4 Nowcasting algorithm formulation

There are many ways in which the problem formulation could be changed to improve the performance of the nowcasts. As a first step one could predict a window of pixels similar to [31] rather than a single pixel point. Another technique to increase the complexity of the predictions is to predict a sequence of reflectivity values as done by most of the precipitation nowcasting algorithms in literature. In the machine learning framework this is known as the "structured prediction" formulation where the structure in the output sequence is exploited by the model to make better predictions. This approach was taken by the Conv-LSTM nowcasting algorithm[60].

6.3.5 Building a Larger Dataset

To explore more complex models would require more data. This can easily be achieved by increasing the size of the domain or predicting the reflectivity at a finer temporal resolution. Finer temporal resolution of IPW can be obtained by methods proposed in Section 6.3.1. Increasing the domain may require more complex modifications such as incorporating reflectivity data from multiple radars to measure the rainfall in a larger domain.

APPENDIX

DFW PRECIPITABLE WATER VAPOR, REFLECTIVITY NETWORK

This appendix gives the locations of the sensor assets used for the studies in this thesis.

Weather radar reflectivity data came from the KFWS WSR-88D weather radar located in Fort-Worth, TX (32.569 Deg Lat, -97.299 Deg Lon). This radar defined the center point of our 300 km by 300 km region of study. The table in Figure A.1 gives the locations of the GPS stations used along with the location of the closest ASOS weather station from which the met variables needed for IPW estimation were obtained. Table A.1 gives the location of the long baseline stations used for the double differencing processes.

Table A.1: Long baseline stations

Site ID	Location	Latitude(d)	Longitude(d)	Height(m)
AC20	Girdwood, Alaska, USA	60.92	-149.35	43.73
CONZ	Concepcion, Chile	-36.84	-73.07	176.22
P019	Fairfield, Idaho, USA	43.30	-115.31	1682.39
UNBJ	University of New Brunswick, Canada	45.95	-66.64	22.8

GPSid	GPSlat	GPSlong	GPSheight	ASOSid	ASOSlat	ASOSlong	ASOSheight	GPSgeoid	GPSMSL	ASOS_GPS	Network
okar	34.1684639	-97.169244	235.8	K1F0	34.15	-97.11	257	-26.045	261.845	-4.845	net1
okdn	34.4793111	-97.966553	314.3	KDUC	34.466	-97.96	339	-25.651	339.951	-0.951	net1
txbn	33.6067278	-96.175331	160.5	KDUA	33.95	-96.4	213	-26.224	186.724	26.276	net1
txbt	31.03265	-97.479008	177.1	KTPL	31.133	-97.4	208	-27.053	204.153	3.847	net1
txbu	30.7504611	-98.184386	438	KBMQ	30.733	-98.23	389	-24.668	462.668	-73.668	net1
txbx	30.7178389	-96.396622	83.6	KCFD	30.716	-96.33	112	-26.409	110.009	1.991	net1
txc2	30.8765028	-96.972342	97	KT35	30.883	-96.96	123	-26.074	123.074	-0.074	net1
txc3	31.8098139	-99.422094	498.3	KCOM	31.833	-99.4	517	-25.751	524.051	-7.051	net1
txck	31.3226333	-95.435908	87.5	KDKR	31.3	-95.4	106	-26.947	114.447	-8.447	net1
txco	33.1652639	-96.627944	161.9	KTKI	33.183	-96.58	179	-25.697	187.597	-8.597	net1
txda	32.7999861	-96.672914	160.6	KDAL	32.85	-96.85	158	-26.043	186.643	-28.643	net1
txdc	33.2362222	-97.608672	255.3	KLUD	33.25	-97.58	319	-27.449	282.749	36.251	net2
txde	33.2104528	-97.162778	178.8	KDTO	33.2	-97.2	196	-26.683	205.483	-9.483	net2
txea	32.4027694	-98.808933	407.5	KBKD	32.716	-98.88	392	-28.408	435.908	-43.908	net2
txes	32.3696972	-96.862775	163.7	KJWY	32.45	-96.91	217	-26.595	190.295	26.705	net2
txge	33.1319667	-96.055539	134.9	KGVT	33.066	-96.06	163	-25.57	160.47	2.53	net2
txgl	31.4721722	-98.567969	459.8	KMNZ	31.666	-98.15	396	-25.601	485.401	-89.401	net2
txgr	32.2403861	-97.754444	177.5	KGDJ	32.45	-97.81	237	-27.896	205.396	31.604	net2
txhi	31.9892111	-97.129797	153.8	KINJ	32.083	-97.1	209	-27.093	180.893	28.107	net2
txhm	31.6994889	-98.106747	345.1	KMNZ	31.666	-98.15	396	-26.766	371.866	24.134	net2
txja	33.1948222	-98.145625	326	KXBP	33.183	-97.83	260	-28.045	354.045	-94.045	net2
txka	32.5717556	-96.314278	115.9	KTRL	32.716	-96.26	145	-25.319	141.219	3.781	net2
txke	32.4097083	-97.323236	227.9	KCPT	32.35	-97.43	260	-27.506	255.406	4.594	net3
txmn	31.9100833	-97.661911	210.6	KACT	31.616	-97.23	151	-27.51	238.11	-87.11	net3
txmv	33.1618361	-95.221089	132.5	KOSA	33.1	-94.96	111	-26.649	159.149	-48.149	net3
txmw	32.8041694	-98.142889	246.4	KMWL	32.783	-98.06	284	-29.173	275.573	8.427	net3
txmx	31.5951222	-96.524375	119.7	KLXY	31.633	-96.51	166	-25.356	145.056	20.944	net3
txna	32.0417833	-96.538744	105.4	KCRS	32.033	-96.4	133	-25.57	130.97	2.03	net3
txno	33.7756556	-97.726028	258.3	K0F2	33.6	-97.78	336	-26.319	284.619	51.381	net3
txol	33.3560139	-98.7497	331.8	KRPH	33.116	-98.55	342	-28.487	360.287	-18.287	net3
txpa	33.6742361	-95.557025	145	KSLR	33.166	-95.61	149	-27.029	172.029	-23.029	net3
txpi	31.7244944	-95.594925	124.6	KPSN	31.783	-95.7	129	-26.423	151.023	-22.023	net3
txru	31.7849028	-95.126172	146	KJSO	31.866	-95.21	206	-26.621	172.621	33.379	net3
txsg	32.8557167	-97.344172	181.7	KFTW	32.833	-97.36	214	-27.801	209.501	4.499	net4
txsr	33.5915722	-96.607003	194.3	KGYI	33.716	-96.66	228	-25.829	220.129	7.871	net4
txst	32.2325917	-98.182194	376.6	KSEP	32.216	-98.16	402	-27.857	404.457	-2.457	net4
txsy	33.6024333	-99.258442	367.3	KCWC	33.85	-98.48	305	-28.398	395.698	-90.698	net4
txta	30.564225	-97.445042	147.7	KGTU	30.683	-97.68	240	-26.159	173.859	66.141	net4
txth	33.1789694	-99.1679	371.9	KBKD	32.716	-98.88	392	-29.062	400.962	-8.962	net4
txty	32.2496167	-95.393611	120.1	KTYR	32.366	-95.4	165	-26.258	146.358	18.642	net4
txwa	31.5777167	-97.110511	101.7	KACT	31.616	-97.23	151	-26.844	128.544	22.456	net4
txwe	32.7588972	-97.823478	337.4	KMWL	32.783	-98.06	284	-28.658	366.058	-82.058	net4
txwf	33.853925	-98.505556	280.2	KSPS	33.983	-98.5	308	-27.701	307.901	0.099	net4
zfw1	32.83065	-97.066469	155.2	KDFW	32.9	-97.01	174	-27.252	182.452	-8.452	net4

Figure A.1: GPS stations and ASOS ststions table.

BIBLIOGRAPHY

[1] A, Nagarajan, MC, Jacques, A, Lagace, DL, Pepyne, M, Zink, DJ, and McLaughlin. Lower-cost gps met station design for use in dense network slant path gps-met estimates of tropospheric wet delay and precipitable water vapor. In *Proceedings of 19th Conference on Integrated Observing and Assimilation Systems for the Atmosphere, Oceans, and Land Surface (IOAS-AOLS)* (2015), AMS.

[2] Akilan, A, Azeez, KK Abdul, Balaji, S, Schuh, H, and Srinivas, Y. Gps derived zenith total delay (ztd) observed at tropical locations in south india during atmospheric storms and depressions. *Journal of Atmospheric and Solar-Terrestrial Physics 125* (2015), 1–7.

[3] Alber, Chris, Ware, Randolph, Rocken, Christian, and Braun, John. Obtaining single path phase delays from gps double differences. *Geophysical Research Letters 27*, 17 (2000), 2661–2664.

[4] Arel, Itamar, Rose, Derek C, and Karnowski, Thomas P. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *Computational Intelligence Magazine, IEEE 5*, 4 (2010), 13–18.

[5] Bai, Zhengdong, and Feng, Yanming. Gps water vapor estimation using interpolated surface meteorological data from australian automatic weather stations. *Journal of Global Positioning Systems 2*, 2 (2003), 83–89.

[6] Barry, Roger G, and Chorley, Richard J. *Atmosphere, weather and climate*. Routledge, 2009.

[7] Batelaan, PD, Sato, T, Slobin, SD, and Reilly, H. Development of a water vapor radiometer to correct for tropospheric range delay in dsn applications. *DSN Progress Report 42 33* (1976), 77–84.

[8] Bengio, Ian Goodfellow Yoshua, and Courville, Aaron. Deep learning. Book in preparation for MIT Press, 2016.

[9] Bevis, Michael, Businger, Steven, Chiswell, Steven, Herring, Thomas A, Anthes, Richard A, Rocken, Christian, and Ware, Randolph H. Gps meteorology: Mapping zenith wet delays onto precipitable water. *Journal of applied meteorology 33*, 3 (1994), 379–386.

[10] Bevis, Michael, Businger, Steven, HERRING, THOMASA, Rocken, Christian, ANTHES, RICHARDA, and WARE, RANDOLPHH. Gps meteorology- remote sensing of atmospheric water vapor using the global positioning system. *Journal of Geophysical Research 97*, D14 (1992), 15787–15801.

[11] Bock, Y, Behr, J, Fang, P, Dean, J, and Leigh, R. Scripps orbit and permanent array center (sopac) and southern californian permanent gps geodetic array (pgga). *The Global Positioning System for the Geosciences* (1997), 55–61.

[12] Breiman, Leo. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[13] Bringi, VN, and Chandrasekar, V. *Polarimetric Doppler weather radar: principles and applications*. Cambridge University Press, 2001.

[14] Cheung, P, and Yeung, HY. Application of optical-flow technique to significant convection nowcast for terminal areas in hong kong. In *The 3rd WMO International Symposium on Nowcasting and Very Short-Range Forecasting (WSN12)* (2012), pp. 6–10.

[15] Davis, JL, Herring, TA, Shapiro, II, Rogers, AEE, and Elgered, Gunnar. Geodesy by radio interferometry: Effects of atmospheric modeling errors on estimates of baseline length. *Radio science 20*, 6 (1985), 1593–1607.

[16] de Haan, Siebren, Holleman, Iwan, and Holtslag, Albert AM. Real-time water vapor maps from a gps surface network: Construction, validation, and applications. *Journal of Applied Meteorology and Climatology 48*, 7 (2009), 1302–1316.

[17] Dietterich, Thomas G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (2000), Springer, pp. 1–15.

[18] Doviak, Richard J, and Zrnic, Dusan S. *Doppler Radar & Weather Observations*. Dover, 1993.

[19] Dow, John M, Neilan, RE, and Rizos, C. The international gnss service in a changing landscape of global navigation satellite systems. *Journal of Geodesy 83*, 3-4 (2009), 191–198.

[20] Duan, Jingping, Bevis, Michael, Fang, Peng, Bock, Yehuda, Chiswell, Steven, Businger, Steven, Rocken, Christian, Solheim, Frederick, van Hove, Terasa, Ware, Randolph, et al. Gps meteorology: Direct estimation of the absolute value of precipitable water. *Journal of Applied Meteorology 35*, 6 (1996), 830–838.

[21] FORSYTHE, JOHN M, KIDDER, STANLEY Q, FUELL, KEVIN K, LEROY, ANITA, JEDLOVEC, GARY J, and JONES, ANDREW S. A multisensor, blended, layered water vapor product for weather analysis and forecasting. *Journal of Operational Meteorology 3*, 5 (2015).

[22] French, Mark N, Krajewski, Witold F, and Cuykendall, Robert R. Rainfall forecasting in space and time using a neural network. *Journal of hydrology 137*, 1 (1992), 1–31.

[23] Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, vol. 1. Springer series in statistics Springer, Berlin, 2001.

[24] Grumm, Richard H, and Hart, Robert. Standardized anomalies applied to significant cold season weather events: Preliminary findings. *Weather and forecasting 16*, 6 (2001), 736–754.

[25] Gurtner, Werner, and Estey, Lou. Rinex-the receiver independent exchange format-version 3.00. *Astronomical Institute, University of Bern and UNAVCO, Bolulder, Colorado.* (2007).

[26] Herring, TA, King, RW, and McClusky, SC. Gamit reference manual. *GPS Analysis at MIT, release 10* (2015), 36.

[27] Hoffman-Wellenhof, B, Lichtenegger, Herbert, and Wasle, Elmar. Gnss?global navigation satellite systems. *GPS, GLONASS, Galileo and more. Wien: Springer-Verlag* (2008).

[28] Inoue, Hanako Y, and Inoue, Toshiro. Characteristics of the water-vapor field over the kanto district associated with summer thunderstorm activities. *SOLA 3* (2007), 101–104.

[29] Iwasaki, Hiroyuki, and Miki, Takahiro. Diurnal variation of convective activity and precipitable water over the" semi-basin". preliminary study on the mechanism responsible for the evening convective activity maximum. *????. ? 2 ? 80*, 3 (2002), 439–450.

[30] Johnson, JT, MacKeen, Pamela L, Witt, Arthur, Mitchell, E De Wayne, Stumpf, Gregory J, Eilts, Michael D, and Thomas, Kevin W. The storm cell identification and tracking algorithm: An enhanced wsr-88d algorithm. *Weather and Forecasting 13*, 2 (1998), 263–276.

[31] Klein, Benjamin, Wolf, Lior, and Afek, Yehuda. A dynamic convolutional layer for short range weather prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 4840–4848.

[32] Kuligowski, Robert J, and Barros, Ana P. Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. *Weather and Forecasting 13*, 4 (1998), 1194–1204.

[33] LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[34] Louppe, Gilles. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502* (2014).

[35] Lynch, Peter. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics 227*, 7 (2008), 3431–3444.

[36] Marshall, John S, and Palmer, W Mc K. The distribution of raindrops with size. *Journal of meteorology 5*, 4 (1948), 165–166.

[37] McGovern, Amy, Supinie, TIMOTHY, Gagne, II, Collier, M, Brown, RA, Basara, J, and Williams, J. Understanding severe weather processes through spatiotemporal relational random forests. In *2010 NASA conference on intelligent data understanding (to appear)* (2010).

[38] Mecikalski, John R, Williams, John K, Jewett, Christopher P, Ahijevych, David, LeRoy, Anita, and Walker, John R. Probabilistic 0–1-h convective initiation nowcasts that combine geostationary satellite observations and numerical weather prediction model data. *Journal of Applied Meteorology and Climatology 54*, 5 (2015), 1039–1059.

[39] Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[40] Radhakrishna, Basivi, Fabry, Frédéric, Braun, John J, and Van Hove, Teresa. Precipitable water from gps over the continental united states: Diurnal cycle, intercomparisons with narr, and link with convective initiation. *Journal of Climate 28*, 7 (2015), 2584–2599.

[41] Remondi, Benjamin W. Using the global positioning system(gps) phase observable for relative geodesy: Modeling, processing, and results[ph. d. thesis].

[42] Rocken, Christian, Hove, Teresa Van, Johnson, James, Solheim, Fred, Ware, Randolph, Bevis, Mike, Chiswell, Steve, and Businger, Steve. Gps/storm-gps sensing of atmospheric water vapor for meteorology. *Journal of Atmospheric and Oceanic Technology 12*, 3 (1995), 468–478.

[43] Roebber, Paul J. Visualizing multiple measures of forecast quality. *Weather and Forecasting 24*, 2 (2009), 601–608.

[44] Rumelhart, David E, Durbin, Richard, Golden, Richard, and Chauvin, Yves. Backpropagation: The basic theory. *Backpropagation: Theory, Architectures and Applications* (1995), 1–34.

[45] Ruzanski, Evan, Chandrasekar, V, and Wang, Yanting. The casa nowcasting system. *Journal of Atmospheric and Oceanic Technology 28*, 5 (2011), 640–655.

[46] Saastamoinen, J. Atmospheric correction for the troposphere and stratosphere in radio ranging satellites. *The use of artificial satellites for geodesy* (1972), 247–251.

[47] Seko, Hiromu, Nakamura, Hajime, Shoji, Yoshinori, and Iwabuchi, Tetsuya. The meso-. gamma. scale water vapor distribution associated with a thunderstorm calculated from a dense network of gps receivers. *????. ? 2 ? 82*, 1B (2004), 569–586.

[48] Shangguan, Ming. *Analysis and derivation of spatial and temporal distribution of water vapor from GNSS observations*. PhD thesis, Technische Universität Berlin, 2014.

[49] Shi, Junbo, Xu, Chaoqian, Guo, Jiming, and Gao, Yang. Real-time gps precise point positioning-based precipitable water vapor estimation for rainfall monitoring and forecasting. *Geoscience and Remote Sensing, IEEE Transactions on 53*, 6 (2015), 3452–3459.

[50] Shoji, Yoshinori, Yamauchi, Hiroshi, Mashiko, Wataru, and Sato, Eiichi. Estimation of local-scale precipitable water vapor distribution around each gnss station using slant path delay. *SOLA 10*, 0 (2014), 29–33.

[51] Snay, Richard A, and Soler, Tomás. Continuously operating reference station (cors): history, applications, and future enhancements. *Journal of Surveying Engineering 134*, 4 (2008), 95–104.

[52] Spilker, JJ. Gps signal structure and performance characteristics. *Global Positioning System 1* (1980), 29–54.

[53] Srivastava, Nitish, Hinton, Geoffrey E, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 1 (2014), 1929–1958.

[54] Suparta, Wayan, and Rahman, Rosnani. Spatial interpolation of gps pwv and meteorological variables over the west coast of peninsular malaysia during 2013 klang valley flash flood. *Atmospheric Research 168* (2016), 205–219.

[55] Sutskever, Ilya, Martens, James, Dahl, George E, and Hinton, Geoffrey E. On the importance of initialization and momentum in deep learning. *ICML (3) 28* (2013), 1139–1147.

[56] Tabios, Guillermo Q, and Salas, Jose D. A comparative analysis of techniques for spatial interpolation of precipitation1, 1985.

[57] Terradellas, E, and Téllez, B. The use of products from ground-based gnss observations in meteorological nowcasting. *Advances in Geosciences 26*, 26 (2010), 77–82.

[58] Tralli, David M, and Lichten, Stephen M. Stochastic estimation of tropospheric path delays in global positioning system geodetic measurements. *Bulletin géodésique 64*, 2 (1990), 127–159.

[59] Wolfe, Daniel E, and Gutman, Seth I. Developing an operational, surface-based, gps, water vapor observing system for noaa: Network design and results. *Journal of Atmospheric and Oceanic Technology 17*, 4 (2000), 426–440.

[60] Xingjian, SHI, Chen, Zhourong, Wang, Hao, Yeung, Dit-Yan, Wong, Wai-kin, and Woo, Wang-chun. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems* (2015), pp. 802–810.

[61] Yoshinori, SHOJI. Retrieval of water vapor inhomogeneity using the japanese nationwide gps array and its potential for prediction of convective precipitation. *????. ? 2 ? 91*, 1 (2013), 43–62.

[62] Zahraei, Ali, Hsu, Kuo-lin, Sorooshian, Soroosh, Gourley, JJ, Lakshmanan, Valliappa, Hong, Yang, and Bellerby, Tim. Quantitative precipitation nowcasting: a lagrangian pixel-based approach. *Atmospheric Research 118* (2012), 418–434.