THESIS

DEEP NEURAL NETWORK BASED RAIN/NO-RAIN CLASSIFICATION AND RAIN RATE ESTIMATION

Submitted by

Jay U. Potnis

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2022

Master's Committee:

Advisor: V. Chandrasekar

Margaret Cheney
Thomas Siller

ABSTRACT

DEEP NEURAL NETWORK BASED RAIN/NO-RAIN CLASSIFICATION AND RAIN RATE ESTIMATION

Quantitative Precipitation Estimation is the process of computing rainfall rate or rainfall accumulation based on the state of the atmosphere. Atmospheric conditions can be described by using observations from meteorological instruments. Extreme weather events caused due to high rainfall can be dangerous in terms of loss of property and life. To prevent such disasters, accurate QPE algorithms that analyze and estimate the amount of rainfall observed in a region are critical. Moreover, rain rate estimates are crucial products in making management decisions in water, energy, construction infrastructure, and many other institutions. Researching state-of-the-art rainfall estimation techniques that make use of reliable remote sensing equipment such as satellites and radars is important as deploying rain gauges everywhere is not possible and is not a viable option. As rain precipitation is a complicated phenomenon, depending on multiple factors in the atmosphere, research is being done in this domain for many decades and the goal is to improve the accuracy of estimation by using new state-of-the-art methods.

Weather radars are reliable remote sensing instruments that are used to capture the different properties of weather in form of products called moments. The goal of this work is to use weather radars in conjunction with Deep Neural Networks to provide solutions to multiple tasks in the QPE domain. Neural networks can be used for precipitation flagging such as classifying rain and no rain events. They can also be used for estimating the rain rates at specific coordinates or along regions. Though multiple empirical relationships between radar moments and rain rate already exist, this work provides good state-of-the-art alternatives to these equations and can even achieve comparable accuracy.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my Advisor Dr. Chandra for guiding me throughout the completion of this research. His immense knowledge, continuous motivation and guidance was the driving force for completion of this research. I would also like to thank my thesis committee members; Dr. Margaret Cheney and Dr. Thomas Siller for reviewing my research and taking time out of their schedule to attend this presentation. I would give special thanks to Sounak Biswas and EunYeol Kim for providing immense guidance and suggestions during this research. I would also like to express my deepest gratitude towards my Family for their continuous support throughout every stage of this Research. Without your support none of this would have been possible. Finally, I would like to thank everyone present here for attending my research presentation today.

This research was supported by NASA PMM program and also motivated by the AQPI program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Quantitative Precipitation Estimation (QPE) and Quantitative Precipitation Forecasting (QPF) have been two major research areas in the field of hydrometeorology for decades. In this thesis, QPE using Machine Learning techniques has been explored. QPE is the process of quantifying the amount of rainfall observed over a region. QPE is essential in hydrological analysis which is crucial for streamflow and flooding forecast, drought assessment, and numerical weather prediction [1]. There are several ground based in-situ observation instruments such as rain gauges and disdrometers that can accurately measure the amount of rainfall observed at a coordinate. However, oftentimes, it is non-practical and nonviable to deploy rain gauges over a large area especially in remote locations. To solve this problem, several remote sensing instruments like weather radars and satellites are being examined for their ability to derive meaningful QPE products. Rainfall cannot be directly measured by radars and satellite. Weather radars operate by emitting EM pulses and are able to measure the back scattered power and phase change of the echoes from hydrometeors. Satellites on the other hand are able to measure the radiances from Earth's surface and cloud tops in form of brightness temperatures. Therefore, it is an ongoing research area to accurately estimate rainfall from radar and satellite observations.

Weather radars have a high spatial and temporal resolution than precipitation measurement instruments like rain-gauges and disdrometers. The US National Weather Service (NWS) has several weather radar deployments all over the CONUS which are operational and aid in weather monitoring and forecasting. Moreover, NEXRAD Volume Coverage Pattern (VCP) scans provide 360-degree coverage of the atmosphere with a high range. Vertical Profiles of radar polarimetric variables can be constructed from VCP scans. Vertical Profiles of Reflectivity helps reveal the vertical structure of storm. They help to study how the storm microphysics at higher altitudes results in the surface observations. Reflectivity at a height and rainfall rate at the ground are known to have dependencies between them. This can be expressed as an empirical relation between the two.

However, there is a considerable amount of error introduced in such relationship due to parameterization. Moreover, measurement biases in Z and R can further limit estimation accuracy [2, 3]. Additionally, these relations are highly dependent on geographical regions, time of the year, and terrain height. In order to make these estimation techniques robust, it critical to adjust the estimator coefficients and correct biases in observations. A multitude of algorithms have been developed for improving radar-based QPE accuracy based on microphysics or Multiple Radar Multiple Sensor (MRMS) methods.

Machine Learning based methods recently gained more importance due to the advancements in data analytics and high-performance computing. This thesis uses Deep Neural Networks, a sub-branch of Machine Learning, to improve accuracy of radar based QPE compared to traditional algorithms such as DROPS, NEXRAD's Dual-Pol based QPE Product and MRMS [4] [5]. Deep Neural Network is a data-driven approach that uses high-dimensional space exploration using mathematics to discover the hidden relationships between the input and output variables. In this thesis, DNNs have been used to estimate the complex relationship between surface rainfall rate and vertical profiles of dual-pol measurements from the radar. As Machine learning requires a huge amount of data to achieve generalization of the output, the use of machine learning in many fields is restricted due to the lack of data availability. Weather radar data has been historically collected and well archived for several decades, Deep Neural Networks can harness this to achieve results and accuracies.

1.1 Problem Statement

The goal of this thesis is to demonstrate that Machine Learning can be used as a potential approach for addressing different problems related to QPE such as rain / no-rain classification and rain rate estimation. Machine learning methods can achieve good performance with a lesser number of sensors and radars by harnessing their ability to explore multidimensional data space to find optimum solution points. As Machine Learning is a data-driven approach, so data quality control is very important. In addition, optimizing the performance of the models by making use of appropriate model parameters and training hyperparameters is also demonstrated.

1.2 Research Objectives

This thesis makes these four main contributions that can help further research in DNN based weather radar QPE.

1. Implementation of Classification DNN model for precipitation flagging, i.e. classification of rain/no-rain condition on ground based on Vertical Profile of Radar.

2. Introducing the technique of using class weights in Binary Crossentropy (Log Loss) function to reduce effects of imbalanced dataset and improve Recall and F1 score.

3. Implementation of Regression DNN model for estimation of rain-rate on the ground based on Vertical Profile of Reflectivity, Differential Reflectivity and K_{dp}.

4. Using Classification DNN's output as a feature in Regression DNN model for improving the quality of rain-rate estimates.

1.3 Overview

This thesis starts with an Introduction and Problem Statement that helps in understanding the premise for the problem and the information needed to understand the work presented. The following chapter has a thorough literature review of all the existing and state-of-the-art techniques for Rain / No-Rain classification and Rain rate estimation based on Ground Radar moments. This helps in understanding the trends of development in QPE using weather radars. In the consecutive chapter, Research Methodology on data processing, DNN model training, DNN model implementation for rain/no-rain classification, and rain-rate estimation have been explained in detail. This section also compares the results of DNN models used in this thesis with baseline models. Finally, possible research ideas in continuation with this work have been discussed.

Chapter 2

Background

Cloud formation occurs because of the cooling and condensation of water vapors in the Earth's Atmosphere. When these small droplets condense onto one another, their size increases. When these drops become heavier for the atmosphere's vertical pressure to hold, they fall towards Earth's surface in the form of precipitation. This process is called collision-coalescence by which warm rain forms. The type and physical state of precipitation depend on the atmospheric conditions observed during that time. Precipitation can take liquid or solid forms such as rain, hail, snow, graupel, drizzle, etc. Precipitation can be broadly classified into two categories, Stratiform, and Convective. Stratiform precipitation can be observed when a warm front meets a cold mass of air. Stratiform Precipitation can be formed by ice nucleation and subsequent melting of ice crystals or it can form by collision coalescence. Therefore, stratiform precipitation (also called dynamic precipitation) produces huge stable and horizontally developed clouds. On the other hand, convective precipitation is observed when localized air is heated more than the surrounding air. Therefore, convective precipitation (also called thermodynamic precipitation) causes the formation of unstable vertical clouds called cumuliform. Stratiform precipitation is usually distributed over large regions with steady rainfall, whereas convective precipitation is localized and undergoes a frequent change in the intensity of the precipitation.

Accurate measurement of precipitation is essential for studying Earth's hydrological cycle. This helps in the prediction and mitigation of natural events like flooding and drought assessment. Moreover, QPE is also used in measuring or analyzing streamflow and has many uses in Numerical Weather Prediction (NWP) [1]. To estimate the amount of precipitation accurately, it is essential to observe the state of the atmosphere. This is can be done at local scales by using surface based in-situ observation instruments such as rain gauge, disdrometers, ceilometers, wind profilers etc. These instruments are used for measuring various quantities related to meteorology, precipitation, atmospheric turbulence near the earth's surface. On the other hand, remote sensing instruments are used for observation of the atmosphere over a larger extent. It is is very useful in locations where deployment of in-situ instruments are not feasible. Remote sensing of precipitation can be performed using ground based instruments such as weather radars, lidars as well as space based instruments such as satellites [6–11]. A collection of these helps in qualitative and quantitative analysis of the atmospheric properties. A few tasks in QPE are rain/no-rain classification (also known as rain flagging) and rainfall estimation. Rain detection and rainfall estimation are important tasks in the field of hydrometeorology which in turn impacts socio-economic life. Therefore, it is essential to research models that accurately estimate rain-rate. Rainfall is usually estimated using two metrics: rainfall rate and rainfall accumulation.

Rainfall estimation is done by measuring the quantity of rainfall observed in a region over a period of time. Rainfall Rate (RR) is usually measured in mm/h. When rain falls, its intensity is influenced by the type of precipitation (stratiform/convective), temperature, hydrometeors, raindrop size, cloud density, drop size distribution, and many more other factors. Therefore, accurate estimation of rainfall is a challenging task. The most accurate way of measuring rainfall accumulation is using rain gauges. Rain gauges are surface observation in-situ instruments that use tipping or weighing mechanisms to measure the actual rainfall observations. As rain gauges are in-situ instruments, they are only able to measure rainfall at a particular location. Oftentimes, deploying rain gauges over a large area is expensive and operationally nonviable especially in complex terrains. Moreover, large regions are usually sparsely populated with rain gauges which results in poor spatial resolution of the rainfall data. In addition, maintenance of rain gauge network can be very difficult owing to sensor faults and errors. To overcome these issues, weather radars and satellites are used which have high spatial resolution and can cover huge areas. However, radar and satellite observations are prone to measurement errors and bias which limit the accuracy of rainfall estimation.

Weather radars are one of the most common ways of analyzing the earth's atmosphere. They are echo-based remote sensing devices that emit radio waves of constant wavelength. Doppler weather radars work on the principle of scattering of the EM wave and measuring the back scattered echoes.

They transmit EM pulses when these waves hit an object in the sky, they are absorbed or scattered in different directions. The waves which are back scattered are measured by the radar. The intensity of the echo defines the size of the object as well as the distance of the object from the radar. When these waves are reflected, they experience a change in phase and frequency. The phase and frequency of the echoes are compared with that of the transmitted wave to come up with Doppler Velocity. Doppler Velocity is the radia; component of the velocity vector of the scattering object. A weather radar repeats this process over different elevation angles in a 360-degree direction. Each of these 360 degree sweep at a fixed elevation angle is called a Plan Position Indicator (PPI) scan. An accumulation of multiple such PPIs is called a Volume Coverage Pattern (VCP) scan. Just like a pixel is the smallest data point in an image, the smallest data point in a radar's VCP scan is called a gate. Since the beam expands with respect to the distance from the radar, the range volumes expand as they move away from the radar. Smaller range volumes result in accurate measurements whereas distant range volumes result in lower accuracy since the beam averages over a larger area..

The intensity of radar's transmitted power echoed back to the radar is called reflectivity. Reflectivity (Z) is measured in dBZ (decibels of Z). Reflectivity is the intensity of reflected energy at a single elevation scan of the radar. Reflectivity (dBZ) increases as the strength of the reflected power increases. Therefore, reflectivity can be associated with hydrometeor size. The lowest value of Z is caused by light fog, smoke, and ground clutter, whereas the highest values of Z are caused by large hail.

2.1 Empirical R-Z Relations

Base reflectivity, also called Z, is the mean returned power from the horizontal transmitted channel. Several relations between Z and RR have been discovered in the past. This variation in the relations is largely due to the ambiguity in Drop Size and Drop Size Distribution (DSD) of rainfall. One of the most famous models for DSD is the exponential expression discovered by Marshal-Palmer in 1948 .

$$N(D) = N_0 e^{-\Lambda D} \tag{2.1}$$

Here, D is the diameter of rain drop, N(D) is the distribution function of rain drops i.e. drop size density between D + dD and N_0 is the value of N(D) when D = 0. Λ is the slope parameter expressed by rain rate R (in mm hr^{-1})

A good value for Λ was found out by Marshal and Palmer which was applicable over a wide range of rain rates [12].

$$\Lambda = 41R^{-0.21}cm^{-1} \tag{2.2}$$

Lots of other values were also reported by [13]. Mathematically, Z can be expressed as the sixth power of the diameter of the hydrometeor over all the hydrometeors in a unit volume.

$$Z = \int_0^\infty N(D)D^6\, dD \tag{2.3}$$

Here, D: Diameter of rain drop in mm, N(D): Distribution function of the rain drops

Thus, if we use Marshal-Palmer's Distribution function in Z-R relation we get:

$$Z = \int_0^\infty D^6 N_0 e^{-\Lambda D} dD = 6!N_0\Lambda^{-7} \tag{2.4}$$

$$Z = 200R^{1.6} \tag{2.5}$$

There are several such possibilities of equations depending on the ambiguous value of Drop size distribution function obtained from the drop size density. Therefore this relation can also be commonly represented as:

$$Z = aR^b \tag{2.6}$$

Where a and b are constants of proportionality. Depending on various models analyzed from data, a can range between 127 to 505 and b can range from 1.41 to 2.39 [14].

2.2 Empirical R-Z_{dr}

Dual-Polarization radars like WSR-88D can produce horizontal reflectivity (Z_H) and vertical reflectivity (Z_V) measurements due to the dual polarized configuration of their emitted beams. Differential Reflectivity (Z_{dr}) is the difference between the mean returned powers of horizontal reflectivity and vertical reflectivity values and it is expressed in decibels (dB).

$$Z_{dr} = Z_H - Z_V \tag{2.7}$$

From a Logarithmic perspective, Z_{dr} is the ratio of horizontal polarized reflectivity to vertically polarized reflectivity in dB.

$$Z_{dr} = 10 \log \frac{Z_H}{Z_V} \tag{2.8}$$

Negative values of Z_{dr} indicate that the hydrometeors have a larger vertical size as compared to their horizontal dimension. Values close to zero mean that the hydrometeors are spherical. Positive values of Z_{dr} indicate that the hydrometeors are elongated in the horizontal dimension. Z_{dr} can be very useful in identifying hydrometeor shape size and orientation.. During rainfall, rain drops falling towards the ground have an oblate spheroid shape. From a Z_{dr} point of view, this means that a positive Z_{dr} would be observed when the radar points in the direction of falling rain drops. Thus, Z_{dr} can be used for determining drop size distribution and it can aid in rainfall accuracy by providing additional information on particle size.. [15] derives the relationship of rain rate from Z_{dr}. The equations are derived by substituting the relation of Z_H and Z_V into the Z_{dr} equation.

2.3 Empirical R-K_{dp} Relations

Along with the uncertainty in Drop size distribution, the biases and beam blockage while measuring Z also contribute to difficulty in QPE. Some of the common errors in Z measurement are radar calibration errors, attenuation, bright band contamination, anomalous propagation, beam blockage, and beam broadening due to range. However, the phase measurements are not affected by these limitations. Therefore, it has been documented in literature that rainfall estimation using K_{dp} can mitigate these practical issues [16]. K_{dp} is a derived from Differential Phase Shift (PHIDP). Dual Polarization Radars emit horizontally and vertically polarized waves. When these pulses pass through a hydrometeor, they get attenuated and result in a change in their phase. Smaller particles that are bigger across the horizontal dimension cause more scattering at the horizontal polarization than the vertical polarization. This difference in the phase shift between the horizontal and vertical polarized beams is called PHIDP. The extent of this phase shift is dependent on the size, concentration, and permittivity of hydrometeors. As higher rain rates are usually a result of a high concentration of oblate raindrops, higher rain rates experience more phase shifts.

To find out the location of bigger-sized hydrometeors resulting in higher phase shifts, one-half the range derivative of PHIDP is taken [16]. This derived product is called K_{dp}. Being phase measurement, K_{dp} is resistant to radar calibration errors, attenuation, and beam blockage [17]. One more advantage is that K_{dp} is less susceptible to variations in DSD therefore it also solves the Z-R relation's ambiguity in the drop size problem. These advantages of K_{dp} make it a good factor that can be used in the quantification of rain rates [18].

The relationship between K_{dp} and R is exponential. [19] derived the R-K_{dp} relationship based on Marshal-Palmer's DSD assumption.

$$R_{DP} = 20.35 K_{DP}^{0.8666} \tag{2.9}$$

Even though this relation is based on the Marshal-Palmer DSD model, it is expected to give better accuracy for estimating rain-rates as it is independent of variations in DSD.

Empirical relations of Z-R and R-K_{dp} prove that Z and K_{dp} are very important radar moments when it comes to QPE. The accuracy of these equations is restricted due to uncertainty in DSD and several errors and bias in measurements. Neural Network models are good in finding out hidden relations between such parameters and can automatically create smart adaptable models based on such hidden relations.

2.4 Machine Learning and Deep Neural Networks

Machine Learning as a definition is a subset of algorithms that are capable of task-based learning independently from inputs to outputs without human assistance on the explanation of the relation between the inputs and the outputs. The algorithm should be able to establish a connection between inputs and outputs just by looking at the dataset. Deep Neural Networks are a branch of Machine Learning which use structured modeles based on the human brain and the synaptic network of neurons. Most Machine Learning techniques require lesser computation whereas deep learning techniques consist of long computational structures consisting of micro calculations. Previously called Artificial Neural Networks, DNN are inspired based on the biological function of brain.

2.4.1 History of Deep Neural Networks

The first Neural Network was the McCulloch-Pitts neuron. Invented in 1943, McCulloch-Pitt's neuron was based on a linear model which was capable of recognizing two categories of inputs by testing whether the result of a function is positive or negative. This was not an adaptive model and hence the weights needed in this structure were set by a human. Later in 1950, the perceptron model was invented. The perceptron model was the first model that could adaptively learn weights that were required to make this categorization in McCulloch-Pitt's neuron. Invented in 1960, the Adaptive Linear Element (also known as ADALINE) was capable of predicting a real number based on inputs. The training algorithm used to adapt the weights of ADALINE was what we famously recognize as the Stochastic Gradient Descent algorithm today. The models used by Perceptron and ADALINE were linear, they could not learn more complex functions like the Exclusive-OR function. Several research contributions by several researchers added to this neuron and lead to the development of the Multi-Layer Perceptron Model that we know today. After 1960, rigorous experimentation and research in this field continued. This is mainly due to the exceptional improvement in the computational capability to support such computationally expensive Neural Structures. After 2000's lots of new activation functions, loss functions and

Neural Network Structures were invented. Currently, Neural Networks have not only excelled in analyzing structured data, but it is also showing great outcome in solving problems concerning unstructured data like image, speech, language, etc.

2.4.2 Deep Neural Networks

A Deep Neural Network consists of multiple layers each consisting of multiple neurons. Each neuron in the previous layer is connected to each neuron in the next layer. Neuron links are associated with weights and biases which help in facilitating the training ability of Neural Networks. Furthermore, every layer in the Neural Network is associated with an activation function. The simplest and default activation function being the linear activation function [20].

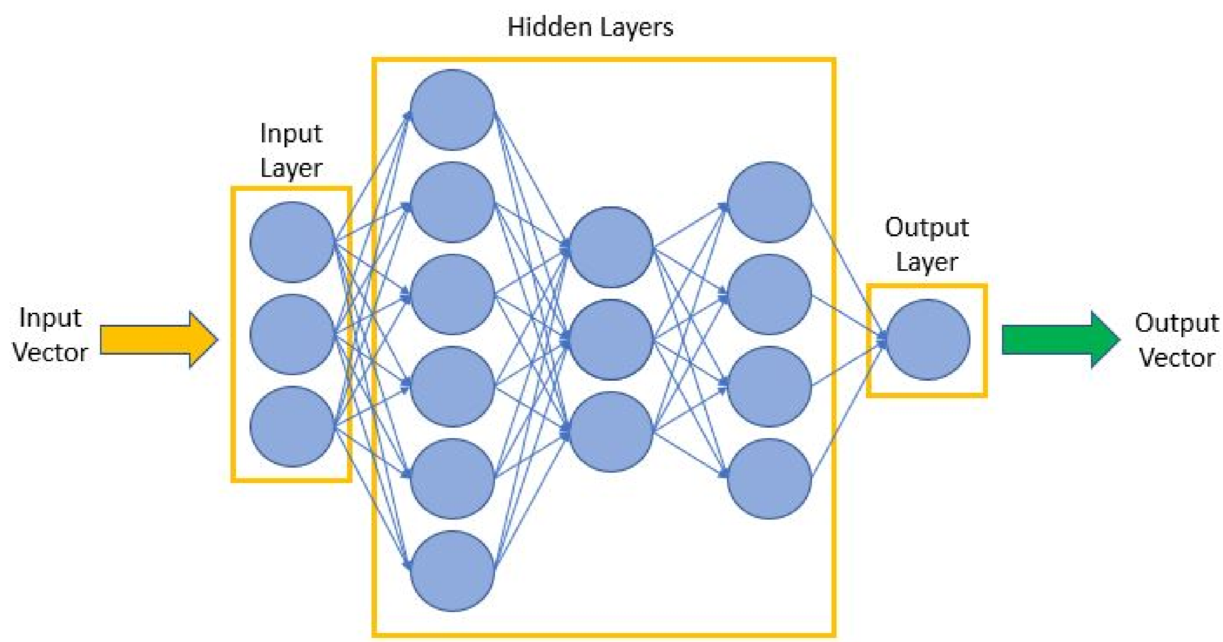


Figure 2.1: Deep Neural Network Architecture

A DNN model operates in 2 modes: training and inference. Initially, all the weights and biases in the DNN are set to random values. The random values are also set using defined strategies to prevent vanishing and exploding gradients problem [21]. The model learns to represent the output data in the training phase. The training phase is divided into two stages: forward propagation and backpropagation.

Forward Propagation

In forward propagation, inputs are transformed into outputs by going through computations governed by each layer and neuron based on their weight, bias, and activation functions. After an output is obtained, it is compared with the expected output value. The loss between the expected and actual value is calculated using the loss function of choice. This ends the forward propagation stage. Figure 2.2 shows the architecture of a single layer feed forward network, where x is the

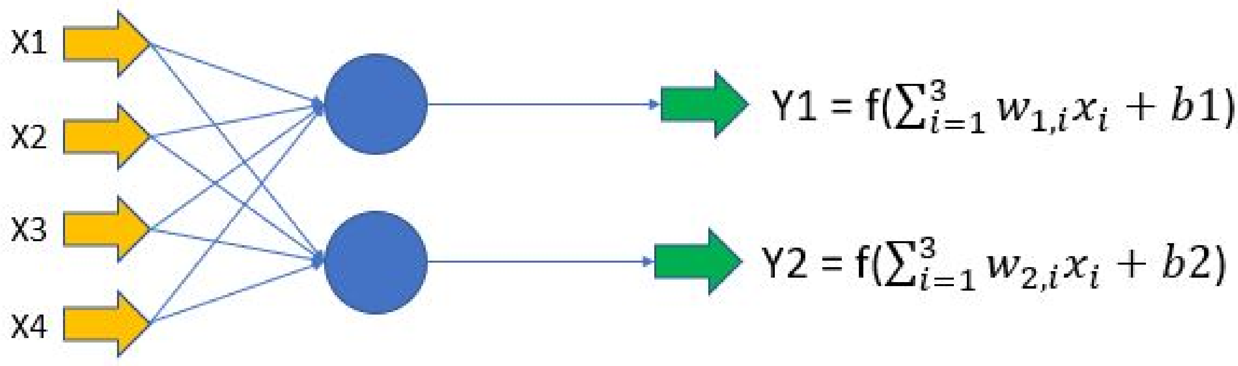


Figure 2.2: Single Layer Feedforward Neural Network

input vector, y is the output vector, $w_{l,i}$ are the weight values at layer l and neuron i respectively. b_l is the bias at layer l. General Equation of a Neuron's output from Figure 2.2 is given in 2.10 and 2.11. Where i is the number of neuron and l is the number of layers.

$$y = f(wx + b) \tag{2.10}$$

$$y = f(\sum_{i=1}^{n} w_{j,i} x_i + b_j) \tag{2.11}$$

Activation Functions

Just as the name suggests, the role of an activation function is to activate the value provided to the function. In simpler terms, it models the input of the function to derive an output governed by the activation function equation. Activation functions transform the summed up value obtained from a neuron and pass it to the next layer in the Neural Network or to the output of the Neural Network. Activation functions are broadly classified into two categories: linear and non-linear. Some of the commonly used activation functions are shown here along with their equations and curves.

1. Rectified Linear Unit (ReLU): $f(z) = max(0, z)$
2. Leaky ReLU: $f(z) = max(0.1z, z)$
3. Sigmoid: $f(z) = \frac{1}{1+e^{-z}}$
4. Hyperbolic tangent (tanh): $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

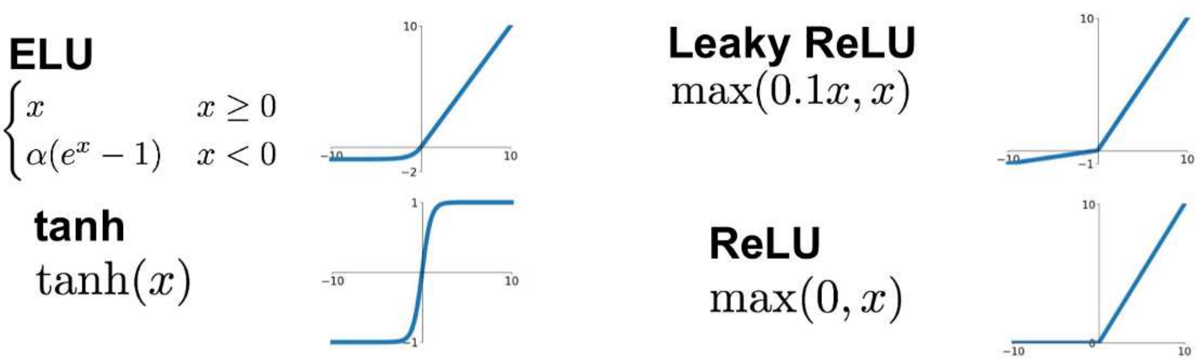


Figure 2.3: Common Activation Functions used in DNN

Figure 2.3 shows the commonly used non linear activation functions in machine learning.

Back Propagation

During the training phase, the forward propagation stage is always followed by a back propagation stage. In the back propagation stage, gradient values are calculated based on the error calculated from the forward propagation stage. These gradients are used to update the weights and biases in the model. One iteration of this process is called an epoch. This is repeated for several epochs until a minima is found in the design space. This completes the training phase of the Neural Network Model. The inference phase is nothing but the forward propagation step without back propagation. In the inference phase, the output is calculated based on the learned values of weights and biases. Figure 2.4 shows the flowchart of the back propagation procedure used in Deep Neural Networks.

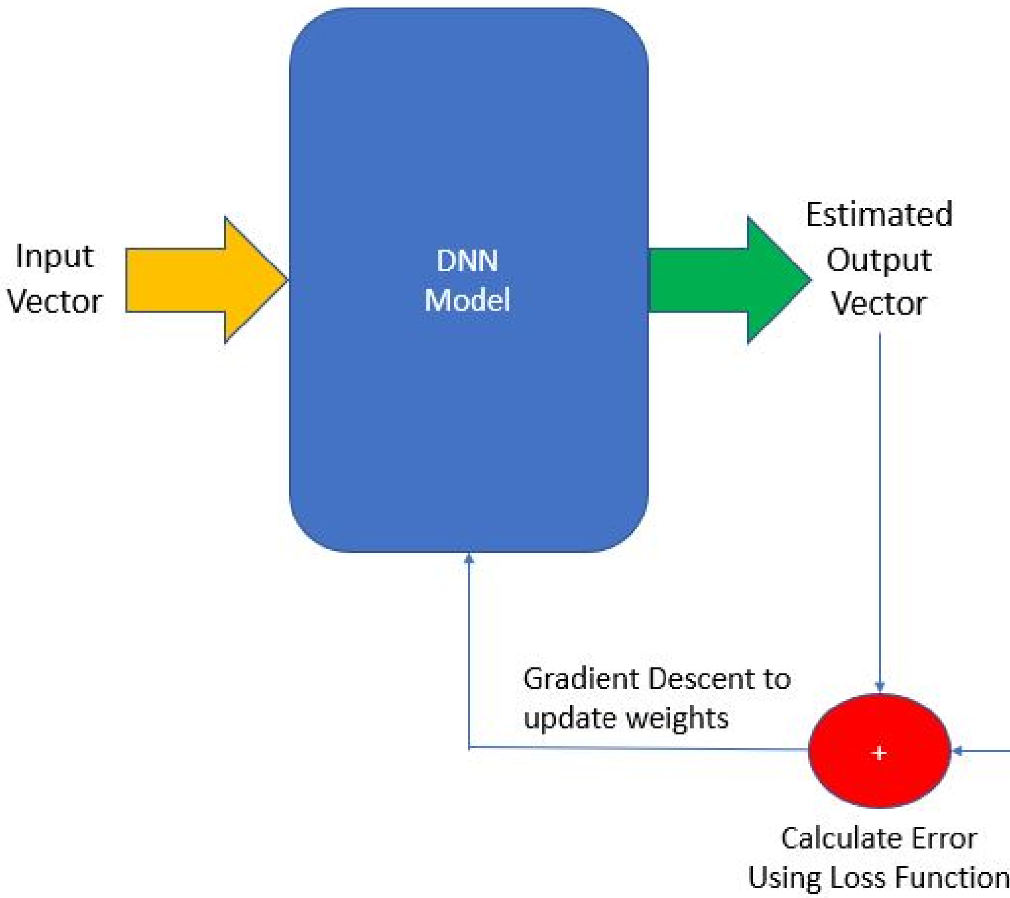


Figure 2.4: Flowchart of Back Propagation Step in DNN

$$w_{i,j(new)} = w_{i,j(old)} + \rho \frac{\delta E}{\delta w_{i,j(old)}} \tag{2.12}$$

2.12 shows the equation of new weights dependent on old weight values, loss function and learning rate. Here, E is the loss function used in the DNN model and ρ is the learning rate of the model. $w_{i,j}$ are the weights of links in the structure.

Loss Functions

Loss functions are used in optimization strategies in order to update weights and improve the model. The model decides the direction of weight update depending on the loss generated by the loss function. Additionally, cost functions also help in quantifying the ability of the model to estimate correct output, and the final goal of DNN training is to reduce this loss on the test dataset. Commonly used loss functions in DNNs are Crossentropy, Binary Crossentropy, Mean Squared Error, Mean Absolute Error, Root Mean Squared Error.

Chapter 3

Literature Survey on State-of-the-Art QPE Techniques

The methods used for QPE have evolved a lot in the past decades. QPE based on dual-pol measurements have been studied extensively in the past. Many relations like the R(Z), R(Z, Z_{dr}), R(K_{dp}), R(Z, Z_{dr}, K_{dp}) were developed. There are many variations of such equations with different parameters and biases derived based on data with different DSD models. Radars and satellites have been used for decades for estimation of rain-rate due to the existence of these empirical relations. In recent years, sophisticated QPE algorithms based on an ensemble of sensors started being introduced [22–26]. These techniques use multiple overlapping sensors like radars, gauges, disdrometers, satellites in conjunction with each other to estimate rain rates. These methodologies restricted in the 1990s and early 2000s due to limitations in data transmission bandwidths and computational capabilities [27]. These multiple sensor-based QPE models require data to be transferred to central regional or national processing centers. Zhang, 2016 [26] implements a Multi-Radar Multi-Sensor (MRMS) system invented at the National Centers for Environmental Prediction (NCEP). This technique uses 3D mosaics from weather radars as base data and integrates it with atmospheric, lightning, rain gauge, satellite, and environmental data and uses it in ensemble NWP models and algorithms for QPE. MRMS creates high-resolution Spatio-temporal mosaics seamlessly in real-time. Some advantages of this approach over empirical relation-based QPE are better accuracy, reduced measurement and calibration errors due to single radar use, and better modeling of the QPE based on atmosphere. The algorithms and NWP models used by MRMS are compute expensive and need to be carried out on servers with high computational capability.

In the last decade, Machine Learning has started to become a popular area of research in QPE. The main reasons for this trend are increased data and computing capability of computers due to advancements in Graphical Processing Units (GPUs). ML-based solutions have gained massive popularity in almost every field and lots of them have even proved to be more effective than long researched solutions. Researchers in the field of weather and climate study are also increasingly being involved in adopting them to solve problems in data analysis, modeling, and post-processing of NWP models [28].

One of the earliest ML-based approaches for precipitation flagging was introduced in [29]. This approach used vertical profiles of WSR-88D radar with 3 km by 3 km horizontal grid and 4 vertical grids from 1.5 km to 4.5 km as input to the neural network. Kohonen Neural Network was used for classifying events into rain/no-rain conditions observed on the ground and a 72% correct classification rate was observed. Later in 2001, [30] used radial basis function (RBF), a kernel-based function along with the same data structure, and found that this improved their correct classification rate to 82%. They discovered that using this classification scheme before applying the best possible Z-R equation for QPE resulted in almost 18% improvement in rainfall estimation. This adaptive RBFNN used 9-point input vectors which increase the complexity of the neural network.Li, 2002 [31] builds on the previous work to create a Rainfall Rate estimation Network which is trained along with the Classification Network.Amin, 2009 [32] merges the classification network into the rain estimation network for obtaining lower error on the testing set due to overestimation of rain rate. RBFNN was later also used by Feng, 2016 [33] to estimate rainfall accumulation using Cuckoo Search for optimizing centers, links, and length of the Neural Network. All of these approaches experience a reduction in rain-rate estimation accuracy due to misclassification of rain events and overestimation of rain rate.

Different techniques in machine learning such as Decision Trees, Random Forests, Artificial Neural Networks can find relations between dependent and independent variables based on the provided data without any preliminary knowledge of scientific relation between them [34]. This is the biggest reason behind the popularity of Machine Learning methods in current research. Kusiak, 2013 [35] used multiple data-mining-based approaches which also consisted of random forest, decision tree, support vector machines, and K-nearest neighbors for QPE using WSR-88D

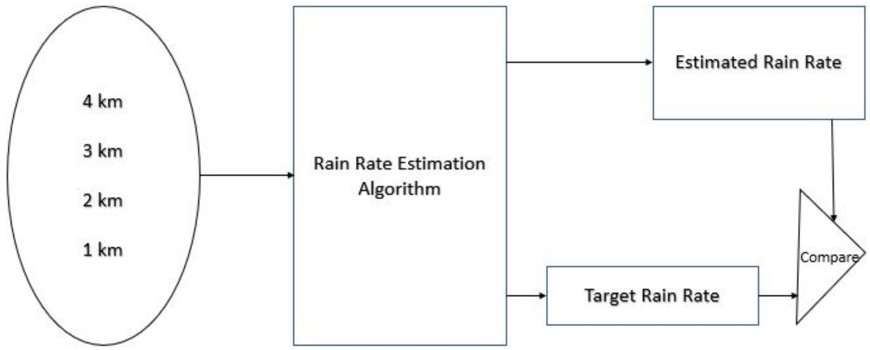


Figure 3.1: Overview of process of Estimating RR from Radar Data [31] [32].

and rain gauge data. Yang, 2017 [3] uses Terrain based weighted random forest (TWRF) method to estimate rainfall rate from VPR and provides a better solution for tackling the bias based on different terrain heights.

Artificial Neural Networks (ANN) / Deep Neural Networks (DNN) is a field of Machine Learning which is inspired by the neuro-synaptic structure of the brain. ANNs fit perfectly in atmospheric remote sensing applications due to their ability to draw out and model complex multivariate non-linear relationships. Yuan, 2020 [36] and Chiang, 2007 [37] use Recurrent Neural Networks (RNN) for QPE and QPF. It was found that RNNs produce better rainfall estimates as compared to R-Z relationships. [38] estimated rain rate using a Volume-to-point approach which maps the whole data volume to a point using Convolutional Neural Networks (CNN). Chen, 2019 [39] estimates rain rates using a combination of radar and disdrometer DSD data. The model was based on a Deep Neural Network (DNN). Chen, 2020 [40] used a fusion of Ground Radar and TRMM Satellite data in a Multi-Layer Perceptron-based Deep Neural Network for estimating precipitation. Tian, 2020 [41] introduces two neural networks based on precipitation estimation using weather radars. The two models used Back Propagation Neural Network (BPNN) and Convolutional Neural Network (CNN) respectively. According to their experiments and result, both models outperformed traditional equation-based methods by 75-85% lower Mean Squared Error Accuracy.

Not a lot of research has been done in the field of precipitation estimation using weather radars and Neural Networks combination. The existing research methods show great capability for the application of Machine Learning in this domain. Precipitation can be ambiguous depending on many regional and seasonal variations. All these variations cannot be efficiently captured using empirical equations. Machine learning has the potential to find hidden relationships between input features. This property of ML makes it an easier approach that can drive faster research findings and advancements.

Chapter 4

Research Methodology and Results

4.1 Data Description

The dataset used in this work comes from 2 sensors: weather radars and rain gauges. DNNs used in this thesis are supervised learning based models. Supervised DNNs require a set of inputs and labels for training. Weather radar moments like Z, Z_{dr} and K_{dp} are used as inputs whereas rain gauges are used as labels. As dual-polarized radar moments are required to measure PHIDP, a Dual-Pol radar's base product, NEXRAD (Next-Generation Radar) also known as WSR-88D network of radars was chosen. TRMMs ground validation rain gauges were chosen for label creation.

4.1.1 Next Generation Weather Radar (NEXRAD)

The technical name for NEXRAD is Weather Surveillance Radar, 1988, Doppler (WSR-88D) and after its year of deployment and Doppler configuration. The NEXRAD network of radars is operated by the National Weather Service (NWS), an agency of the National Oceanic and Atmospheric Administration (NOAA), and the Federal Aviation Administration (FAA). 160 High-Resolution NEXRAD radars are deployed all over the United States for complete coverage of the Northern American region. All the data gathered from 1991 till present has been recorded and documented using Google Cloud and Amazon Web Services. This network of radars was chosen as primary data in this work due to its easy availability, ease of access, and presence of Dual-Pol moments.

WSR-88D radars are Dual Polarization S-band radars that work at a frequency between 2700MHZ and 3000MHZ. Figure 4.1 shows the locations of the WSR-88D radars on the US Map. These radars have two primary units: Radar Data Acquisition unit (RDA) and Radar Product Generator (RPG). The RDA unit samples the atmosphere and produces the base and Dual-Pol moments. The base moments include reflectivity, velocity, and spectral width. Dual-Pol moments include dif-

Figure 4.1: NEXRAD sites on Map [42]

ferential reflectivity, differential phase, and correlation coefficient. The RDA is composed of the transmitter, antenna/pedestal, receiver, and signal processor. The RPG unit processes the data from the RDA to produce base products and other meteorological and hydrological products. The RPG is also responsible for the real-time distribution of base data and products. The radar works in two operating modes, clear air mode and precipitation mode. NEXRAD radars work in clear air mode when there is little movement of air in the weather, whereas precipitation mode is used during intensive weather activity. In clear air mode, the radar rotates slowly around the 360-degree axis and has five elevation angles (0.5 degrees to 4.5 degrees). This helps it to detect smaller airborne dust particles. Clear air mode is also used for measuring light snow. Clear air mode samples are recorded every 10 minutes. On the contrary, precipitation mode has faster antenna rotation along the 360-degree azimuth and has 14 elevation angles (0.5 degrees to 19.5 degrees) which allow for faster sampling. Samples are obtained every 6 minutes in this mode. Table 4.1 lists out the different characteristics and properties of WSR-88D radars.

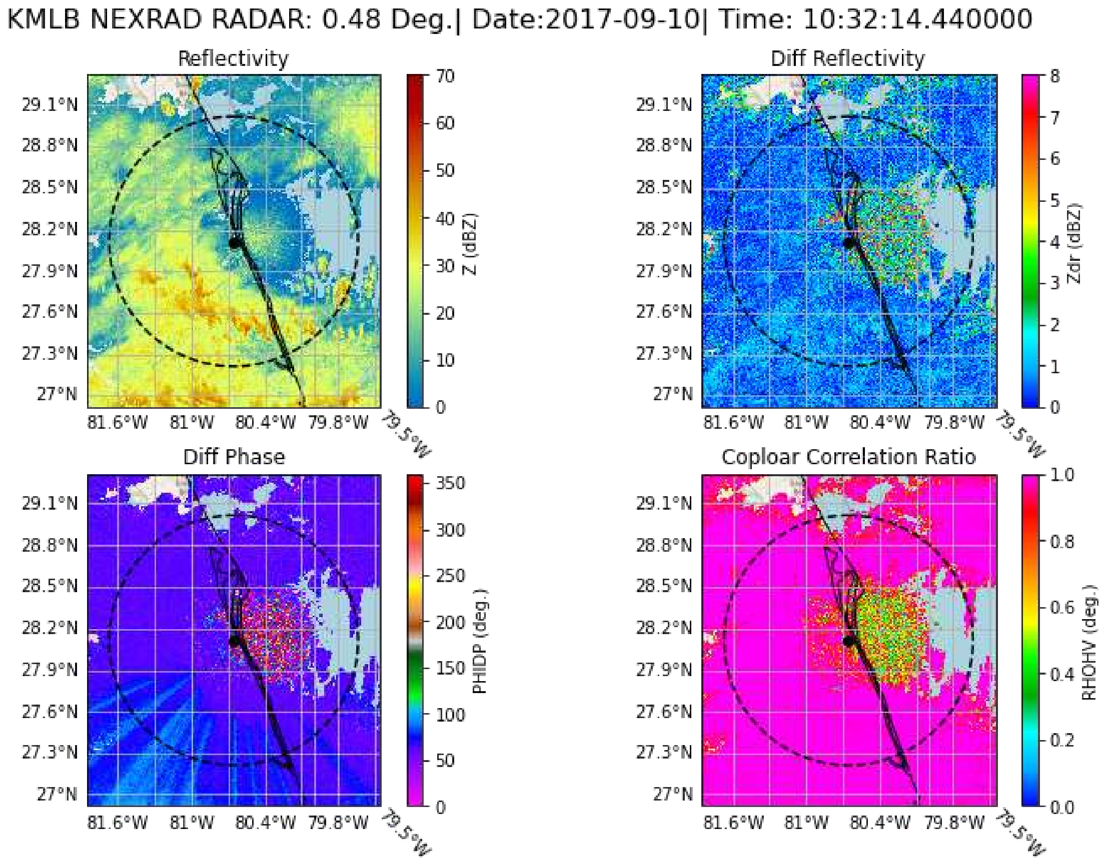


Figure 4.2: Example PPI Plot of base radar moments at 0.48 Degrees of KMLB NEXRAD site

Table 4.1: Characteristics of WSR-88D Wether Radar.

Parameter	**Description**
Radar Type	S-Band, coherent chain
Frequency	2.7 GHZ to 3.0 GHZ
Peak Power	700kW
Average Power	300 to 1300 Watts
Polarization	Dual Polarization (Horizontal and Vertical)
Base Moments	Reflectivity, Velocity, Spectral Width
Dual-Pol Moments	Differential Reflectivity, Correlation coefficient, Differential Phase

There are 3 levels of data obtained from WSR-88D radars. Level I data is the raw analog signal data recorded by the receiver in the RDA unit. Level II data consists of the base moments and dual-pol moments generated by the signal processor in the RDA unit. Level III data has hydrological and meteorological end products that are used for weather analysis, forecasting, issuing warnings, and weather tracking. Figure 4.2 shows PPI plots of different radar moments.

NEXRAD Level III Digital Precipitation Rate

Digital Precipitation Rate is a Level III NEXRAD Product. It estimates the instantaneous rain-rate using the Dual-Pol moments of the radar. DPR is expressed in inches/hr unit. As it is computed based on Level II data, DPR measurements are available at same timestamps as Level II data. This work compares the performance of rain-rates obtained from DNN models created in this work to the NEXRAD DPR product.

4.1.2 TRMM Ground Validation Rain Gauge Network

Tropical Rainfall Measuring Mission (TRMM) was a research satellite launched by the National Aeronautics and Space Administration (NASA) between 1997 to 2015. This mission was aimed at improving the understanding of variability and distribution of precipitation near tropical and sub-tropical regions of Earth. The TRMM satellite provided groundbreaking precipitation observations for 15 years before it was discontinued due to the depletion of its fuel. TRMM was later replaced by Global Precipitation Mission (GPM) an advanced successor.

During the TRMM and GPM missions, different ground sensors were also deployed in order to obtain validation data that could be compared with the TRMM and GPM observations for analysis and research. These ground sensors included ground radars, rain gauges, disdrometers, and some field campaigns. The rain gauge data from these missions are used as validators in this work for labeling the data. TRMM era gauges were tipping buckets with loggers and required manual downloading. These were later replaced by telemetered gauges in the GPM era for simplicity. The GV gauge data is obtained in the raw format and then converted into 3 basic formats that are distributed for use. The formats include GAG, GMIN, and 2A56 formats. In this work, we will

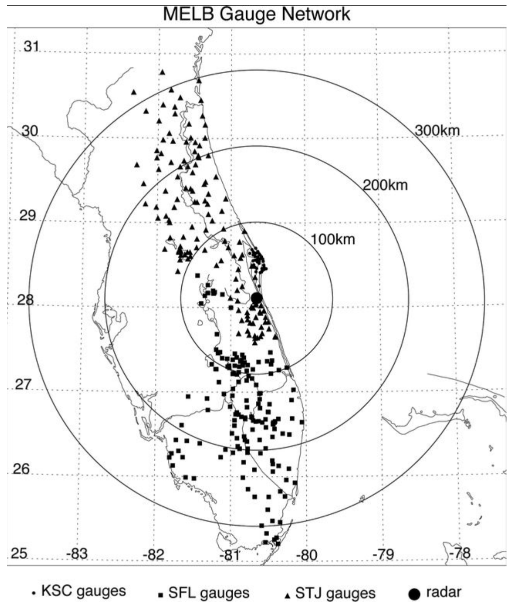


Figure 4.3: TRMM Ground Validation Gauge network around Melbourne, FL [43]

use the GMIN format of data as our raw unprocessed data. In machine learning, data needs to be converted, aligned, and organized in order to make it compatible for use. The raw data cannot be used directly due to the following reasons:

1. Essential data needs to be selected from the raw sensor data and needs to be brought into a definite structure to make it readable by the DNN.

2. Data from the sensors are usually stored in a different form and need to be converted using multiple data processing algorithms.

3. Raw sensor data may have missing values and DNN models do not handle missing data well. Good data imputation and approximation schemes are needed to complete the input data.

4. Sensor data could be faulty, it is important to remove faulty data from the dataset to improve the generalization capability of the DNN.

Finally, due to the high volume of data needed, processing it manually is almost impossible. This introduces the necessity of automating the data processing activities. There are 2 stages of data processing required in this work, pre-processing and post-processing. The below sections explain the respective processing stages in detail and how automation is used to make it seamless and fast.

For this work we will be using the data from NEXRAD KMLB radar which is based in Melbourne, Florida region (Coordinates: 28°6'48"N 80°39'15"W, Elevation: 35 feet). For validation and labels, TRMM/GPM Ground Validation Gauges based in the same region are used. This area has 3 rain gauge networks named KSC (Kennedy Space Center), SFL (South Florida WMD), STJ (St. John's River WMD). These networks have a total of 288 rain gauges. Figure 4.3 shows a plot of all the gauges near Melbourne, FL [43].

4.2 Pre-processing of datasets

Each NEXRAD VCP file is distributed in a compressed format. The compressed file can have a ZIP, TAR, GZIP or ar2v format depending on the year in which the data was collected and archived [42] [44]. Consecutive radar scans could be 6 minutes to 10 minutes apart depending on the mode of operation of the radar. Whereas rain gauge data is available every 1 minute. As radar data has a high spatial resolution compared to rain gauges, it is necessary to align both these sensors in space. Moreover, the temporal resolution of radar is different from that of gauges, therefore they need to be aligned in time as well. Thus, the main part of preprocessing step is aligning them in space and time to get a unified dataset.

This work uses the vertical profile of radar reflectivity, Differential Reflectivity and Specific Differential Phase to estimate rain rates using DNN. As K_{dp} is a derived radar moment, this work uses Dual-Polarization Radar Operational Processing System (DROPS) utility to estimate K_{dp} from dual-polarization Differential Phase [45]. NEXRAD data is available in polar coordinates, and it is essential to first convert them into cartesian coordinates for obtaining vertical profiles. The reliable range of NEXRAD data usage is 230 km. The raw data available from KMLB NEXRAD contains sweeps at different altitudes. A plot of radar's moments at an elevation is called a Plan Position Indicator Scan (PPI). After cartesian gridding, radar moments at constant altitudes are obtained by extracting and interpolating from the variable altitude sweeps. If a single sweep of this data is plotted it is called a Constant Altitude Plan Position Indicator (CAPPI) scan. Figure 4.4 shows a comparison of a PPI scan and CAPPI scan.

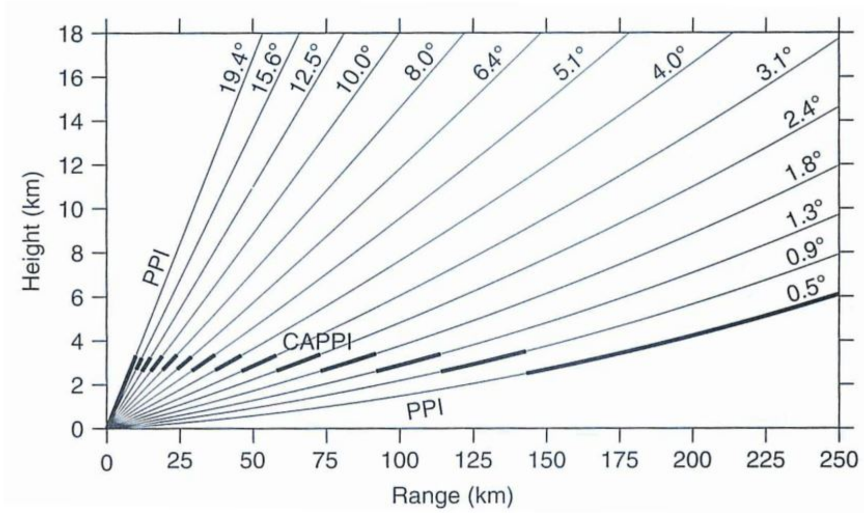


Figure 4.4: Comparison of Plan Position Indicator (PPI) sweep with Constant Altitude Plan Position Indicator sweep (CAPPI) [46]

This work creates 200*200 horizontal grids with 1km x 1km horizontal resolution and 8 vertical levels with 500m vertical resolution. Cartesian Gridding was performed using Py-ART [47] only on data within 100 km range of radar because the height of the lowest elevation goes to around 1.8km at 100km. If data at this 100km is gridded, the data moments interpolated at the lower levels of CAPPI would be inaccurate and can result in bad data in our dataset. Figure 4.6 shows the structure of a sample grid with 1km vertical resolution. The lowest level is 500m and the highest level is 4km. Figure 4.5 shows a side-by-side comparison of a PPI plot at sweep = 1 with a CAPPI plot at 1km height. These plots were made from NEXRAD KMLB data from 2016 when Hurricane Matthew hit the city of Melbourne, FL.

The TRMM/GPM rain gauges are dispersed along a big region around Melbourne, Florida. Since only the data in the 100km range of KMLB is considered, all gauges not in this range were removed. It was found that there are 110 gauges in this range. These gauges were used for alignment with KMLB's CAPPI data in space and time. Figure 4.7 Plots the physical locations of all the 110 TRMM/GPS rain gauges in KMLB's 100km range radius.

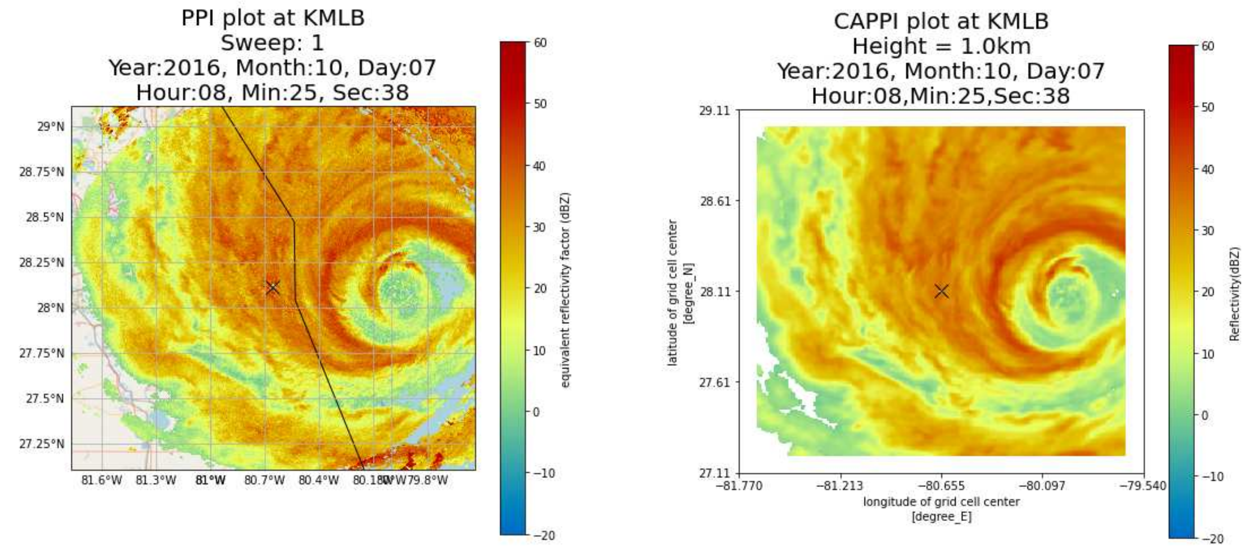


Figure 4.5: Side by Side Comparison of PPI (sweep = 1) and CAPPI (height = 1km)plots from NEXRAD KMLB site. Timestamp: Hurricane Matthew, 2016/10/07

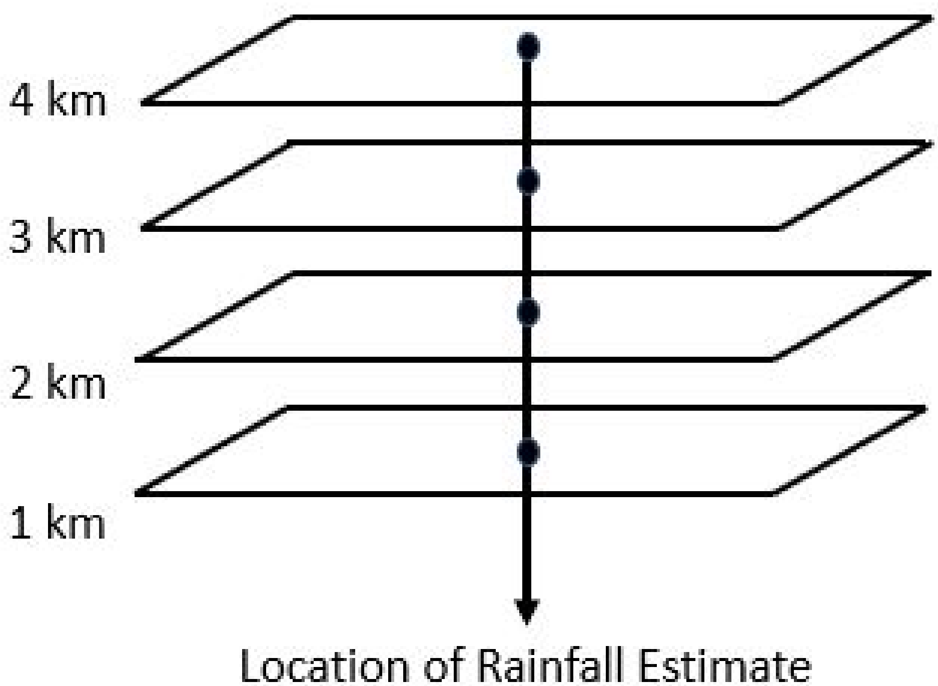


Figure 4.6: Spatial Alignment of Rain Gauges with Vertical Profile.

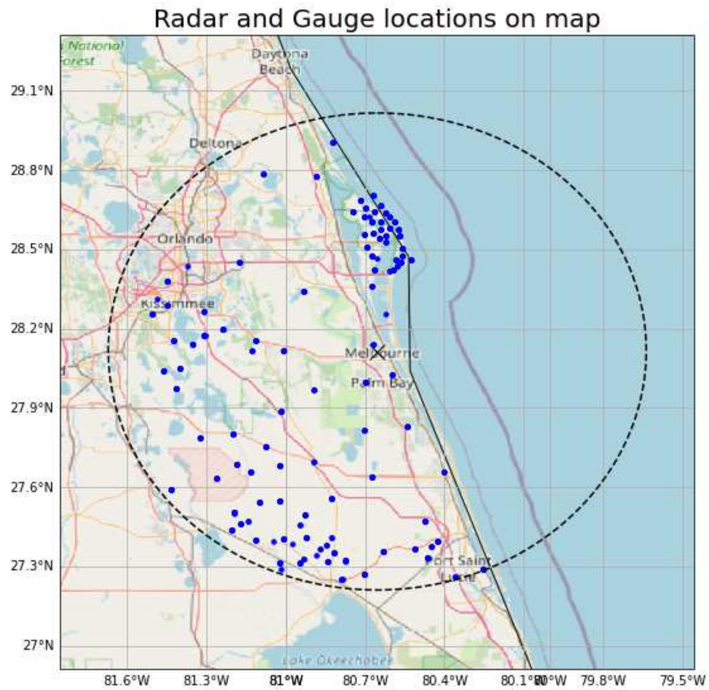


Figure 4.7: TRMM/GPM rain gauges in NEXRAD KMLB's 100km range radius

NEXRAD samples data every 6-10 minutes whereas gauges sample data every 1 minute. Therefore, hourly rain rates from gauge samples within +/- 2.5 minute range of every NEXRAD sample were accumulated and averaged together to get the hourly rain rates for that NEXRAD scan. Figure 4.8 shows an example of temporal aligning performed on the gauge and radar datasets to obtain a unified dataset. Additionally, NEXRAD Level 3 products are also available for every NEXRAD VCP scan. Digital Precipitation Rate, a level 3 product is also used for verification of the model created in this work. Rain Rate from DPR product also needs to be aligned in space and time with the rain gauge. This is also done using a procedure to aligning radar moments. As DPR is in inches/mm, it also needs to be converted into mm/h, which is the standard rain rate unit used throughout this thesis. Figure 4.9 shows a plot of aligned dataset with Z on Y-axis (Left) RR on Y-axis (Right) vs Timestamp on X-axis.

The aligned samples are recorded for every gauge and radar combination while discarding the NEXRAD's ar2v format data. NEXRAD files are 100MB per scan and saving them is resource expensive. Discarding these files is economical in terms of memory occupancy. The final dataset

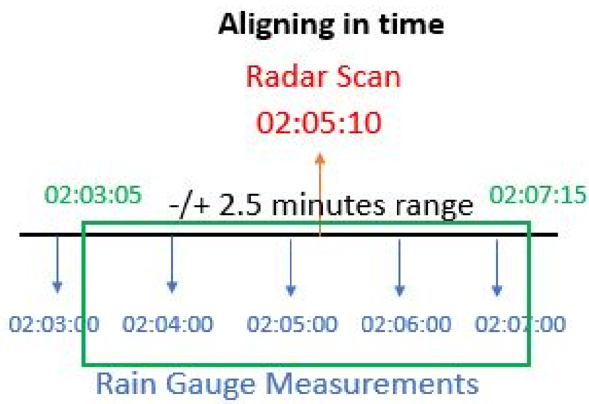


Figure 4.8: Timestamp based temporal aligning of gauges and radar

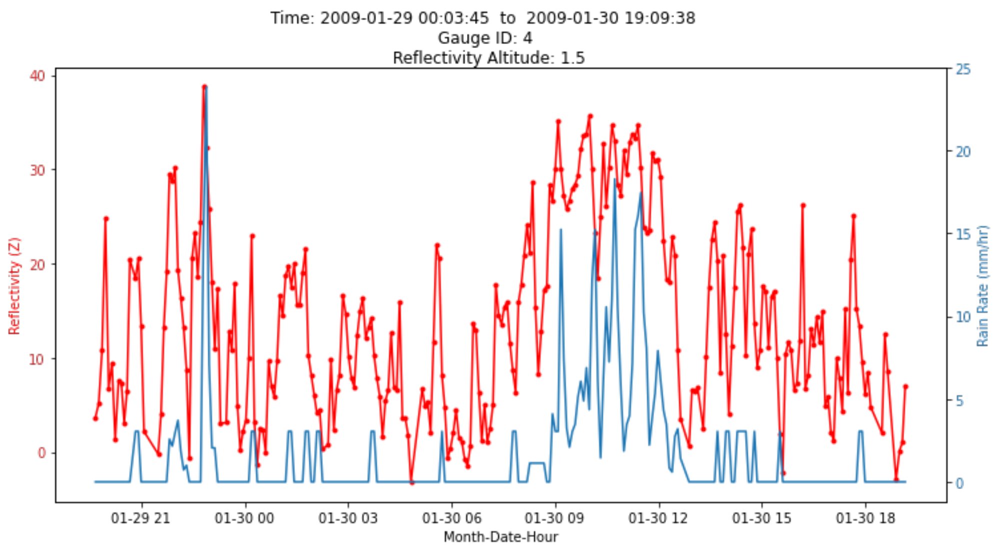


Figure 4.9: Plot of aligned dataset. Z and RR vs Timestamp

is saved in CSV format. The combined size of the entire aligned training, testing, and validation datasets used in this work is around 80 Mega Bytes.

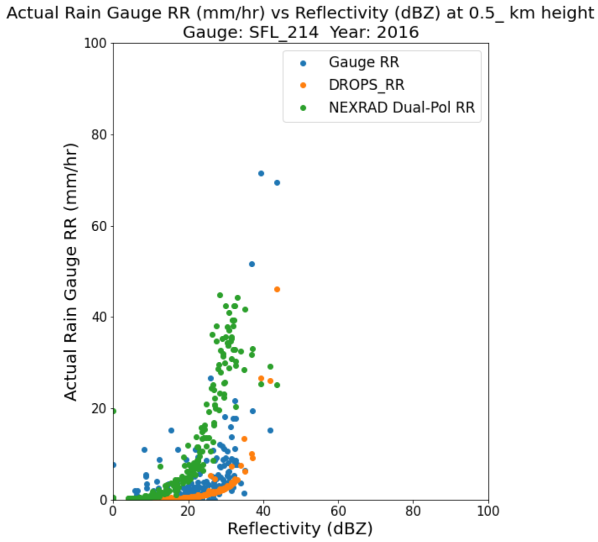
4.3 Post-processing of Datasets

Rain gauges are point instruments and they are susceptible to faults and errors. It is important to clean out the faulty rain gauges in order to keep the dataset fault free. If the dataset is fed to the DNN model without sorting out, the model would try to detect hidden patterns in the faulty data as well. This does not help in good generalization of the model and can lead to overfitting. Empirical Z-R relations can be helpful in observing if the data is faulty. The curve of Reflectivity at a height (from the ground) versus Rain Rate is an exponential curve similar to the Marshal-Palmer curve. Gauges that severely deviate from such a pattern are faulty and were removed from the dataset. This simple mechanism is very helpful in removing major errors that would certainly disturb the training of a good DNN model. Figure 4.10 shows Z-R plots with reflectivities at different heights vs rain rate from rain gauges.

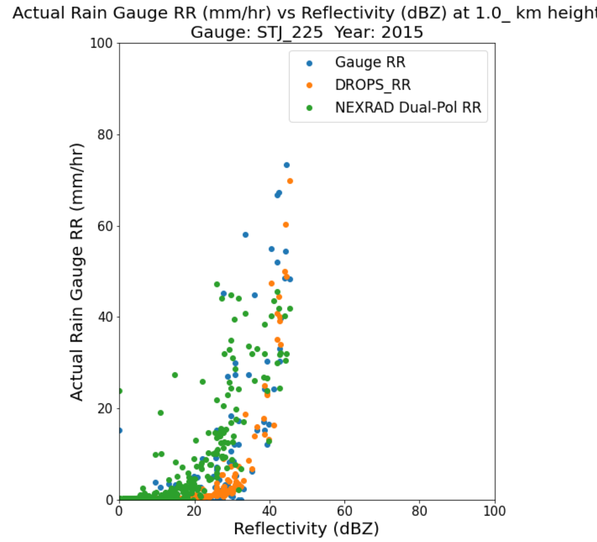
Table 4.2: Details of classes present in data.

Data Year	Data Month	Number of total samples	Rain Cases	No-Rain Cases
1995	August,September	79,566	7,106	72,460
2015	August	89,860	4,027	85,833
2016	October	14,990	4,429	10,561
2017	August, September	28,529	2,073	26,456
2018	July	2,322	1,109	1,213

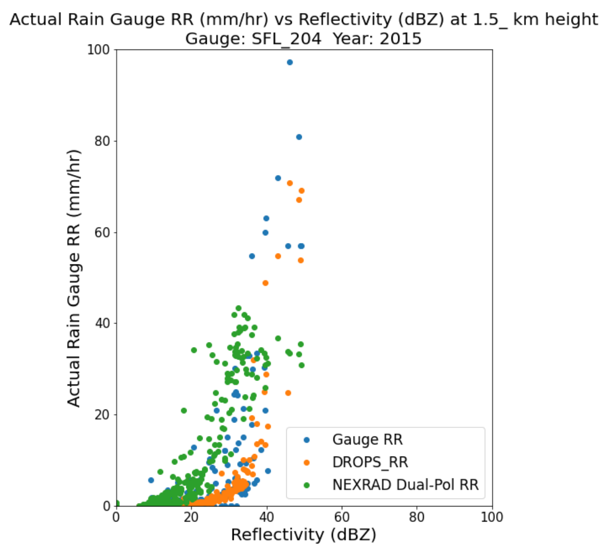
This work starts with experiments on the fundamental task of rain flagging (rain/no-rain classification) and progresses by implementing incremental solutions to problems encountered. Until finally this knowledge is used to create a DNN model for QPE. Table 4.2 expands on the Year and Month of data used along with total samples, the number of rain cases, and no-rain cases present for the respective data.



(a) R-Z plot of SFL_214 gauge at 0.5 km height in October 2016.



(b) R-Z plot of STJ_225 gauge at 1.0 km height in August 2015.



(c) R-Z plot of SFL_204 gauge at 1.5 km height in August 2015.

Figure 4.10: Z-R plots with reflectivities at different heights vs rain rate from rain gauges

Table 4.3: Splitting of data into datasets

Dataset	**Data Year and Months**
Training	Aug 2015
Validation	October 2016
Testing	Aug/Sept 2017, July 2018
Testing for RBFNN	Aug/Sept 1995

4.4 Feature Scaling in Dataset

The range of features inside a dataset can be different due to the vast variation in values or due to different min-max values of the features. Some optimization algorithms like Gradient Descent, where step-based changes are made, do not handle range irregularity well. The difference in the range between the range of features causes different step sizes for gradient descent between the features. Feature scaling is used to ensure that the gradient descent smoothly moves into the local minima and the gradient descent steps are updated uniformly along with all the features. Feature scaling can be done by standardization, normalization, or regularization of the inputs.

Normalization shifts and rescales values so that they are ranged between 0 and 1. Whereas standardization centers the values around the mean with a unit standard deviation. Doing this converts the mean of the feature to zero and the distribution of the feature is a unit standard deviation. Standardization is used as the only feature scaling technique applied in this work. Equation 4.1 specifies the standardization formula. whereas equation 4.2 shows the normalization formula.

$$X' = (X - X_{MEAN}/X_{STDDEV}) \tag{4.1}$$

$$X_{NORM} = (X - X_{MIN}/X_{MAX} - X_{MIN}) \tag{4.2}$$

4.5 Rain/No-rain Classification using DNN

Models for classification of events into rain/no-rain cases are very useful in analyzing the Earth's water cycle. Such types of models are popularly also called precipitation flagging / rain flagging algorithms. This work implements such a classification model with the help of vertical profile of radar moments. This model aims at classifying rain or no-rain condition observed on the ground by using the vertical profile of radar as a feature. This type of DNN model is called a logistic regression model. Logistic regression is the process of classifying input cases into discrete binary classes. Vertical profile of radar moments help in providing a great snap of the atmospheric and hydrometeor conditions just above the radar.

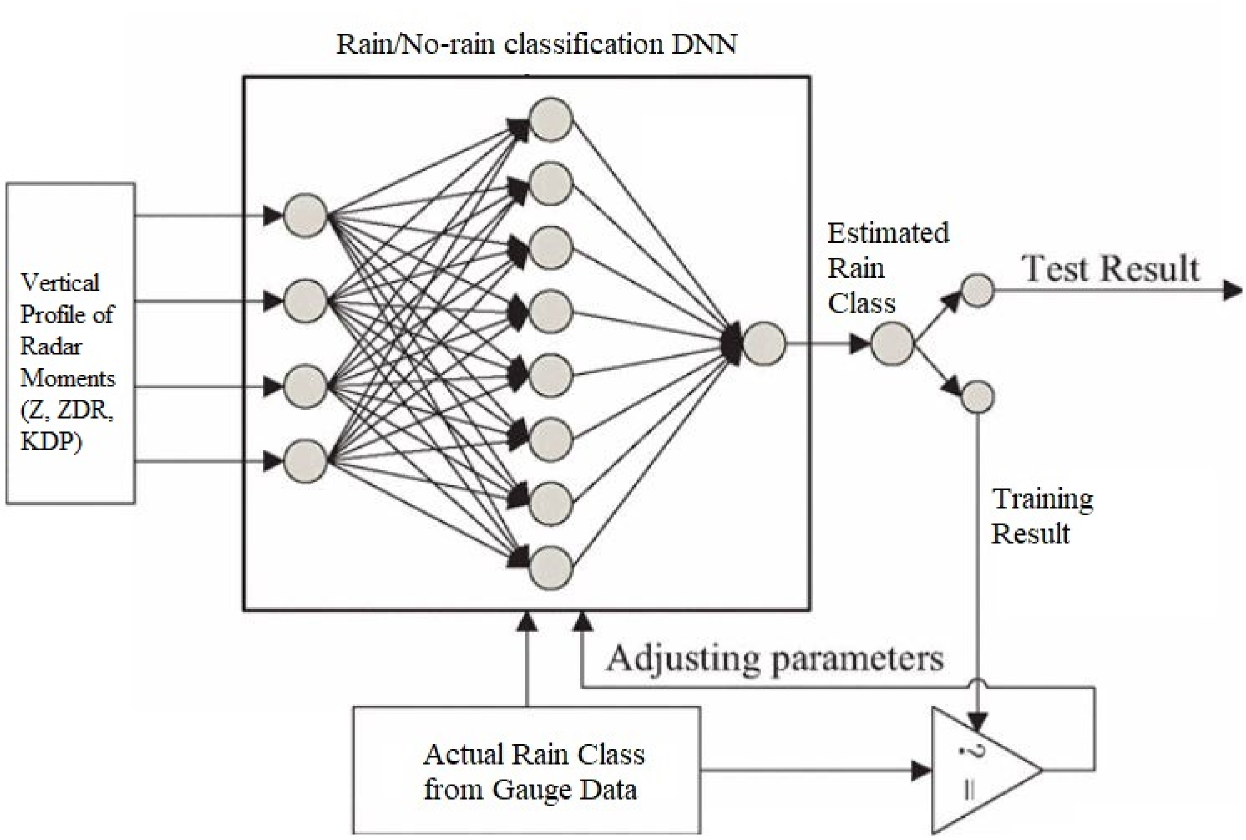


Figure 4.11: Workflow of Rain/No-rain Classification using DNN

Figure 4.11 shows the data flow and high-level architecture of the classification DNN models used for rain / no-rain classification task. The vertical profile of different combination of radar moments is given to the DNN model. The DNN model tries to classify the input events into rain / no-rain cases. Actual observations from rain gauges are used to calculate the error and re-train the model to increase it's performance.

4.5.1 Activation Functions

The output obtained from the DNN in logistic regression problems should indicate the probability of a Positive (TRUE / HIGH / 1) class. In this case, the output of the logits should be a value between 0 to 1 which indicates the probability of the input case being a Rain Class. A sigmoid function models an output between 0 to 1, shown in Figure 4.12. 4.3 shows the equation of the sigmoid function.

$$f(z) = \frac{1}{1 + exp(-z)} \tag{4.3}$$

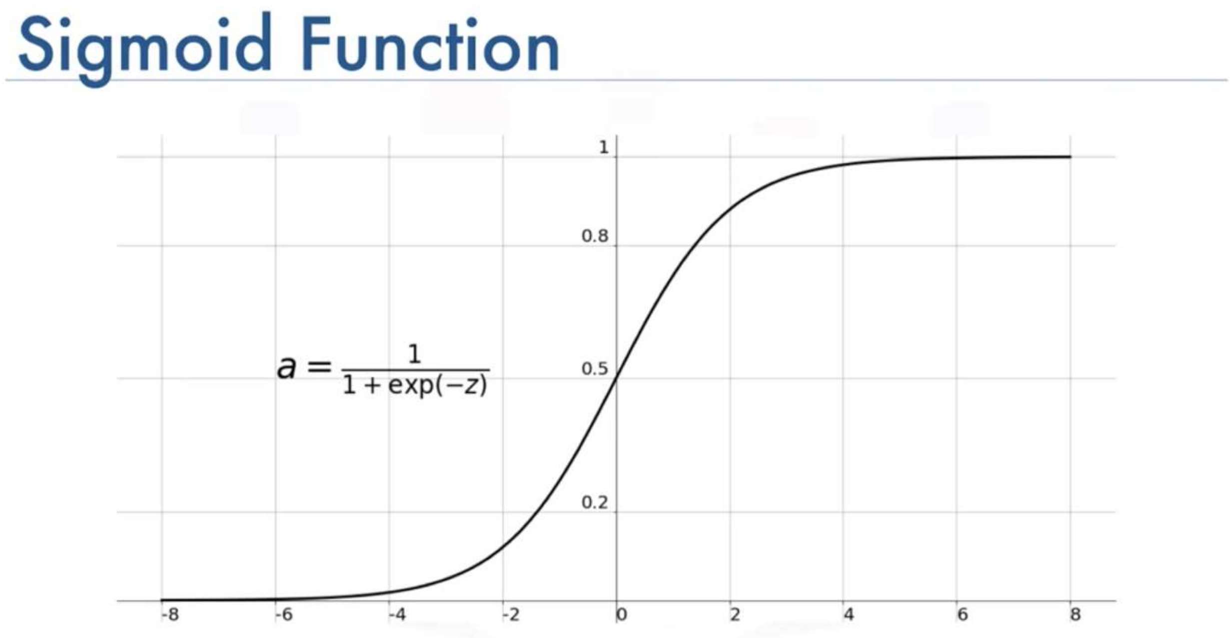


Figure 4.12: Sigmoid Curve along with Sigmoid equation

4.5.2 Loss Function

Binary Crossentropy (also called log loss) is used for calculating loss during the forward propagation phase of the network. During the back-propagation phase, gradients are calculated based on the cost value given by the loss function and the weights and biases are adjusted accordingly. This process is repeated (batch size x epochs) number of times to obtain a final trained model. The choice of the loss function is one of the most important parameters in Machine Learning. Binary Crossentropy is used to calculate the loss in binary classification (logistic regression) problems. This function compares the predicted probability (output of the sigmoid output layer) with the correct class value to calculate the difference. This difference is used to penalize the DNN, calculate gradients and change weights during back-propagation.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i.\log(p(y_i)) + (1-y_i).\log(1-p(y_i)) \tag{4.4}$$

Here, y is label. P(y) is the sigmoid predicted probability of the label. N is the total number of samples.

4.5.3 Evaluation Metrics

The choice of evaluation metrics is very specific to the type of problem in consideration. Classification accuracy and correct classification percentage are the two metrics that are used in almost every binary classification problem. But in meteorology, not only is it important to analyze the correct classification of events, but also the wrong classification of events. Every rain class wrongly classified as no-rain can cause devastating harm to property and life. Whereas every no-rain condition classified as a rain condition can result in wrong hydrological analysis. Therefore, it is very important to also consider true positives, true negatives, false positives, and false negatives while analyzing the performance of the model. As it is critical to reducing the number of FN as well as FP in this classification problem F1 Score is used as the final analytical metric. TP, TN, FP, FN are the basic metrics required to calculate Precision, Recall, and F1 Score. They can be easily understood using a Confusion Matrix.

True Positives

Every rain case successfully classified as rain condition is a true positive. True Positive in this context is the total number of rain cases correctly classified by the model.

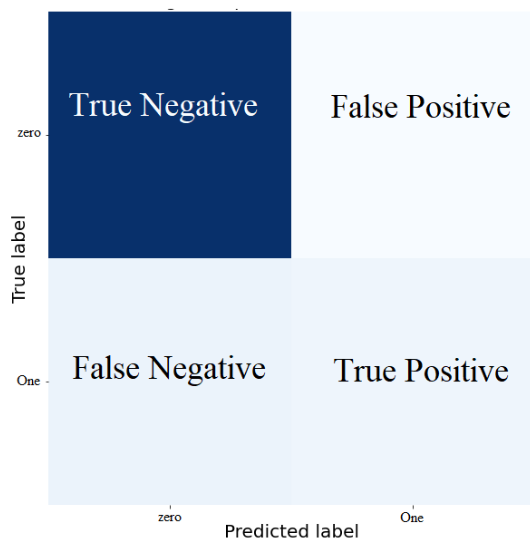


Figure 4.13: Sample Confusion Matrix

True Negatives

Every no-rain case successfully classified as no-rain condition is a true negative. True Negatives in this context is the total number of no-rain cases correctly classified by the model.

False Positives

Every no-rain case classified as a rain case is a false positive. False Positive in this context is the total number of no-rain cases incorrectly classified by the model.

False Negatives

Every rain case classified as a no-rain condition is a false negative. False Negative in this context is the total number of rain cases incorrectly classified by the model.

Precision

Precision measures the proportion of the positive classifications that were classified correctly. In rain / no-rain classification problem, it is the percentage of correct rain cases from all cases that were predicted as rain case. It is calculated as the ratio of TP to sum of TP and FP. Maximizing precision will minimize the false-positive errors.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.5}$$

Recall

Recall measures the proportion of actual positive conditions identified correctly in the dataset. In rain / no-rain classification problem, it is the percentage of correctly classified rain cases out of all rain cases. It is calculated as the ratio of TP to the sum of TP and FN. Maximizing recall will minimize the false-negative errors.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.6}$$

F$_\beta$ Score

In binary classification applications where the dataset is imbalanced towards a class (either 0 or 1), it is important to maintain a balance between precision and recall. High value of precision and low value of recall means that the model is classifying almost all the cases as No-Rain event. Whereas high recall and low precision means that the model is classifying more samples as Rain cases. In rain /no-rain classification problem most cases belong to the No-Rain class, therefore, the model should be capable of correctly classifying Rain cases with better accuracy. Thus, the aim of the models is to have a higher recall and moderate precision. To achieve such balance between precision and recall, F$_\beta$ score is usually used. F$_\beta$ score is the weighted combination of precision and recall [48], [49], [50], [51], [52], [53]. The equation of F$_\beta$ is shown in 4.7.

$$F_\beta Score = \frac{(1 + \beta^2) * Precision * Recall}{(\beta^2 * Precision) + Recall} \tag{4.7}$$

There are three common variations of F$_\beta$ Score based on the value of β. The appropriate variation of this metric is used depending on the nature of the problem. Adjusting the value of β adjusts the amount of weight given to either precision or recall during calculation of the F Score.

1. $F_{0.5}$ Score gives more weight on precision and less weight on recall. This variation is used when precision is more important than recall.

2. F_1 Score gives equal weights on precision and recall. This variation is used when precision and recall are equally important.

3. F_2 Score gives more weight on recall and less weight on precision. This variation is used when recall is more important than precision.

In this problem recall is slightly more important than precision. Using F2 score is not optimal in this application as it gives extreme weight on recall. Therefore F1 score is used and evaluation of models is done manually by observing precision and recall values. 4.8 gives the equation of F1 Score.

$$F_1 Score = 2.\frac{precision.recall}{precision + recall} \tag{4.8}$$

4.5.4 DNN based RC(Z) Rain/No-Rain Classification Model

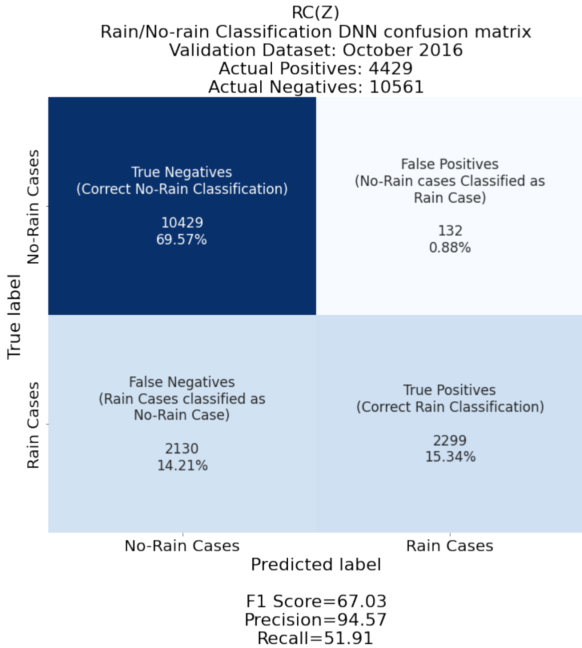
Primarily, Vertical profile of only reflectivity (VPR) is used to classify events into rain / no-rain condition on ground. VPR is a good feature for use in this situation due to a well-defined Z-R relation between Z and Rain Rate. Many historical algorithms used only radar reflectivty for rain flagging. This was mainly due to the unavailability of Dual Polarization radar moments at that time. It would be interesting to see how a DNN based approach using the same input data structure can cope against these historical models. The RC(Z) model converts the VPR from ground radar to rain or no-rain class using DNN. The input to this model consists of 8 values of reflectivity taken at 0.5 km distanced regular heights from 0.5 km to 4 km above the rain gauge and the output is the rain class specifying if rain or no-rain condition was observed on the ground. [30] also uses a similar data structure to approach this problem. It uses RBF-NN as the classification model and tests the performance on Aug/Sept 1995 data. In order to compare the performance of RC(Z) DNN model, it was also tested on the Aug/Sep 1995 dataset.

Training Parameters

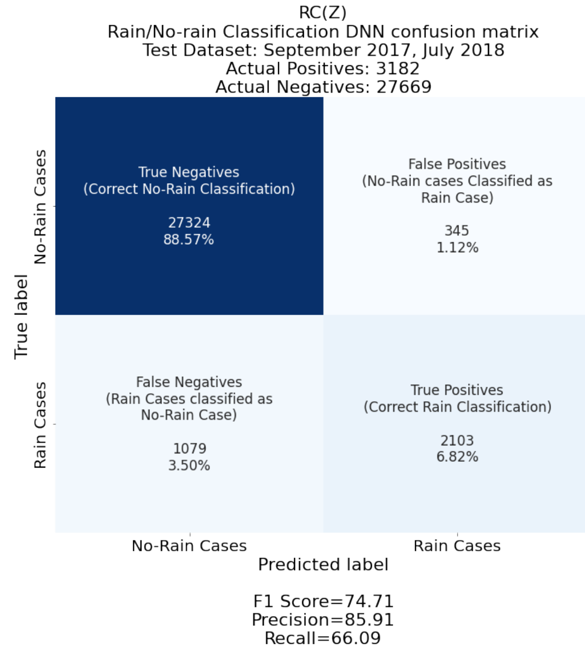
During data pre-processing, VPR data was transformed to form sets of 8 normalized reflectivity values per input. Rain/No-Rain classification from rain gauges are used as labels. Experimentations with the type of optimizer and optimizer hyperparameters were done during the training phase. Stochastic Gradient Descent, Adam and RMS_Prop were tried out along with their different hyperparameters. Stochastic Gradient Descent (SGD) produced the best perfomance while training. Thus, SGD was finalized to be used across all the experiments done for Rain / No-Rain classification poblem. The learning rate was set to 0.05. The training dataset is trained with a batch size of 512 for 5 epochs. The validation dataset is used to analyze the generalization capability of the model. To avoid overfitting of the model to the training dataset, the training process was stopped at 5 epochs. Heatmap-based confusion matrices are used for ease of analysis of results in this work. A confusion matrix is a commonly used visualization standard used to summarize classification problems. It summarizes the TP, TN, FP, and FN of the classifications. Figure 4.14 shows how the model performs on the validation, testing and August / September 1995 datasets.

Model Description

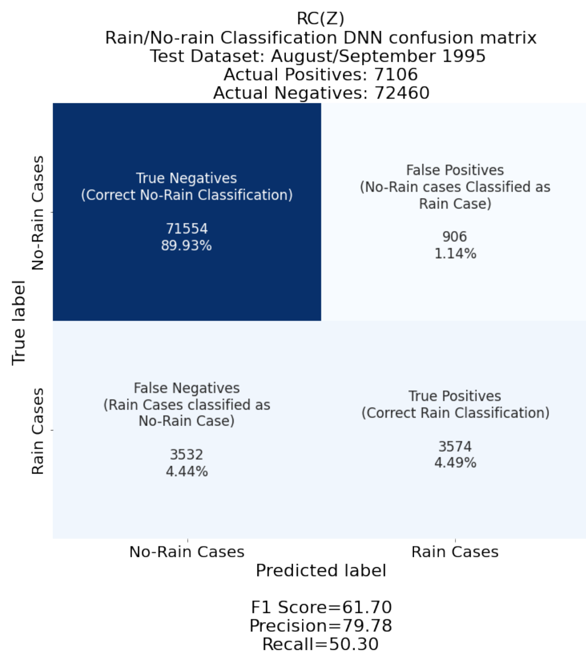
The model consists of 1 input layer, 2 hidden layers, and 1 output layer. All layers in the model are Dense layers with 32 bit floating point numbers to represent weights. ReLU is used as the activation function for the hidden layers. As this is a binary classification problem, sigmoid was chosen as the activation function for the output layer. Class '1' was used to denote rain conditions and class 0 denotes no-rain conditions. The output of the sigmoid layer is binarized using 0.5 as a threshold to obtain rain/no-rain classifications. Figure A.1 shows a plot of the model architecture with details of the number of neurons used for each layer.



(a) Performance of RC(Z) rain/no-rain Classification DNN on Validation Dataset (October 2016).



(b) Performance of RC(Z) rain/no-rain Classification DNN on Test Dataset (September 2017, July 2018)



(c) Performance of RC(Z) rain/no-rain Classification DNN on August/September 1995 dataset

Figure 4.14: Performance of RC(Z) rain/no-rain Classification DNN

Imbalanced Dataset Problem

From Figure 4.14, it can be seen that the model is biased towards No-Rain classification and is more prone to classify events as no-rain event. Also,the precision of the model is high whereas the recall is quite low in comparison. This behaviour can be observed due to the relatively high value of False Negatives. This is not good as the correct classification of rain cases is critical in this application. Thus, it is essential to improve recall of the model. The reason for this problem can be rooted back to the training dataset. The training dataset itself is skewed towards no-rain cases, due to the reason that rainfall is a relatively rare phenomenon as compared to non-rainy events.

The model needs to be carefully adapted to prevent it from skewing towards no-rain classification. This problem is called dataset imbalance. Dataset imbalance can be solved by undersampling or oversampling. Undersampling is a technique wherein the values belonging to the majority class are removed to make the number of samples of both classes equal. Whereas Oversampling is a technique wherein samples belonging to minority classes are duplicated to make the number of samples of both classes equal. As the dataset we are using is largely skewed and every information is useful, removing minority class samples might reduce the generalization capability of the model. Moreover, duplicating samples may cause overfitting of specific rain cases and the model might not generalize the rain cases effectively. To tackle such problems, the technique of training with class weights is used to achieve balanced training. As data based approaches like undersampling and oversampling aren't appropriate, this thesis solves the problem by adding weights to classes in the loss function during training. This approach is called as class weights based training.

Logistic regression uses the Logarithmic Loss function. The log loss function is a convex function and has only one minima.

$$logloss = \frac{1}{N}\sum_{i=1}^{N}[-(y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i))] \tag{4.9}$$

Here, N is the total number of samples. y_i is the actual value of the target class and $\hat{y}_i$ is the prediction probability of the target class.

Adding constant multipliers in this equation will increase the penalty of the loss function. These constant multipliers are called weights. Adding weights with uneven penalties will result in uneven penalization of the classes. Thus the modified log loss equation becomes:

$$logloss = \frac{1}{N}\sum_{i=1}^{N}[-(w_1(y_i * \log(\hat{y}_i) + w_0(1-y_i) * \log(1-\hat{y}_i)))] \quad (4.10)$$

Here, w_0 is the class weight of Class 0 (No-Rain cases). w_1 is the class weight of Class 1 (Rain Cases). The values of these weights are calculated as shown below:

$$w_j = n\_samples/(n\_classes * n\_samples_j) \quad (4.11)$$

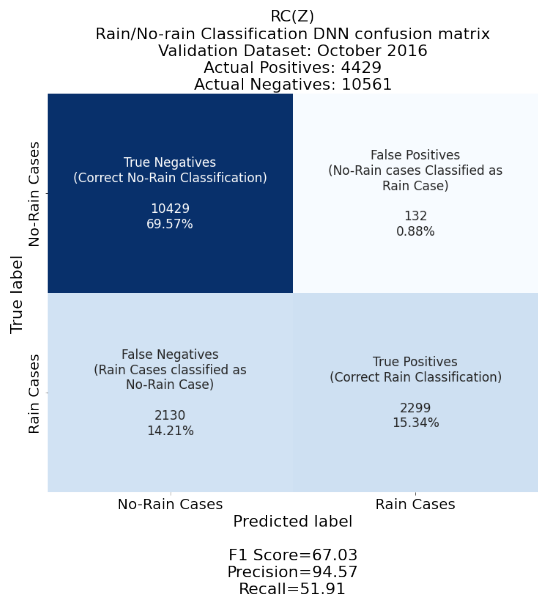
Here, j is the class in consideration, w_j is the weight value for the class, n_samples is the total number of samples in the dataset, n_classes is the number of unique classes in the dataset and $n\_samples_j$ is the number of samples of the respective class. The weight values for the training dataset were obtained as follows:

Table 4.4: Classes and their weights for training

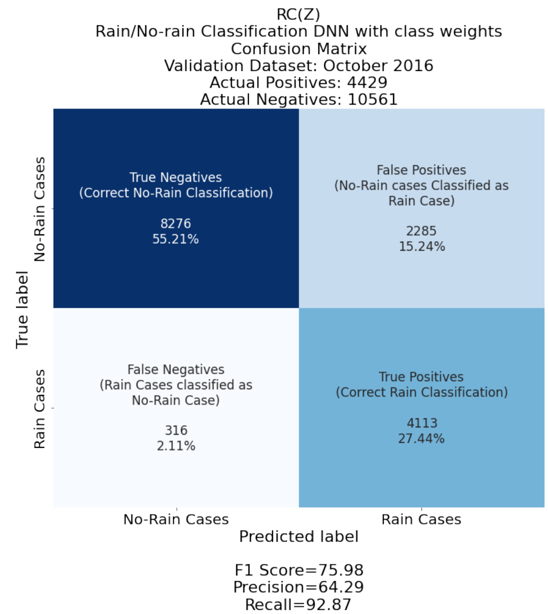
Classes	Number of Samples	Weight Values
Class 0	85,833	0.52
Class 1	4,027	11.16

Total samples in training dataset: 89,860

Finally, a model with a similar structure, hyperparameters, and training parameters is trained using the weighted log loss and class weights. An evident improvement in the results was observed. Figure 4.15 and Figure 4.16 compare the performance of the plain RC(Z) Rain / No-rain DNN classification model with a class weights training based RC(Z) Rain / No-rain DNN classification model.

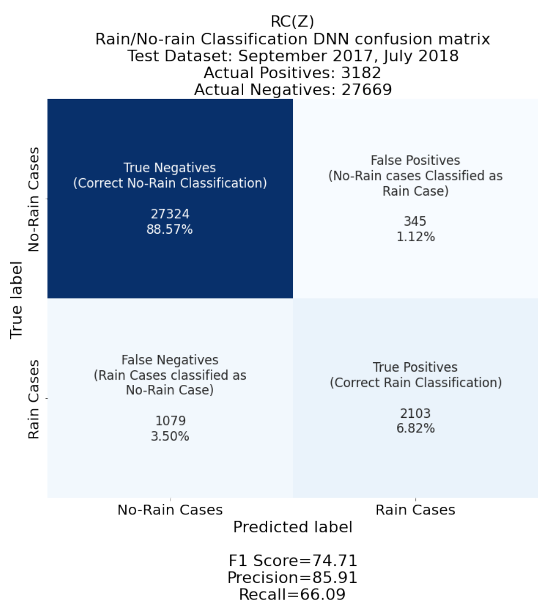


(a) Performance of RC(Z) rain/no-rain Classification DNN without class weights on Validation Dataset (October 2016).

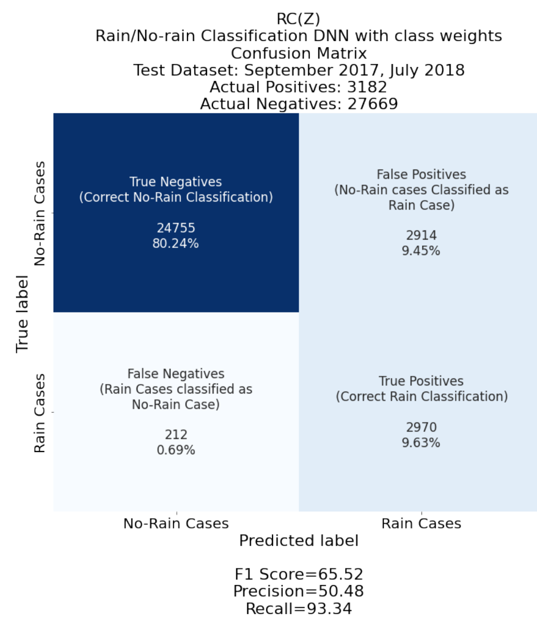


(b) Performance of RC(Z) rain/no-rain Classification DNN using class weights on Validation Dataset (October 2016).

Figure 4.15: Comparison of Performance of RC(Z) rain/no-rain Classification DNN with and without using Class Weights on Validation Dataset (October 2016)



(a) Performance of RC(Z) rain/no-rain Classification DNN without class weights on Test Dataset (September 2017, July 2018)



(b) Performance of RC(Z) rain/no-rain Classification DNN using class weights on Test Dataset (September 2017, July 2018)

Figure 4.16: Comparison of Performance of RC(Z) rain/no-rain Classification DNN with and without using Class Weights on Test Dataset (September 2017, July 2018)

From Figure 4.15 it can be seen that the F1 score of RC(Z) DNN trained using class weights has a better F1 score. This can be mainly rooted due the increased Recall value and a decent Precision value. It is true that the precision value decreased from 0.946 to 0.643 but in such critical applications where predicting rain cases is more important than predicting no-rain cases, having lesser False Negative value is advantageous, this tradeoff is important to get a good Recall value over Precision. [51], [52], [53] explain this tradeoff for prioritizing Recall over Precision for critical applications. Some applications also use an F_β Score by weighting Recall to analyze such cases [51]. From Figure 4.16 it can be seen that the F1 Score of Test dataset decreased by using class weight during training. This is due to the decreased value of precision as compared to non-class weights based approach. Even though the precision and F1 score decreased, it can be seen that the value of recall has increased and that was what was expected from the class weights based approach. Now that the model can gain good recall value, more vertical profile values of different moments can be added to see if it can improve the F1 Score and balance out the Precision and Recall distribution.

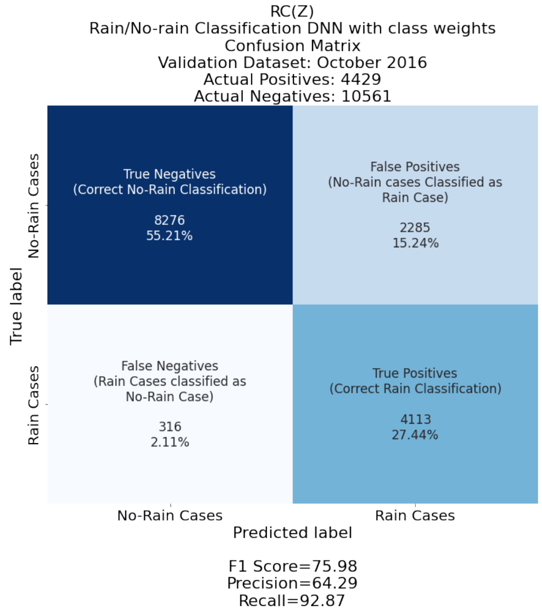
4.5.5 DNN based RC(Z-Z_{dr}) Rain/No-Rain Classification Model

Using only reflectivity for rain / no-rain classification gives out decent results but the precision of the model is quite low. From [15] it can be seen that Differential Reflectivity has a potential to observe hydrometeor types to some extent and can also model drop size distribution. Therefore, using vertical profile of Z_{dr} should improve the performance of the rain / no-rain classification. This model will be given a total of 16 values as input (8 values of vertical profiles of Z and 8 of vertical profile of Z_{dr}) taken at 0.5 km distanced regular heights from 0.5 km to 4 km above the rain gauge. The optimization algorithm used is still SGD but the learning rate was made smaller and set to 0.001. The batch size is also reduced to 32 keeping the number of epochs constant at 5. Smaller batch size was chosen to update the weights more frequently. With batch size of 32, the model trains slowly as compared to the RC(Z) model.

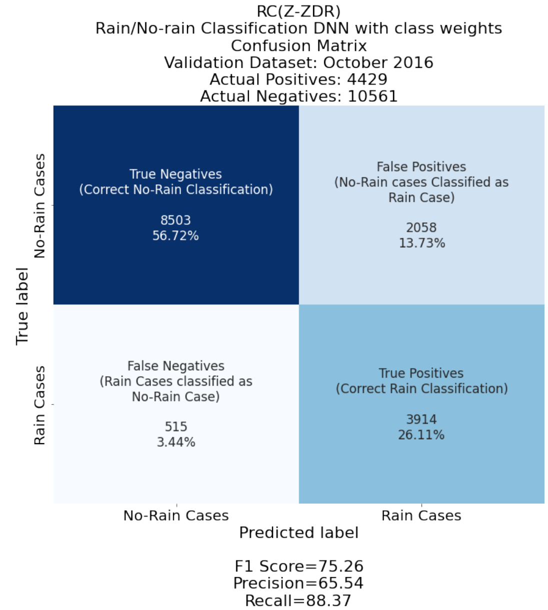
Model Description and Trainig Parameters

The model consists of 1 input layer, 6 hidden layers, and 1 output layer. The number of neurons in each layer was also increased to accommodate for the increased input size. As this is a binary classification problem, sigmoid was chosen as the activation function for the output layer. Out of the 6 hidden layers 3 layers are Dropout layers and 3 layers are Dense layers. In order to avoid overfitting of the model and gain more generalization, dropout layers were used. Dropout layers randomly inactivate few neurons during every forward pass. This forces the model to improve performance while relying on fewer neurons. Dropout layers are only used during training of the model. Figure A.2 shows a plot of the model architecture with details of the number of neurons used for each layer.

Figure 4.17 and Figure 4.18 compare the performance of the RC(Z) rain/no-rain classification DNN model and RC(Z-Z_{dr}) rain/no-rain classification model. Class weights were used during training of both the models.



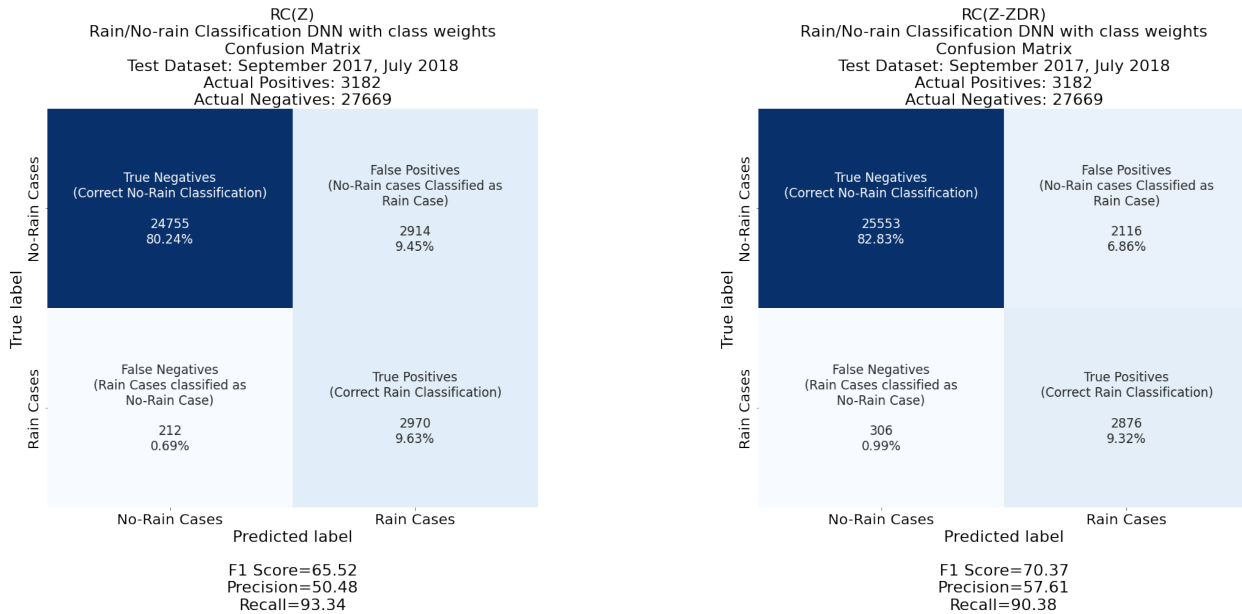
(a) Performance of RC(Z) rain/no-rain Classification DNN on Validation Dataset (October 2016).



(b) Performance of RC(Z-Z_{dr}) rain/no-rain Classification DNN on Validation Dataset (October 2016).

Figure 4.17: Comparison of Performance of RC(Z) and RC(Z-Z_{dr}) rain/no-rain Classification DNN on Validation Dataset (October 2016)

From Figure 4.17 it can be seen that using the vertical profile of Z_{dr} did not have a positive effect on the validation dataset. Moreover, the value of recall reduced by around 0.04. As there was no big degradation of performance observed, the test dataset should be used for further analysis as the validation dataset has lower number of sample rain events compared to test dataset. From Figure 4.18 it can be observed that using vertical profile of Z_{dr} for rain/no-rain classification increased the F1 Score. Moreover, the accuracy and precision of the model also increased with a little degradation in the value of recall. Overall it achieves the highest F1 score with a good balance between precision and recall values. Therefore we can conclude that using Z_{dr} improved the model performance. The value of precision is still less than 0.6, so it might be helpful to further explore if adding K_{dp} improves the performance.



(a) Performance of RC(Z) rain/no-rain Classification DNN on Test Dataset (September 2017, July 2018)

(b) Performance of RC(Z-Z_{dr}) rain/no-rain Classification DNN on Test Dataset (September 2017, July 2018)

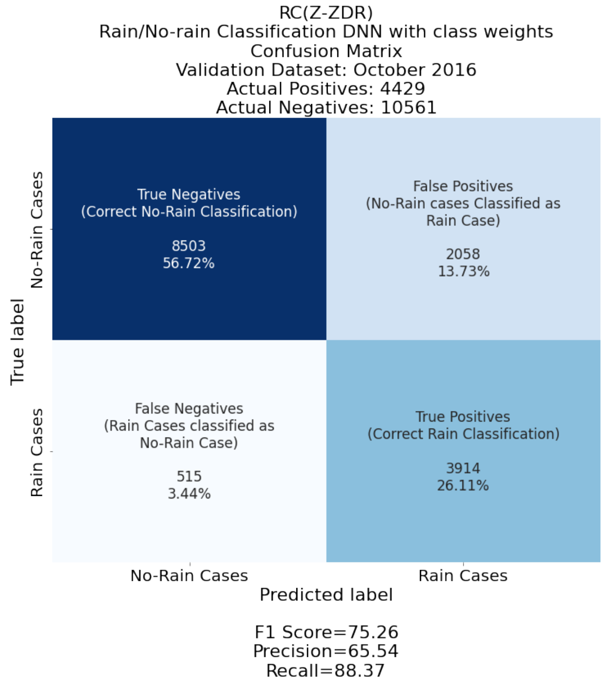
Figure 4.18: Comparison of Performance of RC(Z) and RC(Z-Z_{dr}) rain/no-rain Classification DNNs on Test Dataset (September 2017, July 2018)

4.5.6 DNN based RC(Z-Z_{dr}-K_{dp}) Rain/No-Rain Classification Model

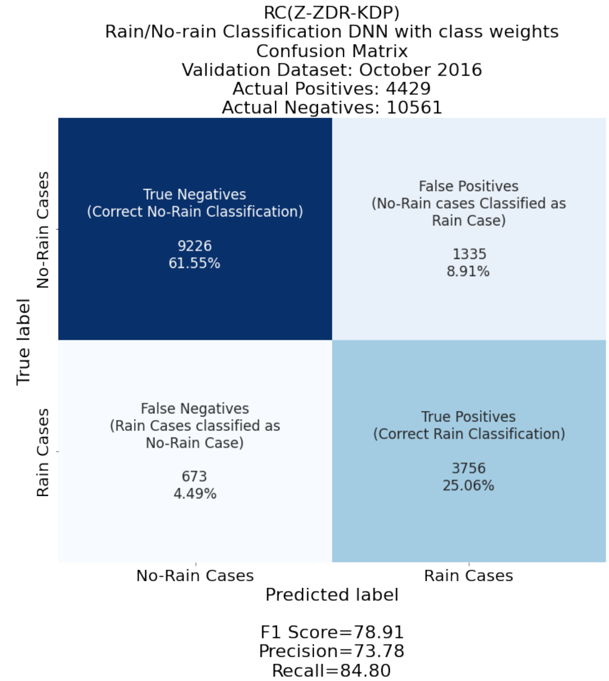
From equation 2.9 and the chapter 3 it is evident that K_{dp} has a relation with rain rate. Using vertical profiles of K_{dp} should also improve the performance of the model. This model will be given 24 values as input (8 values of vertical profiles of Z, Z_{dr} and K_{dp} each) taken at 0.5 km distanced regular heights from 0.5 km to 4 km above the rain gauge.The optimization algorithm used is still SGD but the learning rate was increased to 0.005. The batch size is also increased to 128 and epochs increased to 7. The model continues to learn until 7 epochs and startes to overfit training data beyond 8th epoch. Thus 7 epochs were chosen for training.

Model Description and Trainig Parameters

The model consists of 1 input layer, 6 hidden layers, and 1 output layer. Similar to the RC(Z-Z_{dr}) DNN structure. The output layer activation function is still sigmoid. Number of neurons in each hidden layer were increased to accommodate the increased number of inputs. Figure 4.19



(a) Performance of RC(Z-Z_{dr}) rain/no-rain Classification DNN on Validation Dataset (October 2016).



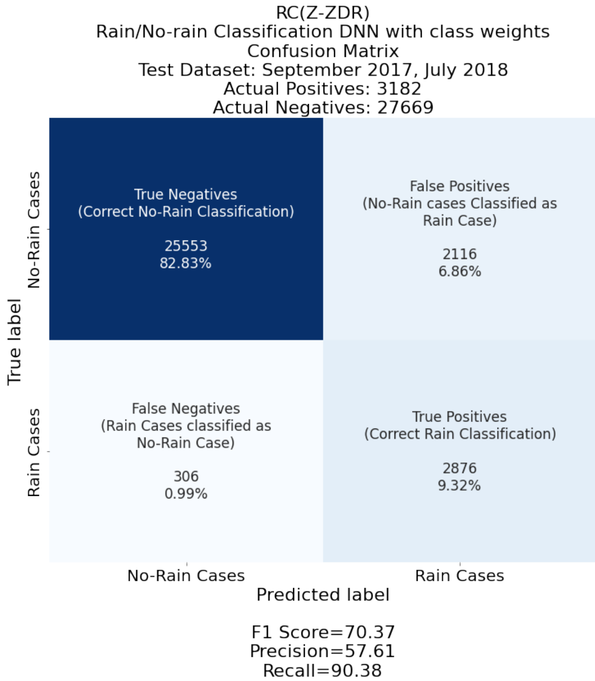
(b) Performance of RC(Z-Z_{dr}-K_{dp}) rain/no-rain Classification DNN on Validation Dataset (October 2016).

Figure 4.19: Comparison of Performance of RC(Z-Z_{dr}) and RC(Z-Z_{dr}-K_{dp}) rain/no-rain Classification DNN on Validation Dataset (October 2016)

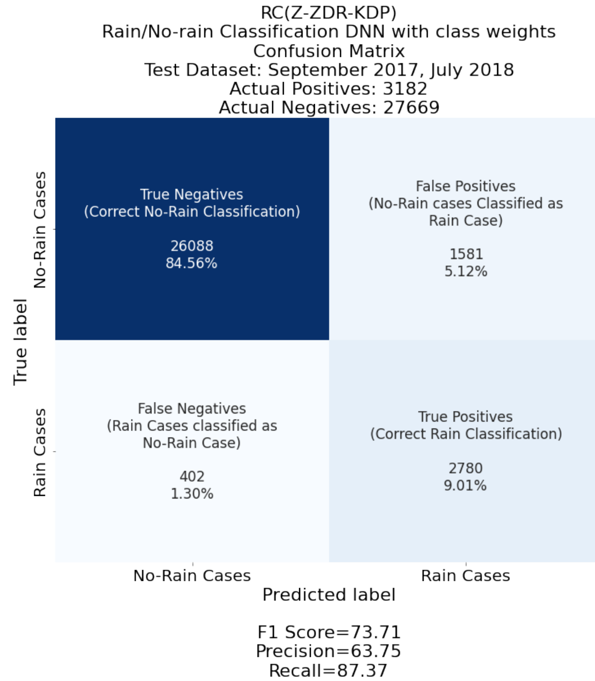
and Figure 4.20 compare the performance of the RC(Z-Z_{dr}) model rain/no-rain classification DNN model and RC(Z-Z_{dr}-K_{dp}) rain/no-rain classification model.

From Figure 4.19 it can be seen that the F1 score on the validation dataset increases by 0.08. This increase can be accounted due to an increase in the precision of the model. However the recall value decreases by 0.04. Nevertheless a recall value of 0.848 is still good enough.

From 4.20 it can be observed that the test dataset also shows a trend similar to the validation dataset. Precision increased by approx 0.06 and recall decreased by 0.03. From this result, it can be concluded that this model has a decent precision and recall balance with recall given slightly more importance than precision. Adding more moments for use as inputs to the DNN model was beneficial and lead to an incremental increase to the performance of the Rain/no-rain classification model.A.3 shows a plot of the model architecture with details of the number of neurons used for each layer.



(a) Performance of RC(Z-Z_{dr}) rain/no-rain Classification DNN on Test Dataset (September 2017, July 2018)



(b) Performance of RC(Z-Z_{dr}-K_{dp}) rain/no-rain Classification DNN on Test Dataset (September 2017, July 2018)

Figure 4.20: Comparison of Performance of RC(Z-Z_{dr}) and RC(Z-Z_{dr}-K_{dp}) rain/no-rain Classification DNNs on Test Dataset (September 2017, July 2018)

Comparison between different Rain / No-Rain classification DNN models

Table 4.5 and Table 4.6 compare the performance of different DNN based Rain / No-Rain classification models. Performance is compared mainly based on F1 score. As recall is slightly important in rain/no-rain classification alogrithms, the best model should have high recall and moderate precision. It can be seen that RC(Z-Zdr-KDP) rain / no-rain classification DNN model achieves the best F1 score on validation and test datasets with a good recall value (above 80%) and moderate precision (above 60%).

Table 4.5: Comparison of results between different Rain/No-rain Classification DNN models on Validation Dataset (October 2016)

Model/Algorithm Name	**F1 Score**	**Precision**	**Recall**
RC(Z) rain/no-rain classification DNN	67.03%	94.57%	51.91%
RC(Z) rain/no-rain classification DNN with class weights	75.98%	64.29%	92.87%
RC(Z-Z_{dr}) rain/no-rain classification DNN with class weights	75.26%	65.54%	88.37%
RC(Z-Z_{dr}-K_{dp}) rain/no-rain classification DNN with class weights	78.91%	73.78%	84.80%

Table 4.6: Comparison of results between different Rain/No-rain Classification DNN models on Test Dataset (September 2017, July 2018)

Model/Algorithm Name	**F1 Score**	**Precision**	**Recall**
RC(Z) rain/no-rain classification DNN	74.71%	85.91%	66.09%
RC(Z) rain/no-rain classification DNN with class weights	65.52%	50.48%	93.34%
RC(Z-Z_{dr}) rain/no-rain classification DNN with class weights	70.37%	57.61%	90.38%
RC(Z-Z_{dr}-K_{dp}) rain/no-rain classification DNN with class weights	73.71%	63.75%	87.37%

4.5.7 Comparison of RC(Z-Z_{dr}-K_{dp}) DNN model with Baseline Methods

The results of Z-Z_{dr}-K_{dp} based rain/no-rain DNN model are compared against two baseline methods: DROPS and NEXRAD DPR. All comparisons are judged based on their F1 scores. One more comparison is done using the RC(Z) DNN with RBF-NN based classification to compare the efficiency of RBFNN against DNN.

Radial Basis Function (RBF) Network-based rain/no-rain classifier

[30] used identical data from NEXRAD and TRMM Ground Validation gauges combination. The data structure used by [30] consisted of Vertical Profile of Reflectivity (Z) at 4 vertical heights (1.5, 2.5, 3.5, 4.5 km) and the horizontal resolution was 1km x 1km. It used a radial basis function to classify data into rain/no-rain conditions on ground. In order to compare between the classification efficiency between RBFNN and DNN, we will compare the results of RBFNN with R(Z) DNN on Aug/September 1995 dataset.

Table 4.7: Comparison of results between RBFNN and DNN based Rain/No-rain Classification on August/September 1995 Dataset

Model/Algorithm Name	**F1 Score**	**Precision**	**Recall**
RBFNN rain/no-rain classifier	66.70%	62.50%	59.00%
RC(Z) rain/no-rain classifier DNN with class weights	71.40%	59.00%	90.60%

Class Weights based RC(Z) DNN rain/no-rain classifier has a greater F1 score than the RBFNN rain/no-rain classifier on the August/September 1995 Dataset. RC(Z) DNN model also has a better recall value with only a little degradation in precision. This result proves the hypothesis that Deep Neural Networks are capable of achieving better modelling of rain-rate as compared to traditional Machine Learning methods.

Dual-Polarization Radar Operational Processing System (DROPS)

The DROPS framework is a non-ML-based approach that uses a combination of Z, Z_{dr}, K_{dp} to estimate rainfall. A combination of Z_{dr} and K_{dp} gives more information regarding DSD and hydrometeor type in consideration. As this information provides better information rainfall estimates, it performs well and is more accurate than the traditional Z-R equations. Non-Zero RR values were transferred to rain class (1) and zero-valued rain rates were considered as a no-rain class (0).

NEXRAD Digital Precipitation Rate

The NEXRAD Digital Precipitation Rate product is an instaneous rainfall rate derived from VCP scans. Dual-Polarization radar moments are used to compute the rain-rate. This product is readily available as a radar's derived product which is used in multiple actual hydrological and meteorological applications. To get Rain Class, Non-Zero RR values were transferred to rain class (1) and zero-valued rain rates were considered as a no-rain class (0). Figure 4.21 and Figure 4.22 show a comparison of performance between RC(Z-Z_{dr}-K_{dp}), DROPS and NEXRAD DPR rain/no-rain Classifications on Validation and Test datasets respectively.

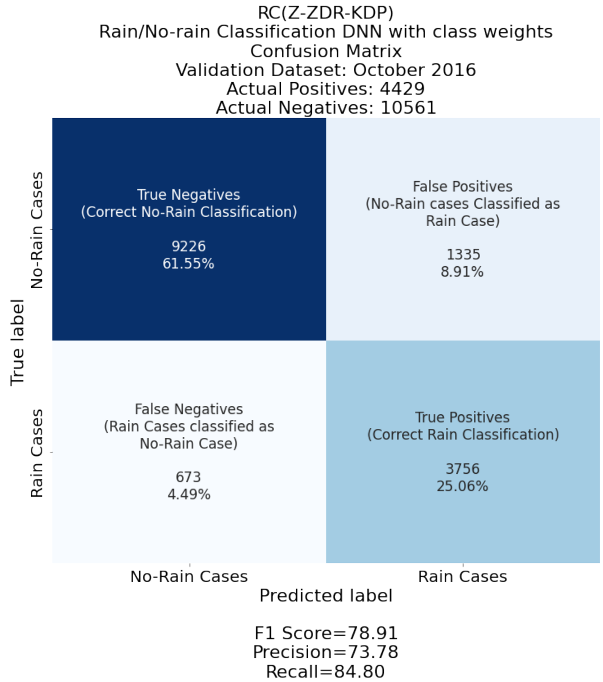
Table 4.8: Comparison of Rain/No-Rain classification performance of RC(Z-Z_{dr}-K_{dp}) DNN, DROPS and NEXRAD DPR algorithm on Validation Dataset (October 2016) obtained from KMLB NEXRAD site

Model/Algorithm Name	F1 Score	Precision	Recall
RC(Z-Z_{dr}-K_{dp}) DNN trained with class weights	78.91%	73.78%	84.80%
DROPS	66.00%	49.78%	97.91%
NEXRAD DPR	61.22%	44.42%	98.46%

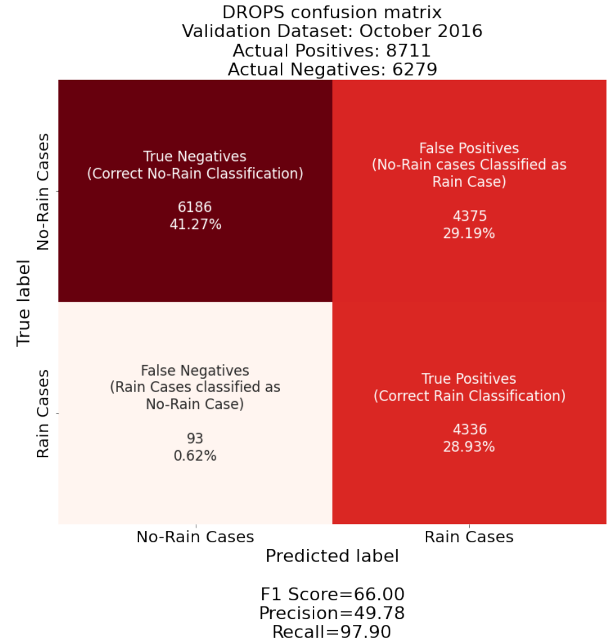
Table 4.9: Comparison of Rain/No-Rain classification performance of RC(Z-Z_{dr}-K_{dp}), DROPS and NEXRAD DPR algorithm on Test Dataset (September 2017, July 2018) obtained from KMLB NEXRAD site.

Model/Algorithm Name	F1 Score	Precision	Recall
RC(Z-Z_{dr}-K_{dp}) DNN trained with class weights	73.71%	63.75%	87.37%
DROPS	41.96%	26.83%	96.23%
NEXRAD DPR	63.54%	47.35%	96.54%

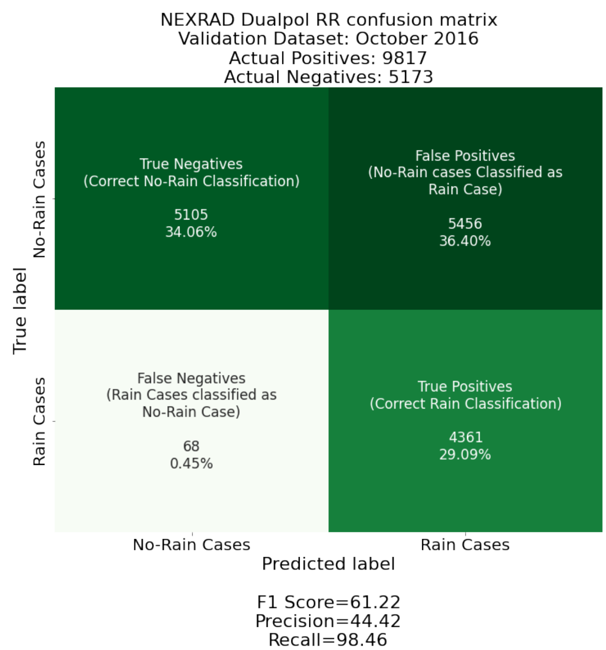
Table 4.8 and Table 4.9 show the comparison of performance of RC(Z-Z_{dr}-K_{dp}) DNN, DROPS and NEXRAD DPR methods. The Evaluation metrics are F1 Score, Recall and Precision. F1 Score and Recall are given more priority due to the nature of the application. From the table, it is evident that the RC(Z-Z_{dr}-K_{dp}) performs the best on both Validation as well as Test Datasets. The RC(Z-Z_{dr}-K_{dp}) DNN model's performance is followed by NEXRAD DPR and DROPS. Thus, this proves the hypothesis that Classification DNNs can be used as potential solutions in rain/no-rain classification problems.



(a) Performance of RC(Z-Z_{dr}-K_{dp}) rain/no-rain Classification DNN on Validation Dataset (October 2016)

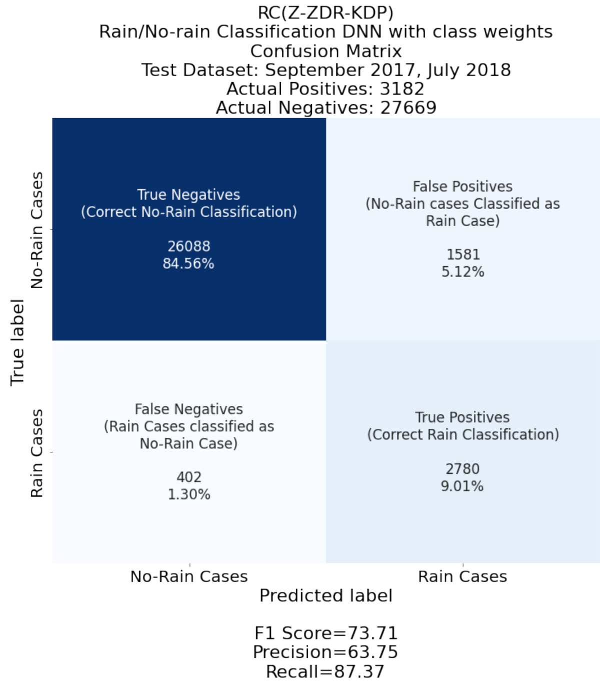


(b) Performance of DROPS rain/no-rain Classification Validation Dataset (October 2016)

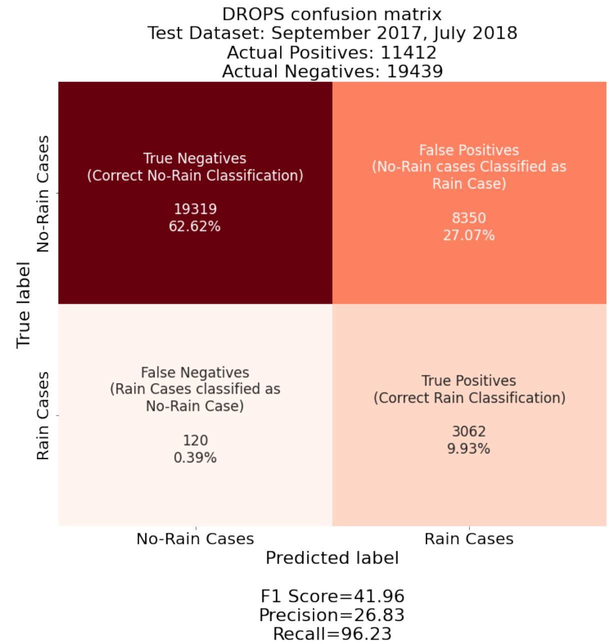


(c) Performance of NEXRAD DPR rain/no-rain Classification Validation Dataset (October 2016)

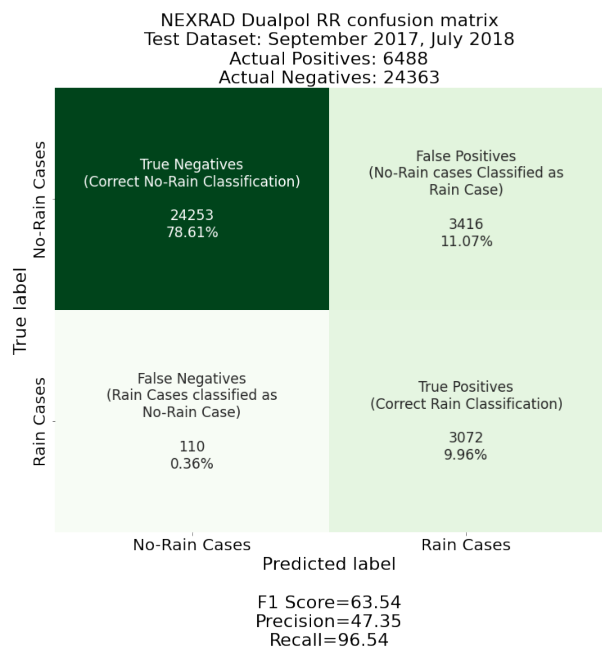
Figure 4.21: Comparison of Performance of RC(Z-Z_{dr}-K_{dp}), DROPS and NEXRAD DPR rain/no-rain Classifications on Validation Dataset (October 2016)



(a) Performance of RC(Z-Z_{dr}-K_{dp}) rain/no-rain Classification DNN on Test Dataset (September 2017, July 2018)



(b) Performance of DROPS rain/no-rain Classification Test Dataset (September 2017, July 2018)



(c) Performance of NEXRAD DPR rain/no-rain Classification Test Dataset (September 2017, July 2018)

Figure 4.22: Comparison of Performance of RC(Z-Z_{dr}-K_{dp}), DROPS and NEXRAD DPR rain/no-rain Classifications on Test Dataset (September 2017, July 2018)

4.6 Preliminary Results of Rain Rate estimation using DNN

Rain rate estimation is the process of estimating the amount of rain that will be collected on the ground in an hour. As Deep Neural Networks can also be used to estimate continuous values, they can be used to estimate rain rates as well. This thesis creates a DNN model that estimates rain rates in mm/h based on the vertical profiles of radar moments. These DNN models need to produce continuous valued rain rates and the values would usually be between 0 to 100 mm/h. These types of problems are called as regression problems and DNN models used to implement the solution are called Regression Deep Neural Networks.

4.6.1 Activation Function

Unlike, logistic regression models, regression models do not estimate discrete values, they can produce continuous values in a range. Therefore, there are lots of possibilities and explorable options for activation functions and loss functions in Regression DNN models. Different activation functions like tanh, ReLU, Leaky-ReLU, ELU were experimented with hidden layers during the training phase. tanh and ReLU activation functions produced the best performance. The performance of both of these activation functions produced very competitive results. As ReLU activation function produced slightly better results, it was chosen to be used for all experimentations further in this section.

In Regression DNN networks Linear activation function is always used in the output layer. The linear activation function is nothing but the function of a line. It helps in propagating the output of the DNN to the outside code. Figure 2.3 shows the equations and curves of these activation functions.

4.6.2 Loss Function

Mean Squared Error is used for calculating loss during the forward propagation phase of the network. The weights and biases are adjusted depending on the gradient value calculated based on the loss returned. MSE was preferred as the loss function over MAE as training with MSE provided a good sense of direction whereas MAE caused underestimation of rain cases. The loss is used to penalize the DNN and change weights and bias depending on the gradients during back propagation.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.12}$$

4.6.3 Evaluation Metrics

These Rain Rate estimation models are evaluated based on five metrics which calculate the difference between the estimated rain rate value and the original rain rate value. These same metrics are also used to calculate the quality of the baseline DROPS and NEXRAD DPR product as well.

Root Mean Squared Error

RMSE measures the average of the square root of the Mean Squared Error between actual values and the values estimated by the DNN model. The Equation for RMSE is shown in Equation (4.13).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{4.13}$$

Mean Absolute Error

MAE measures the average of the absolute difference between the actual values and values estimated by the DNN model. The Equation for MAE is shown in (4.14).

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|(Y_i - \hat{Y}_i)| \tag{4.14}$$

Correlation

Correlation Coefficient is used to measure the linear relationship between two variables. Correlation coefficient ranges between -1 and 1. A perfect negative correlation coefficient has a value of -1 and a perfect positive correlation coefficient has a value of 1. Positive correlation value indicates a positive slope and a negative correlation value indicates a negative slope of relationship. In this case, actual rain rate vs estimated rain rate should ideally be as close to 1 as possible. This measures the capability of the model to accurately estimate rain rate close to the actual rain rate measurement. The Equation for calculating correlation is shown in (4.15).

$$CORR = \frac{\frac{1}{N}\sum_{i=1}^{n}(Y_i - MEAN(Y_i))(\hat{Y}_i - MEAN(\hat{Y}_i))}{\sigma(Y) * \sigma(\hat{Y})} \tag{4.15}$$

Bias

Bias is the ratio of the difference between estimated rain rate values and actual rain rate values to the sum of the actual rain rate values. Bias value should be as close to 0% as possible. A bias close to 0%, indicates that the estimated values are close to the actual values. The formula for calculating bias is shown in (4.16).

$$BIAS = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)}{\sum_{i=1}^{n}\hat{Y}_i} * 100 \tag{4.16}$$

Normalized Standard Error

Normalized Standard error is the error of the model's estimation normalized to the actual observation values. The formula for calculating NSE is shown in (4.17). NSE value is given in percentage and ideally it should be as low as possible.

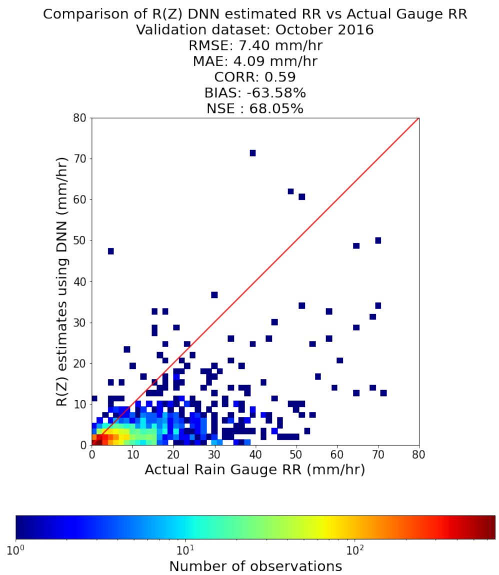
$$NSE = \frac{\sum_{i=1}^{n}|Y_i - \hat{Y}_i|}{\sum_{i=1}^{n}\hat{Y}_i} * 100 \tag{4.17}$$

4.6.4 R(Z) DNN Rain Rate Estimation Model

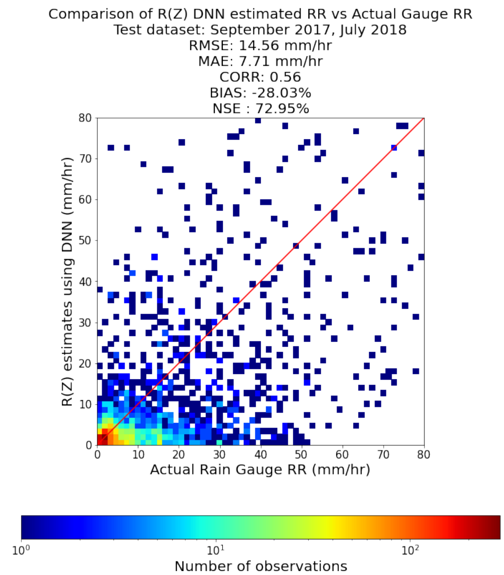
To start with a basic DNN model, vertical profile of reflectivity is used as the input to the model. Due to a good relationship of rain rate and Z, VPR would be a good feature for estimating rain rate. The input to this model consists of 8 values of reflectivity taken at 0.5 km distanced regular heights from 0.5 km to 4 km above the rain gauge. The output of the DNN model will be a continuous value specifying an estimate of rain rate that will be observed on the ground. We will call this model as R(Z) rain rate estimation DNN.

Training Parameters

During data pre-processing, VPR data was transformed to form sets of 8 normalized reflectivity values per input. Actual rain rate observed by the TRMM GV rain gauges are used as labels for training. Different types of optimizers were experimented with during training. Adam, Stochastic Gradient Descent and RMS_Prop were used and experimented with to find the optimizer that performed the best for this problem. It was found out that RMS_Prop optimizer gave the best results, thus it was finalized to be used for all the Regression DNN models in this work. The learning rate was set to 0.0005. The training was done using a batch size of 512 for 10 epochs. The loss value of model on validation set does not improve beyond 10 epochs and the model starts overfitting. Therefore training was done over 10 epochs. Density scatter plots with model estimated rain-rate on Y-axis and Actual Rain-rate on X-axis was used for analysis. Figure 4.23 shows the density plots of RR estimations of R(Z) DNN vs Actual RR from Gauge on Validation and Test Datasets. MSE, MAE, CORR and BIAS are used as evaluation metrics for all the Regression models.



(a) Density Plot of RR estimations of R(Z) DNN vs Actual RR from Gauge on Validation Dataset (October 2016)



(b) Density Plot of RR estimations of R(Z) DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)

Figure 4.23: Density Plots of RR estimations of R(Z) DNN vs Actual RR from Gauge on Validation and Test Datasets

4.6.5 R(Z) 2-Stage DNN Rain Rate Estimation Model

Machine learning models fail to make accurate estimations in QPE due to an imbalance in the training dataset. As discussed earlier, oversampling and undersampling either remove important input feature trends or introduce duplication which does not help in achieving a generalized model. Due to this issue, lots of moderate rain events get estimated as no-rain events. This is responsible for increased error values in rain-rate estimates if the same data is used to train an ML model that estimates rain-rate.

To solve this issue, this work uses the Rain/No-Rain Classification DNN in cascade with a Regression DNN for RR estimation. The pretrained RC(Z-Z_{dr}-K_{dp}) DNN is cascaded with the R(Z) DNN for improved rain rate estimation. We will call this network R(Z) 2-Stage DNN. This model ingests vertical profiles of Z for rain rate estimation. Rain/No-Rain Classification DNN takes in normalized vertical profiles of Z, Z_{dr} and K_{dp} and gives a rain/no-rain classification as output. This output is fed into the R(Z) DNN which produces rain rate estimations. The flow and structure of the 2-Stage DNN Model for RR estimation is shown in the Figure 4.24.

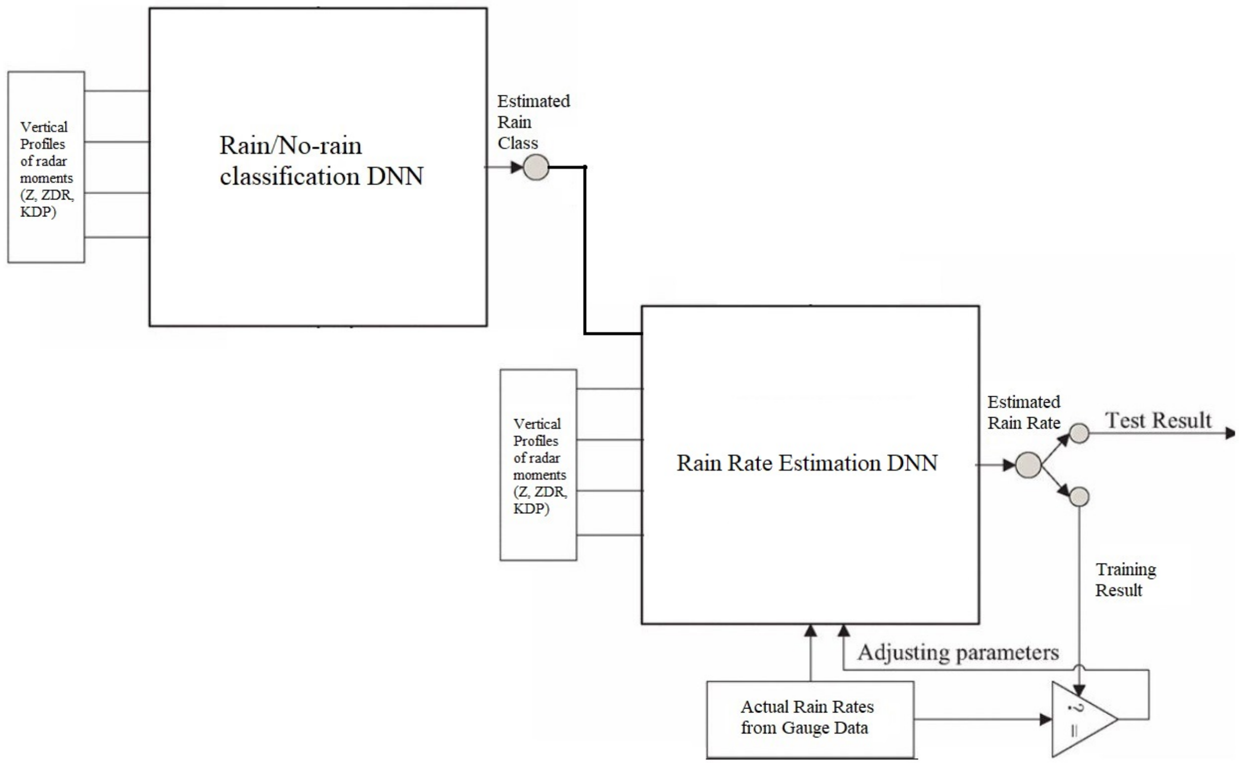
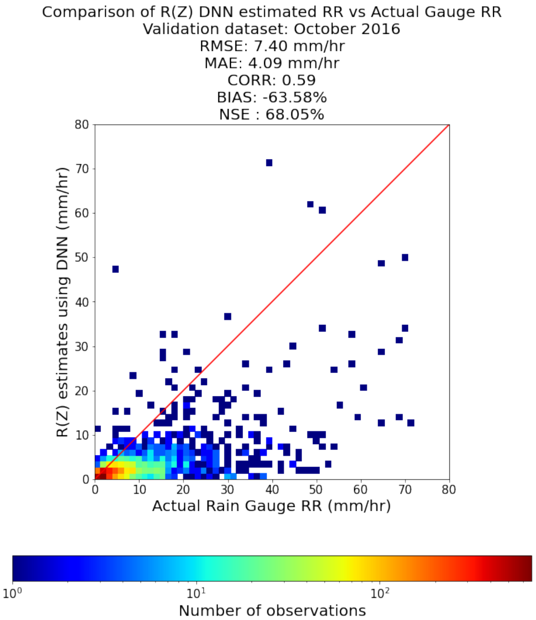


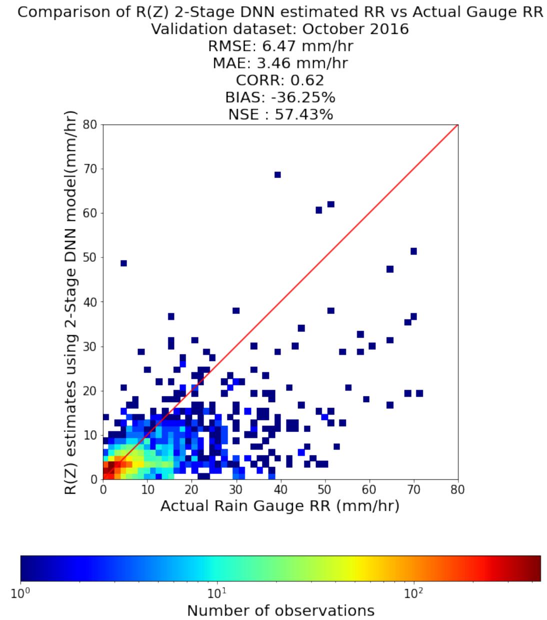
Figure 4.24: Workflow of Rain Rate Estimation using 2-Stage DNN QPE Model

24 values of vertical profile of radar consisting of Z, Z_{dr} and K_{dp} are given as input to the Rain/No-Rain Classification DNN which gives out a rain/no-rain classification. This value is appended with the 8 values of radar reflectivity vertical profile. Finally this 9 valued input vector is supplied to the R(Z) DNN model which estimates the rain rate value seen at the surface of the gauge location. This whole setup is summed up into a single model called the R(Z) 2-Stage DNN QPE Model. Same model architecture and training hyperparameters as R(Z) RR estimation DNN are used for training R(Z) 2-Stage DNN RR estimation model.

The R(Z) RR estimation DNN consists of 1 input layer, 2 hidden layers, and 1 output layer. Both the hidden layers are Dense layers and all layers use float32 precision for weights and activations. After experimenting with different types of activation functions, the ReLU activation



(a) Density Plot of RR estimations of R(Z) DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

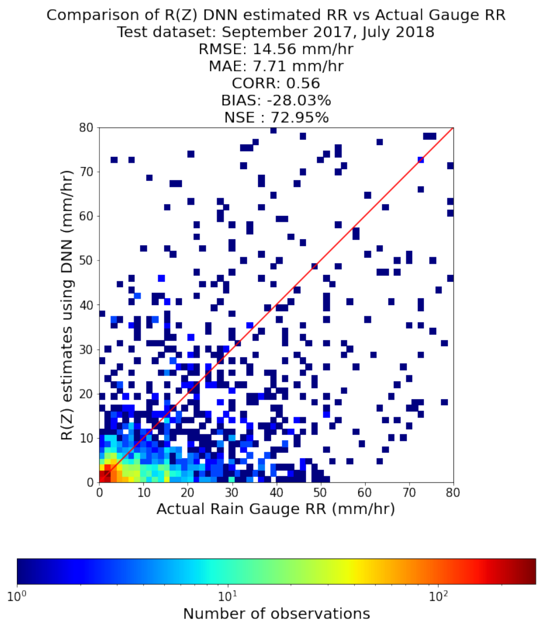


(b) Density Plot of RR estimations of R(Z) 2-Stage DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

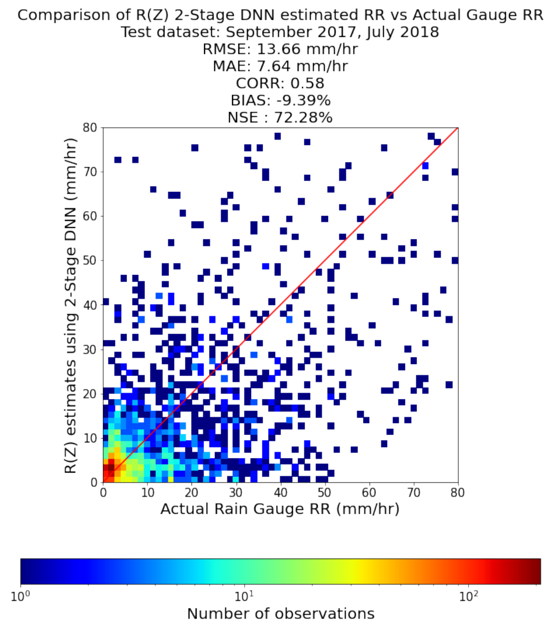
Figure 4.25: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z) DNN and R(Z) 2-Stage DNN on Validation Dataset

function was chosen for use in all the hidden layers. The rain-rate estimates from the rain-gauges are used as labels for the training phase. RMSprop is used as the optimizer for training. The learning rate is set to 0.0001, with momentum = 0.1 and Discounting Factor(rho) =0.7. This model is trained for 10 epochs. Linear activation function was used in the output layer. Figure B.1 shows the architecture of the model.

From Figure 4.25 and Figure 4.26, it can be seen that the Validation set and Testing dataset showed an improvement in the RMSE and NSE values. Even though there was an improvement, there is more opportunity for improvement and adding more moments might be beneficial.



(a) Density Plot of RR estimations of R(Z) DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)



(b) Density Plot of RR estimations of R(Z) 2-Stage DNN vs Actual RR from Gauge Test Dataset (September 2017, July 2018)

Figure 4.26: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z) DNN and R(Z) 2-Stage DNN on Test Dataset

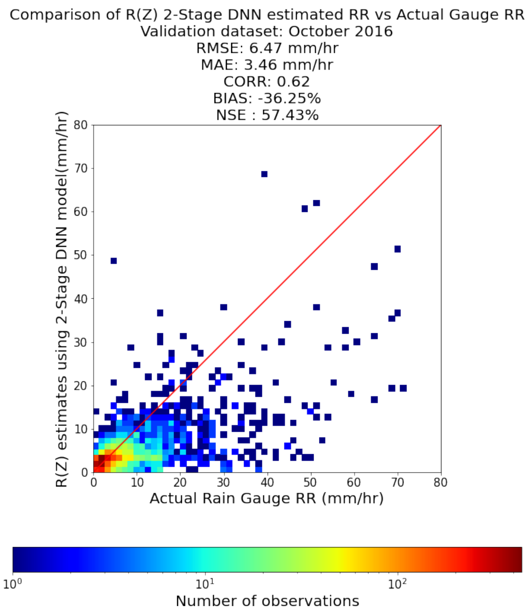
4.6.6 R(Z-Z_{dr}) 2-Stage DNN QPE Model

Empirical relationships show a relationship between rain rate and Z_{dr}. Due to this relation, the value of Z_{dr} should improve the rain rate estimation of the DNN model. The input to this model is a 17 valued vector (8 values of vertical profiles of Z and Z_{dr} respectively, and 1 value from the RC(Z-Z_{dr}-K_{dp}) rain/no-rain classification model.

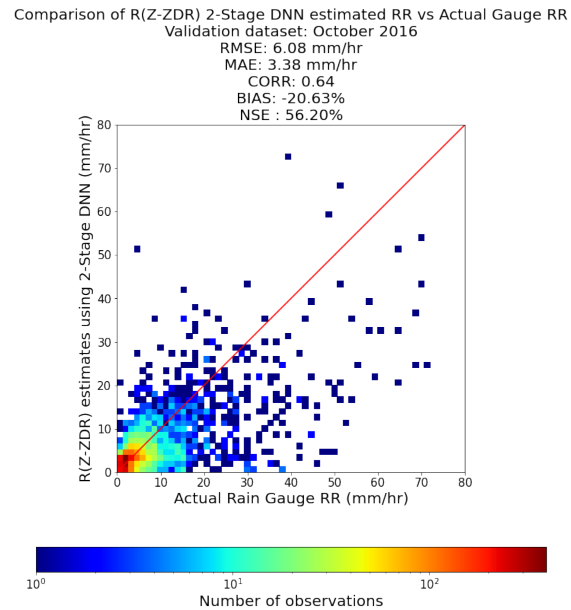
Model Desrciption and Training Parameters

The R(Z-Z_{dr}) 2-Stage DNN consists of 1 input layer, 4 hidden layers and 1 output layer. All the hidden layers are Dense layers in the model. The learning rate is 0.0001, with momentum and discount factor values same as before. This model is trained for 10 epochs keeping the batch size 200. Figure B.3 shows the architecture of the model.

From Figure 4.27 and Figure 4.28 it is evident that adding Z_{dr} improved the evaluation metrics on the data, it definitely helps in solving the underestimation problem of the model. The model is

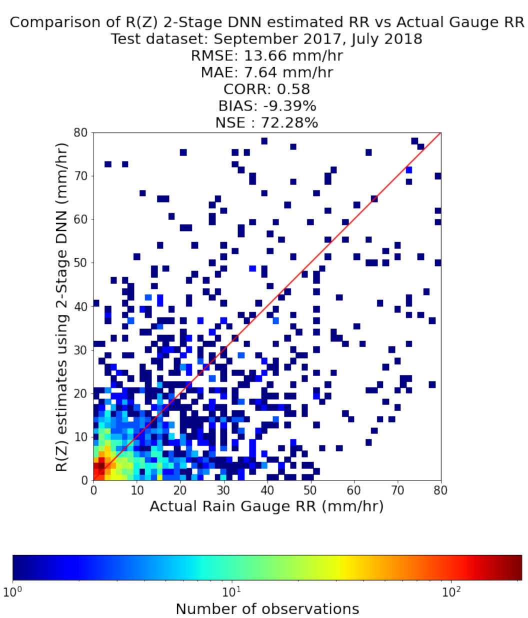


(a) Density Plot of RR estimations of R(Z) 2-Stage DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

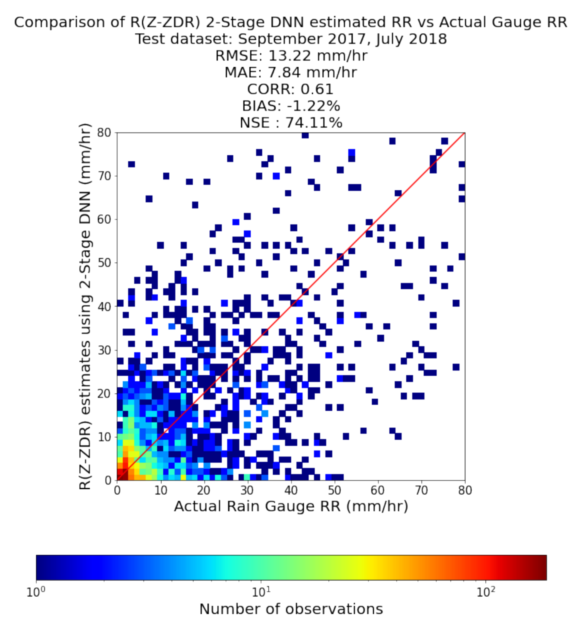


(b) Density Plot of RR estimations of R(Z-Z_{dr}) 2-Stage DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

Figure 4.27: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z) 2-Stage DNN and R(Z-Z_{dr}) 2-Stage DNN on Validation Dataset



(a) Density Plot of RR estimations of R(Z) 2-Stage DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)



(b) Density Plot of RR estimations of R(Z-Z_{dr}) 2-Stage DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)

Figure 4.28: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z) 2-Stage DNN and R(Z-Z_{dr}) 2-Stage DNN on Test Dataset

capable of estimating high values of rain rate. Though this is good, the evaluation metrics can be improved more by adding K_{dp} as there is a good relationship between K_{dp} and RR.

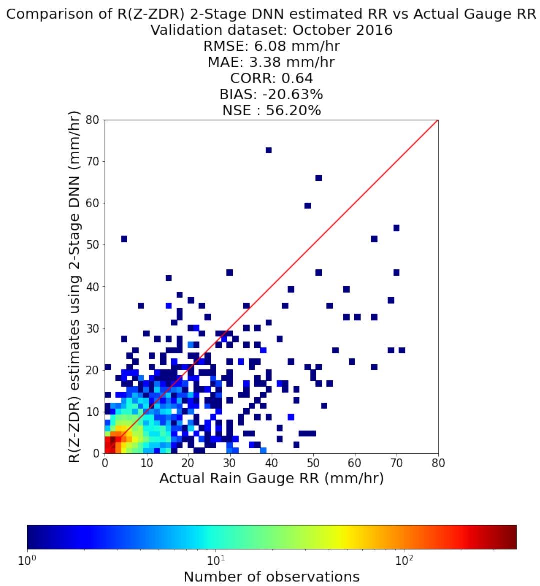
4.6.7 R(Z-Z_{dr}-K_{dp}) 2-Stage DNN QPE Model

Many non-ML-based models use K_{dp} along with Z and Z_{dr} to estimate the rain rate. The expression in (2.9) clearly shows a relationship between K_{dp} and rain rate. This section tries to improve the performance of the DNN model by using vertical profiles of K_{dp} along with Z and Z_{dr} to make rain-rate estimations. The input to this model is a 25 valued vector (8 values of vertical profiles of Z, Z_{dr} and K_{dp} respectively, and 1 value from the RC(Z-Z_{dr}-K_{dp}) rain/no-rain classification model.)

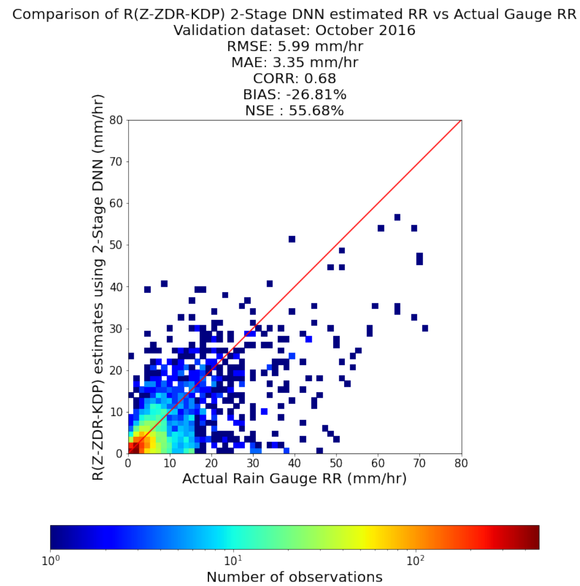
Model Desrciption and Training Parameters

The R(Z-Z_{dr}-K_{dp}) 2-Stage DNN consists of 1 input layer, 4 hidden layers and 1 output layer. All the hidden layers are Dense layers in the model. The learning rate is reduced to 0.0001, with momentum = 0.9 and discount factor value same as before. This model is trained for 15 epochs. Figure B.4 shows the architecture of the model.

From Figure 4.29 and Figure 4.30 we can conclude that using vertical profile of K_{dp} improved the RMSE and NSE of Test Dataset. So, it can be concluded that the R(Z-Z_{dr}-K_{dp}) gave the best performance among all the RR DNN models trained so far.

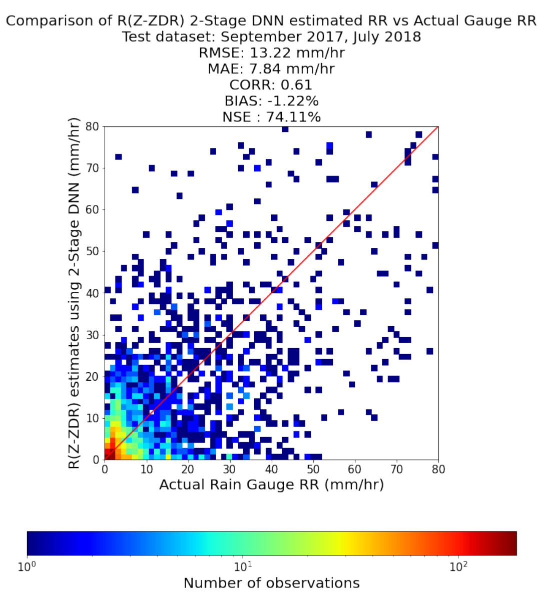


(a) Density Plot of RR estimations of R(Z-Z_{dr}) 2-Stage DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

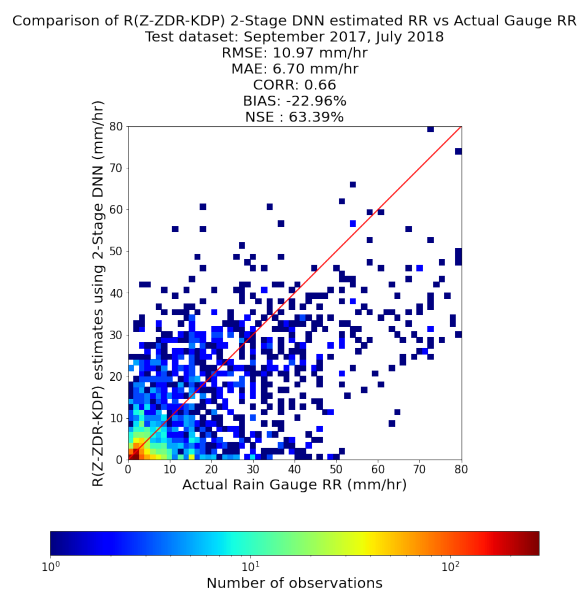


(b) Density Plot of RR estimations of R(Z-Z_{dr}-K_{dp}) 2-Stage DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

Figure 4.29: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z-Z_{dr}) 2-Stage DNN and R(Z-Z_{dr}-K_{dp}) 2-Stage DNN on Validation Dataset (October 2016)



(a) Density Plot of RR estimations of R(Z-Z_{dr}) 2-Stage DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)



(b) Density Plot of RR estimations of R(Z-Z_{dr}-K_{dp}) 2-Stage DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)

Figure 4.30: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z-Z_{dr}) 2-Stage DNN and R(Z-Z_{dr}-K_{dp}) 2-Stage DNN on Test Dataset (September 2017, July 2018)

4.6.8 Comparison of Results from different RR estimation DNN models

Table 4.10 and Table 4.11 compare the performance of different DNN based rain rate estimation models. Performance is compared mainly based on RMSE and NSE values. The best model should have minimum values of RMSE, NSE and MAE. The value of CORR and Bias should be as close to 1 as possible. It can be seen that R(Z-Zdr-KDP) 2-stage DNN model achieves the best performance on almost all evaluation metrics. Thus, this is the best DNN model and it will be used for comparison with baseline models.

Table 4.10: Comparison of performance of different RR estimation DNN models on Validation Dataset (October 2016) obtained from KMLB NEXRAD site

Model/Algorithm Name	RMSE (mm/h)	NSE	CORR	MAE (mm/h)	BIAS
R(Z) DNN	7.40	68.05%	0.59	4.09	-63.58%
2-Stage R(Z) DNN	6.47	57.43%	0.62	3.46	-36.25%
2-Stage R(Z-Z_{dr}) DNN	6.08	56.20%	0.64	3.38	-20.63%
2-Stage R(Z-Z_{dr}-K_{dp}) DNN	5.99	55.68%	0.68	3.35	-26.81%

Table 4.11: Comparison of performance of different RR estimation DNN models on Test Dataset (September 2017, July 2018) obtained from KMLB NEXRAD site

Model/Algorithm Name	RMSE (mm/h)	NSE	CORR	MAE (mm/h)	BIAS
R(Z) DNN	14.56	72.95%	0.56	7.71	-28.03%
2-Stage R(Z) DNN	13.66	72.28%	0.58	7.64	-9.39%
2-Stage R(Z-Z_{dr}) DNN	13.22	74.11%	0.61	7.84	-1.22%
2-Stage R(Z-Z_{dr}-K_{dp}) DNN	10.97	63.39%	0.66	6.70	-22.96%

4.6.9 Comparison with Dual-Polarization Radar Operational Processing System (DROPS) and NEXRAD Dual-Pol Precipitation Rate (DPR)

The DROPS framework and NEXRAD DPR Product were again chosen for comparison. The rain-rate values obtained from these softwares were used without any additional processing. Scatter plots with Rain rate estimations on Y-axis and Actual rain rate values on X-axis are used for visualization of the output. A standard line with linear equation having slope =1 and y-intercept = 0 is drawn on the plots for reference. The observed values on the plots should be as close to the linear line as possible to achieve the lowest RMSE, MAE and NSE. Density scatter plots are also used to observe the density pattern of estimations in the result.

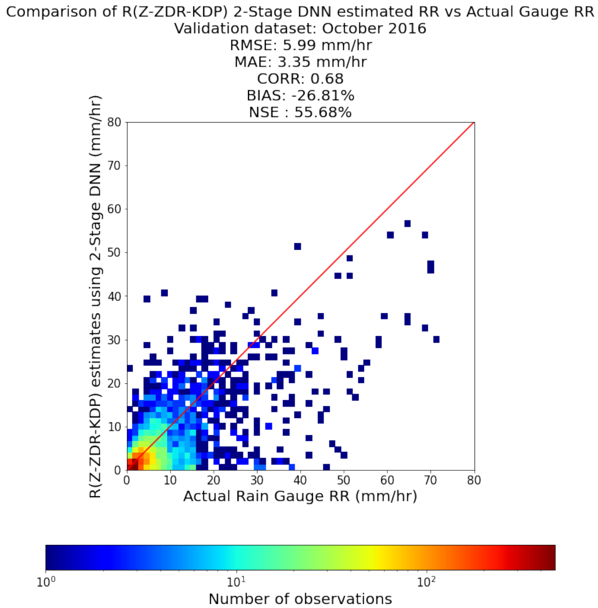
Table 4.12: Comparison of R(Z-Z_{dr}-K_{dp}) RR estimation, DROPS and NEXRAD DPR algorithm on Validation Dataset (October 2016) obtained from KMLB NEXRAD site

Model/Algorithm Name	RMSE (mm/h)	NSE	CORR	MAE (mm/h)	BIAS
2-Stage R(Z-Z_{dr}-K_{dp}) DNN	5.99	55.68%	0.68	3.35	-26.81%
DROPS	6.96	67.35%	0.65	4.05	-61.56%
NEXRAD DPR	15.20	190.59%	0.46	11.47	165.79%

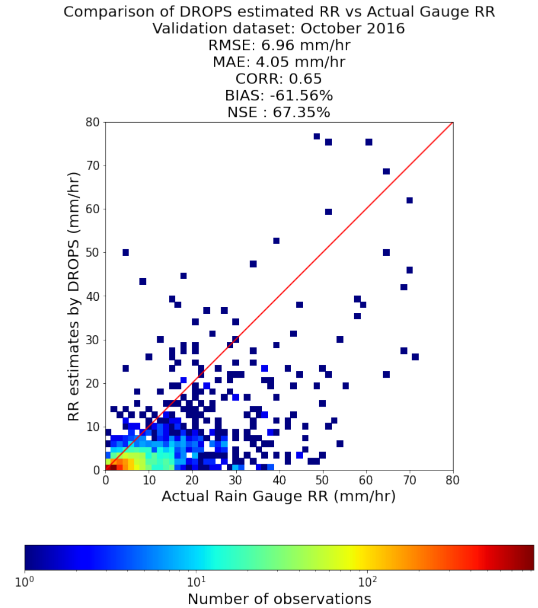
Table 4.13: Comparison of R(Z-Z_{dr}-K_{dp}) RR estimation, DROPS and NEXRAD DPR algorithm on Test Dataset (September 2017, July 2018) obtained from KMLB NEXRAD site

Model/Algorithm Name	RMSE (mm/h)	NSE	CORR	MAE (mm/h)	BIAS
2-Stage R(Z-Z_{dr}-K_{dp}) DNN	10.97	63.39%	0.66	6.70	-22.96%
DROPS	11.41	64.02%	0.66	6.77	-25.94%
NEXRAD DPR	18.53	141.52%	0.36	14.96	102.02%

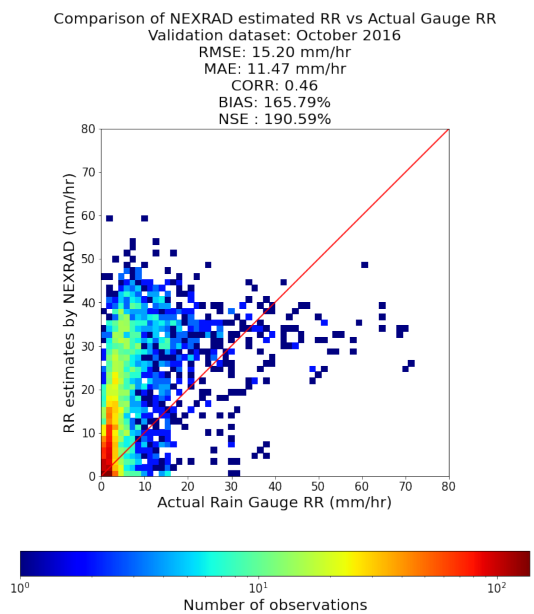
Referring to Figure 4.31 and 4.32 it can be concluded that the R(Z-Z_{dr}-K_{dp}) 2-Stage DNN RR estimation model has the best values of all evaluation metrics on both the validation and test dataset. The results of DROPS and R(Z-Z_{dr}-K_{dp}) are quite comparable and are better than the NEXRAD



(a) Density Plot of RR estimations of R(Z-Z_{dr}-K_{dp}) 2-Stage DNN vs Actual RR from Gauge on Validation Dataset (October 2016)

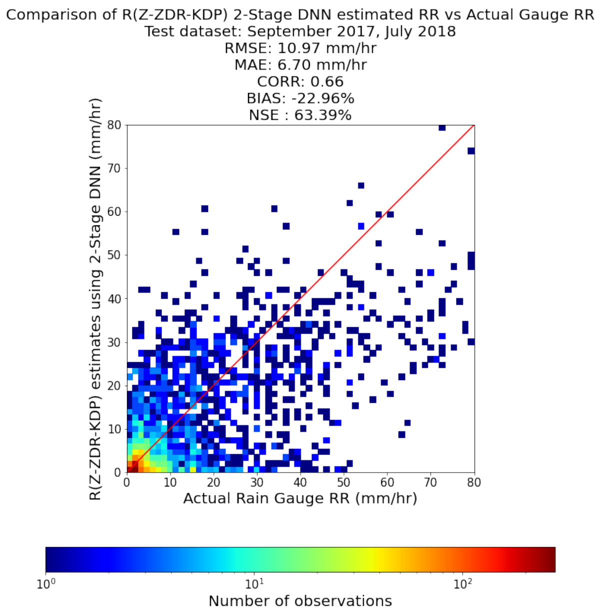


(b) Density Plot of DROPS RR estimations vs Actual RR from Gauge on Validation Dataset (October 2016)

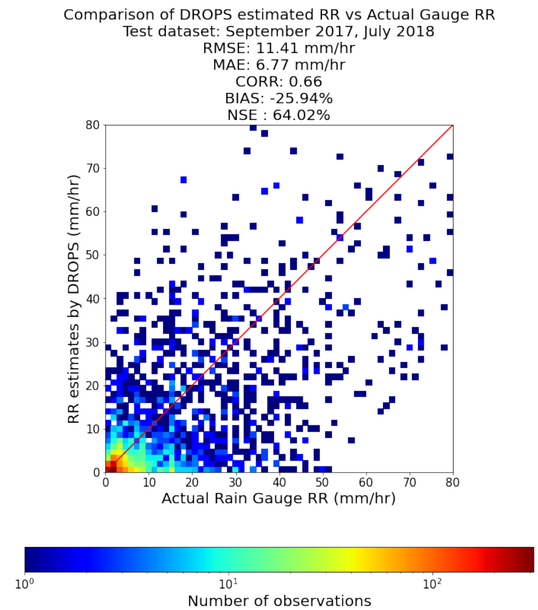


(c) Density Plot of NEXRAD DPR RR estimations vs Actual RR from Gauge on Validation Dataset (October 2016)

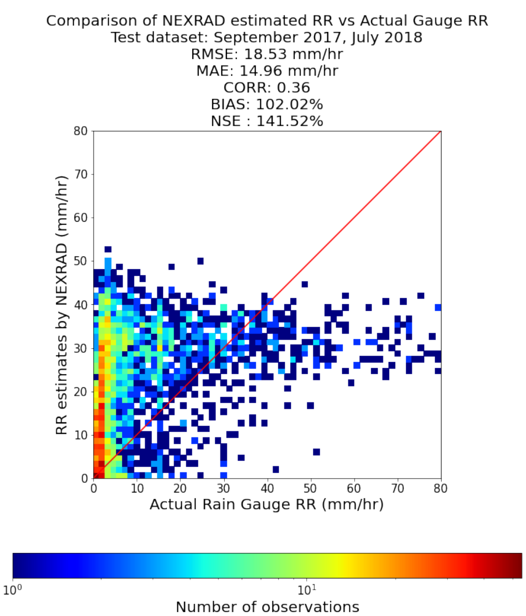
Figure 4.31: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z-Z_{dr}-K_{dp}) 2-Stage DNN, DROPS and NEXRAD DPR on Validation Dataset



(a) Density Plot of RR estimations of R(Z-Z_{dr}-K_{dp}) 2-Stage DNN vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)



(b) Density Plot of DROPS RR estimations vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)



(c) Density Plot of NEXRAD DPR RR estimations vs Actual RR from Gauge on Test Dataset (September 2017, July 2018)

Figure 4.32: Comparison of Density Plots of Estimated RR vs Actual RR between R(Z-Z_{dr}-K_{dp}) 2-Stage DNN, DROPS and NEXRAD DPR on Test Dataset

DPR. Hence, DNN based rain rate estimation models can be used as alternative approaches to empirical equations based algorithms and some other non-ML based algorithms.

Table 4.12 and Table 4.13 show the comparison of evaluations of Rain Rates obtained from R(Z-Z_{dr}-K_{dp}), DROPS and NEXRAD DPR. The Evaluation metrics are RMSE, CORR, MAE and BIAS. From the table, it is evident that the R(Z-Z_{dr}-K_{dp}) performs the best on both Validation as well as Test Datasets. The R(Z-Z_{dr}-K_{dp}) DNN model's performance is followed by DROPS and NEXRAD DPR. The performance of DROPS and R(Z-Z_{dr}-K_{dp}) are very close and quite comparable. Thus, this proves the hypothesis that DNNs can be used as potential RR estimation techniques.

4.7 Environment and Software

The training was done using Google Colab's CPU/GPU cluster. Google Colab has a high-speed cluster of Graphical Processing Units(GPU), Central Processing Units (CPU), and Tensor Processing Units (TPU) that make training seamless and fast. CPUs in these clusters use multi-core Intel Xeon Processors with a 2.2 GHz frequency. The GPUs in this cluster are Nvidia's Tesla P100 GPU with 16GB memory. All data processing and ML training/inference was done using Python programming language. 4.14 lists out all the libraries used along with their versions.

Table 4.14: Description of Python Libraries

Python Library	Version	Description
Python	3.7.12	Python Language
Pyart	1.11.8	Radar data processing library
Nexradaws	1.1	AWS data repository of NEXRAD data
Cartopy	0.20.2	Import Maps
Requests	2.23.0	Query and web scrape data
Geopy	1.17.0	Calculate geographical distances
Conda	4.9.2	Python packages platform
Tensorflow	2.8.0	Machine Learning Library
Keras	2.8.0	API for Machine Learning

Chapter 5

Discussion

5.1 Summary

This research aimed to propose Deep Neural Networks as an alternative solution to various tasks in QPE like rain rate estimation and rain flagging. QPE is mostly done using empirical relations like ZR-Z, R-K_{dp}, or R-Z_{dr} or using composite models based on these relations. This research achieves results comparable to current QPE methods, proving that DNNs are a potential alternative for rain rate estimation. Likewise, DNNs can also be used in solving several other problems in Meteorology.

Datasets consisting of several months of data were gathered for use in this research. As vertical profiles of Dual-Pol Radars are more useful in estimating the rain condition on the surface, vertical profiles from NEXRAD were decided to be used. NEXRAD data is available in Volume Coverage Pattern form which is a Polar Coordinate-based representation, it cannot be used directly. To obtain the vertical profiles, cartesian gridding was done on this data. As a supervised machine learning technique was used in this research, labels were required to train models based on vertical profiles of the radar. These labels consisted of rain rates from the Ground Validation Gauges of NASA's TRMM and GPM Satellites. As the spatial and temporal resolution of NEXRAD and rain gauges did not match, they needed to be aligned in space and time. Spatial aligning was done by mapping the radar's closest gridpoint with a rain gauge. Temporal aligning was done by averaging the rain rate within +/- 2.5 minutes of the radar timestamp. All the aligned datasets were grouped into different datasets called training, validation, testing, and August/September 1995 dataset.

The research methodology starts with creating a DNN for the classification of events based on rain/no-rain conditions observed on the ground. The objective of the model was to classify events into rain/no-rain classes based on the vertical profile of reflectivity. As this is a binary classification problem, a logistic regression model was used. This model is named RC(Z) Rain/No-

Rain classification DNN. The Rain/No-rain classification DNN is given 16 values of reflectivity observed at different heights from 0.5km to 4km with a resolution of 0.5km. The model gives a value between 0 to 1 as the output. This is the probability of the event being a rain event. This value is binarized to give out a definite rain/no-rain classification. The binarization threshold was given as 0.5. As this is a binary classification problem, Binary Crossentropy (also called log loss) was used as the loss function. The metrics used were True Positive, True Negative, False Positive, False Negative, Precision, Recall, and F1 Score. Problems in meteorology need to have a lower number of False Negatives as undetected rain is not the desired observation. To solve this, the F1 Score is used as the primary metric over accuracy. Moreover, datasets in meteorology are always skewed towards no-rain cases. The ratio of rain events to no rain events in the dataset used in this research is around 4.48%. Even if the model classifies all events as no rain cases, the model still achieves an accuracy of 95.52%. Whereas the recall of this model is very low. As the F1 Score is calculated based on precision-recall tradeoff, it is a good metric in such cases. As almost all events in the dataset are no-rain cases, the model's training also gets skewed towards no-rain cases. This problem of biased training comes from biased datasets and techniques need to be used to tackle them. Among the three probable solutions are undersampling, oversampling, and using class weights. Adding class weights to the Binary Crossentropy loss was used as the solution because it handles data imbalance during training without polluting the dataset. This significantly improved the F1 Score of the model. The RC(Z) Rain/No-Rain classification DNN was compared with RBFNN and DROPS Baseline Approaches and it was found that DNN based classification had a better F1 Score in all compared datasets. To further improve the F1 score of the model, vertical profiles of Z_{dr} and K_{dp} were added one by one to the rain / no-rain classification DNN. The models are named RC(Z-Z_{dr}) and RC(Z-Z_{dr}-K_{dp}) rain/no-rain classification DNN. It was observed that addition of every new moment as input to the DNN model imporved the performance of the DNN model. Analysis was done at every step using Confusion Matrices. Finally the rain/no-rain estimate from RC(Z-Z_{dr}-K_{dp}) model was compared with the estimations from DROPS framework and NEXRAD DPR product. Positive rain rates were considered as rain phenomenon and zero rain rate cases were considered as no-rain cases. It was found that RC(Z-Z_{dr}-K_{dp}) gives the best performance on the test as well as validation datasets.

The next challenge for DNN was to estimate the rain rate based on radar data. Rain rate estimation is a very important process in meteorology and hydrology as a lot of water, construction, and infrastructure management decisions are taken based on rain rate estimations. Accurate estimation of rain rate from radars is difficult as rain is dependent on multiple factors like wind, DSD, Drop size, temperature, and many other parameters. Primarily vertical profile of reflectivity was used to train a DNN model to estimate rain rates. This model is named R(Z) RR estimation DNN. It was found that performance degradation can be caused due to no-rain cases being estimated with low-rain values. This is due to the inability of the model to effectively differentiate between rain and no-rain cases. To solve this issue, a 2-Stage DNN QPE model is proposed that utilizes the RC(Z-Z_{dr}-K_{dp}) Rain/No-Rain classification DNN along with a regression model named R(Z) RR Estimation DNN to estimate rain rate. As the classification model is already capable of distinguishing between rain and no-rain events, the rain classification obtained from this model can be used as a confidence value to significantly remove no rain events from being estimated in the low rain category (0 mm/h - 15mm/h). The output of the classification DNN is appended with the data of vertical profile of radar to be passed to the R(Z) 2-Stage DNN RR Estimation model. This research also proposes the use of vertical profiles of Z_{dr} and K_{dp} to estimate RR. Even though relationships between Z, Z_{dr}, K_{dp} and RR exist, Machine Learning models present today do not use Z_{dr} and K_{dp} for rain estimation. Vertical profiles of Z_{dr} and K_{dp} were also added to be used in the DNN for better accuracy. These models are named R(Z-Z_{dr}) and R(Z-Z_{dr}-K_{dp}) RR estimation DNN. These models try to minimize the MSE Loss while training. RMSE, MAE, NSE, CORR and BIAS are used as metrics for analyzing the results of the model. The best DNN model used Z, Z_{dr} and K_{dp} for RR estimation. The rain rate estimated by the R(Z-Z_{dr}-K_{dp}) 2-Stage DNN QPE model is compared with DROPS and NEXRAD DPR rain-rates as the baseline comparison method. It was found that R(Z-Z_{dr}-K_{dp}) RR estimation 2-Stage DNN model and DROPS have comparably similar MAE, CORR and BIAS values with the DNN model having better MSE. These results validate that Deep Neural networks can act as alternate approaches in Precipitation Estimation.

5.2 Future Scope

There are multiple paths using optimization techniques in DNN for obtaining accurate Rainfall Estimations. Machine Learning is starting to emerge as a powerful tool in almost every application and it is the most researched field currently.

5.2.1 Improvement in QPE using terrain height and Melting layer height

Precipitation at surface in vastly dependent on the height of the melting layer from the radar. If terrain height and melting layer heights are provided along with the vertical profiles of radar moments as inputs, the DNN model could be capable of estimating RR better. This could also make the model robust and could become flexible to be deployed in different geographical regions.

5.2.2 Multi-geography based Model

This model performs fairly poorly if used with data belonging to a different region. This is due to the different microphysical properties of clouds observed in the regions. The easiest solution might be to collect data belonging to multiple regions. But as this is essentially a bias-based issue, just data-based training would not result in great results. Some more information based on region and season might be necessary to be included along with their data.

5.2.3 Improving the dataset

The relation between reflectivity and rain rate changes depending on the geography of the region as well as depending on the season. To get a more robust estimation, more data needs to be collected and used for training the model. Moreover, mechanisms for overcoming imbalanced dataset should be implemented. A smart removal of rain cases can make the histogram of the dataset more uniform. Dataset regularization is one of the most difficult task to achieve in meteorology but this would definitely produce far better results.

5.2.4 Improving the Model

Research in Machine Learning is developing at a very fast pace. There are huge amount of model architectures that can be tried out. Moreover there are innumerable amount of activation functions, error functions and hyperparameters to try out and the design space for machine learning models themselves is tremendous. The models in this research could easily achieve better results if an identical model is trained by using continuous weighting methods during regression. Using such techniques can improve the accuracy of QPE based DNN.

5.2.5 Ensemble of Machine Learning models

Ensemble Machine Learning is a branch in machine learning that focuses on using multiple ML algorithms for estimation using a consensus mechanism. A single model is susceptible to errors but using an ensemble-based of multiple machine learning model can result in more accurate results. Every ML model has its special use case and works best in a certain setting. Using an ensemble can leverage the advantages of every ML model to give a better result.

5.3 Conclusion

This thesis establishes a basis for the use of ML models in meteorology. An idea that Deep Neural Networks could be used as an alternative to empirical equations and relations is conveyed through this work. Rainfall estimation was done using a 2-Stage DNN QPE model. This model uses classification and a regression models in conjunction to estimate the rain rate. The model was tested out on different datasets and compared with the DROPS baseline model and NEXRAD DPR product.

Bibliography

[1] Geoffrey M. Bonnin, David Kitzmiller, Dong-Jun Seo, Michael Bryan Smith, and Pedro J. Restrepo. Qpe in nws hydrologic applications : Present and future. 2007.

[2] Isztar Zawadzki. Factors affecting the precision of radar measurements of rain. *Conference on Radar Meteorology of the American Meteorological Society*, pages 251–256, 01 1984.

[3] Xuebing Yang, Qiuming Kuang, Wensheng Zhang, and Guoping Zhang. A terrain-based weighted random forests method for radar quantitative precipitation estimation. *Meteorological Applications*, 24(3):404–414, 2017.

[4] National Weather Service. Dual-pol products. https://www.weather.gov/jan/dualpolupgrade-products#:~:text=Definition%3A%20The%20Digital%20Precipitation%20Rate,occurring%20at%20the%20current%20time.

[5] Michael J Istok, M Fresch, Z Jing, S Smith, R Murnan, A Ryzhkov, J Krause, M Jain, P Schlatter, J Ferree, et al. Wsr-88d dual polarization initial operational capabilities. In *25th Conf. on Int. Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*. Amer. Meteor. Soc. Phoenix, AZ, 2009.

[6] Robert Cifelli, Venkatachalam Chandrasekar, Sanghun Lim, Patrick C Kennedy, Y Wang, and Steven A Rutledge. A new dual-polarization radar rainfall algorithm: Application in colorado precipitation events. *Journal of Atmospheric and Oceanic Technology*, 28(3):352–364, 2011.

[7] Gail Skofronick-Jackson, Walter A Petersen, Wesley Berg, Chris Kidd, Erich F Stocker, Dalia B Kirschbaum, Ramesh Kakar, Scott A Braun, George J Huffman, Toshio Iguchi, et al. The global precipitation measurement (gpm) mission for science and society. *Bulletin of the American Meteorological Society*, 98(8):1679–1695, 2017.

[8] Sounak Kumar Biswas and V Chandrasekar. Cross validation of observations from gpm dual-frequnecy precipitation radar with s-band ground radar measurents over the dallas—fort

worth region. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2085–2088. IEEE, 2017.

[9] Sounak K Biswas, Minda Le, and V Chandrasekar. Identification of snow from gpm-dpr observations and cross validation with s-band ground radar dual polarization measurements. In *2017 XXXIInd General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, pages 1–3. IEEE, 2017.

[10] Jonathan P Olson, Sounak K Biswas, V Chandrasekar, and Steven C Reising. Evaluation of the use of cubesats in atmospheric profiling. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3537–3540. IEEE, 2017.

[11] V Chandrasekar, Sounak Kumar Biswas, Minda Le, and Haonan Chen. Cross validation of raindrop size distribution retrievals from gpm dual-frequency precipitation radar using ground-based polarimetric radar. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 8335–8338. IEEE, 2018.

[12] J. S. Marshall and W. Mc K. Palmer. The distribution of raindrops with size. *Journal of Atmospheric Sciences*, 5(4):165 – 166, 1948.

[13] Radar observation of the atmosphere. l. j. battan (the university of chicago press) 1973. pp x, 324; 125 figures, 21 tables. £7·15. *Quarterly Journal of the Royal Meteorological Society*, 99(422):793–793, 1973.

[14] Merrill I. Skolnik. *Introduction to radar systems*. McGraw-Hill international editions. Electrical engineering series. McGraw-Hill, Boston, [Mass.] ;, 3rd ed. edition, 2001.

[15] T. A. Seliga and V. N. Bringi. Potential use of radar differential reflectivity measurements at orthogonal polarizations for measuring precipitation. *Journal of Applied Meteorology and Climatology*, 15(1):69 – 76, 1976.

[16] V. N. Bringi and V. Chandrasekar. *Polarimetric Doppler Weather Radar: Principles and Applications*. Cambridge University Press, 2001.

[17] D. S. Zrnić and A. Ryzhkov. Advantages of rain measurements using specific differential phase. *Journal of Atmospheric and Oceanic Technology*, 13(2):454 – 464, 1996.

[18] Evaluation of kdp estimation algorithm performance in rain using a known-truth framework. *Journal of Atmospheric and Oceanic Technology*, 38(3):587–605, March.

[19] M. Sachidananda and D. S. Zrnić. Rain rate estimates from differential polarization measurements. *Journal of Atmospheric and Oceanic Technology*, 4(4):588 – 598, 1987.

[20] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

[21] George Philipp, Dawn Song, and Jaime G. Carbonell. Gradients explode - deep networks are shallow - resnet explained. *CoRR*, abs/1712.05577, 2017.

[22] Tim Bellerby, Martin Todd, Dom Kniveton, and Chris Kidd. Rainfall estimation from a combination of trmm precipitation radar and goes multispectral satellite imagery through the use of an artificial neural network. *Journal of applied Meteorology*, 39(12):2115–2128, 2000.

[23] Haonan Chen, V Chandrasekar, and Renzo Bechini. An improved dual-polarization radar rainfall algorithm (drops2. 0): Application in nasa ifloods field campaign. *Journal of Hydrometeorology*, 18(4):917–937, 2017.

[24] Sounak Kumar Biswas, Robert Cifelli, and V Chandrasekar. Improving quantitative precipitation estimation by x-band dual-polarization radars in complex terrain over the bay area in california, usa. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 5411–5414. IEEE, 2020.

[25] Yingzhao Ma, V Chandrasekar, and Sounak K Biswas. A bayesian correction approach for improving dual-frequency precipitation radar rainfall rate estimates. *Journal of the Meteorological Society of Japan. Ser. II*, 2020.

[26] Jian Zhang, Kenneth Howard, Carrie Langston, Brian Kaney, Youcun Qi, Lin Tang, Heather Grams, Yadong Wang, Stephen Cocks, Steven Martinaitis, Ami Arthur, Karen Cooper, Jeff Brogden, and David Kitzmiller. Multi-radar multi-sensor (mrms) quantitative precipitation estimation: Initial operating capabilities. *Bulletin of the American Meteorological Society*, 97(4):621 – 638, 2016.

[27] K. E. Kelleher. Project craft: A real-time delivery system for nexrad level ii data via the internet. *Bull. Amer. Meteor. Soc.*, 88:1045–1057, 2007.

[28] M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadtler. Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200097, 2021.

[29] Rongrui Xiao, V. Chandrasekar, H. Liu, and E. Gorgucci. Detection of rain/no rain condition on ground from radar data using a kohonen neural network. In *IGARSS '98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No.98CH36174)*, volume 1, pages 159–161 vol.1, 1998.

[30] H. Liu, V. Chandrasekar, and Eugenio Gorgucci. Detection of rain/no rain condition on the ground based on radar observations. *Geoscience and Remote Sensing, IEEE Transactions on*, 39:696 – 699, 04 2001.

[31] Wanyu Li and V. Chandrasekar. Rainfall estimation from vertical profiles of reflectivity using neural networks. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 6, pages 3477–3479 vol.6, 2002.

[32] Amin Alqudah, Yanting Wang, and V. Chandrasekar. Improving rainfall estimation from ground based radar measurements using neural networks. In *2009 IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages I–200–I–203, 2009.

[33] Jing Feng, Dinglian Yuan, and Ai Xia Zhou. A rainfall estimation method based on rbfnn. *2016 IEEE Region 10 Conference (TENCON)*, pages 2856–2859, 2016.

[34] Kyuhee Shin, Joon Jin Song, Wonbae Bang, and GyuWon Lee. Quantitative precipitation estimates using machine learning approaches with operational dual-polarization radar data. *Remote Sensing*, 13(4), 2021.

[35] Andrew Kusiak, Xiupeng Wei, Anoop Prakash Verma, and Evan Roz. Modeling and prediction of rainfall using radar reflectivity data: A data-mining approach. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4):2337–2342, 2013.

[36] Qiangqiang Yuan, Huanfeng Shen, Tongwen Li, Zhiwei Li, Shuwen Li, Yun Jiang, Hongzhang Xu, Weiwei Tan, Qianqian Yang, Jiwen Wang, Jianhao Gao, and Liangpei Zhang. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, 241:111716, 2020.

[37] Yen-Ming Chiang, Fi-John Chang, Ben Jong-Dao Jou, and Pin-Fang Lin. Dynamic ann for precipitation estimation and forecasting from radar observations. *Journal of Hydrology*, 334(1):250–261, 2007.

[38] Ting-Shuo Yo, Shih-Hao Su, Jung-Lien Chu, Chiao-Wei Chang, and Hung-Chi Kuo. A deep learning approach to radar-based qpe. *Earth and Space Science*, 8(3):e2020EA001340, 2021. e2020EA001340 2020EA001340.

[39] Haonan Chen, V. Chandrasekar, and Robert Cifelli. A deep learning approach to dual-polarization radar rainfall estimation. pages 1–2, 03 2019.

[40] Haonan Chen, V. Chandrasekar, Robert Cifelli, and Pingping Xie. A machine learning system for precipitation estimation using satellite and ground radar network observations. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):982–994, 2020.

[41] Wei Tian, Lei Yi, Wei Liu, Wei Huang, Guangyi Ma, and Yonghong Zhang. Ground radar precipitation estimation with deep learning approaches in meteorological private cloud. *J. Cloud Comput.*, 9(1), apr 2020.

[42] Radar Operations Center NOAA National Weather Service. Noaa next generation radar (nexrad) level ii base data, 1991.

[43] David B. Wolff, D. A. Marks, E. Amitai, D. S. Silberstein, B. L. Fisher, A. Tokay, J. Wang, and J. L. Pippitt. Ground validation for the tropical rainfall measuring mission (trmm). *Journal of Atmospheric and Oceanic Technology*, 22(4):365 – 380, 2005.

[44] National Centers for Environmental Information. Nexrad data format. https://www.weather.gov/jan/dualpolupgrade-products#:~:text=Definition%3A%20The%20Digital%20Precipitation%20Rate,occurring%20at%20the%20current%20time.

[45] Yanting Wang and V. Chandrasekar. Algorithm for estimation of the specific differential phase. *Journal of Atmospheric and Oceanic Technology*, 26(12):2565 – 2578, 2009.

[46] Tulipa Silva, Paulo Siqueira, Cesar Beneti, Maiko Buzzi, and Leonardo Calvetti. Identification of severe weather event with 3d meteorological radar and mlp. 01 2017.

[47] Jonathan Helmus and Scott Collis. The python arm radar toolkit (py-art), a library for working with weather radar data in the python programming language. *Journal of Open Research Software*, 4, 07 2016.

[48] Yutaka Sasaki. The truth of the f-measure. *Teach Tutor Mater*, 01 2007.

[49] Christopher K. I. Williams. The Effect of Class Imbalance on Precision-Recall Curves. *Neural Computation*, 33(4):853–857, 04 2021.

[50] David Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness correlation. *Mach. Learn. Technol.*, 2, 01 2008.

[51] Claude Sammut and Geoffrey I. Webb, editors. *F1-Measure*, pages 397–397. Springer US, Boston, MA, 2010.

[52] Namgil Lee, Heejung Yang, and Hojin Yoo. A surrogate loss function for optimization of f_β score in binary classification with imbalanced data. *CoRR*, abs/2104.01459, 2021.

[53] Kai Ming Ting. *Precision and Recall*, pages 781–781. Springer US, Boston, MA, 2010.

Appendix A

Model Architectures of Rain/No-Rain Classification DNNs

This Appendix gives detailed description of all the layers and the structure of Classification Neural Models used in this thesis.

A.0.1 RC(Z) Rain Classification Model

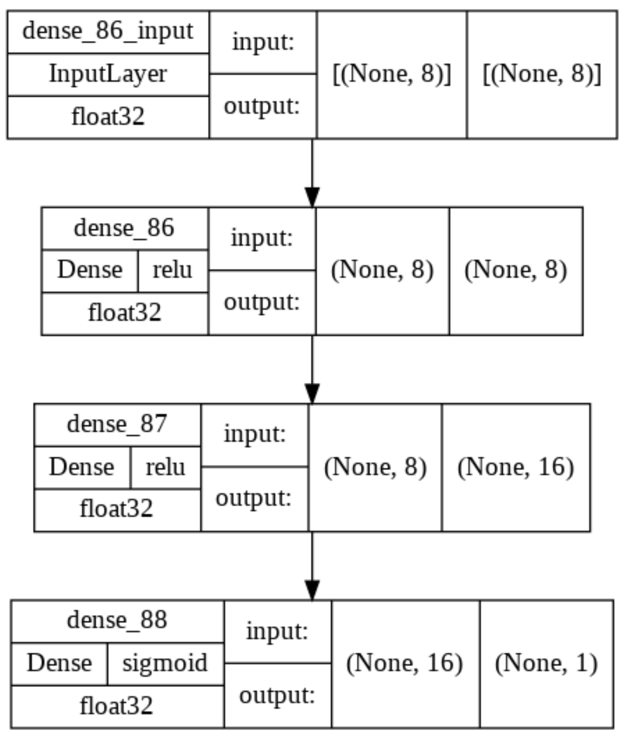


Figure A.1: RC(Z) Deep Neural Network Architectural Structure

A.0.2 RC(Z-Z_{dr}) Rain Classification Model

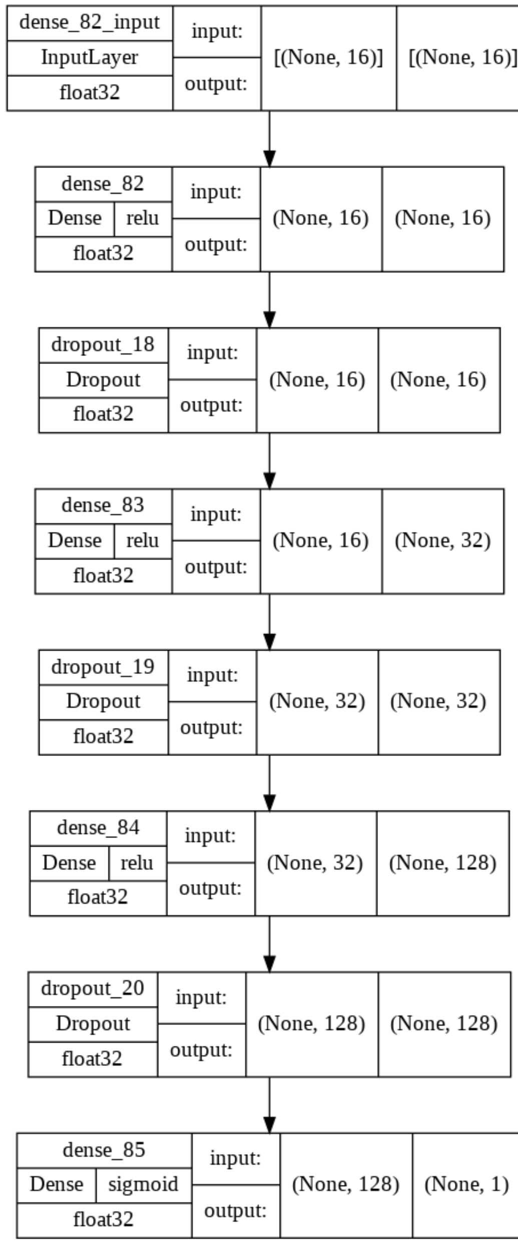


Figure A.2: RC(Z-Z_{dr}) Deep Neural Network Architectural Structure

A.0.3 RC(Z-Z_{dr}-K_{dp}) Rain Classification Model

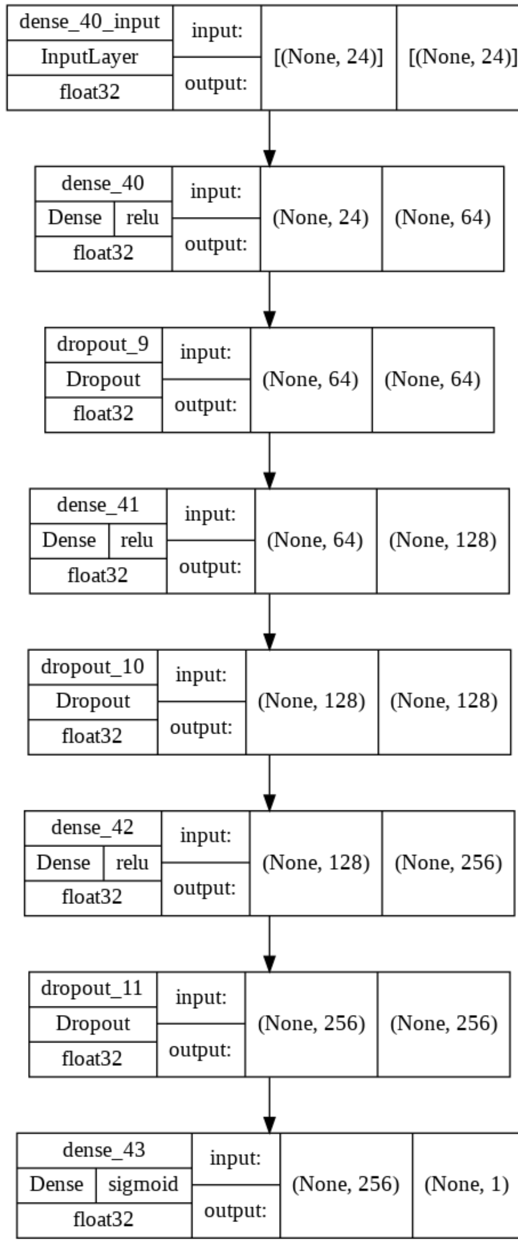


Figure A.3: RC(Z-Z_{dr}-K_{dp}) Deep Neural Network Architectural Structure

Appendix B

Model Architectures of Rain Rate estimation DNNs

This Appendix gives detailed description of all the layers and the structure of Rain Rate estimation Neural Network Models used in this thesis.

B.0.1 R(Z) Rain Rate Estimation Model

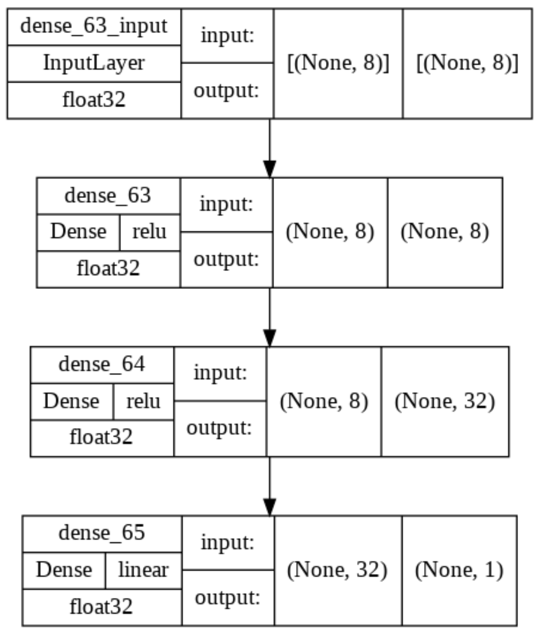


Figure B.1: R(Z) Deep Neural Network Architectural Structure

B.0.2 R(Z) 2-Stage DNN Rain Rate Estimation Model

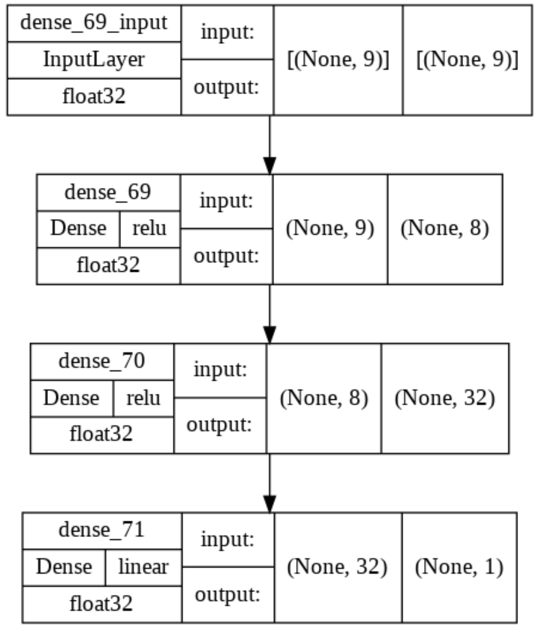


Figure B.2: R(Z) 2-Stage Deep Neural Network Architectural Structure

B.0.3 R(Z-Z_{dr}) 2-Stage DNN Rain Rate Estimation Model

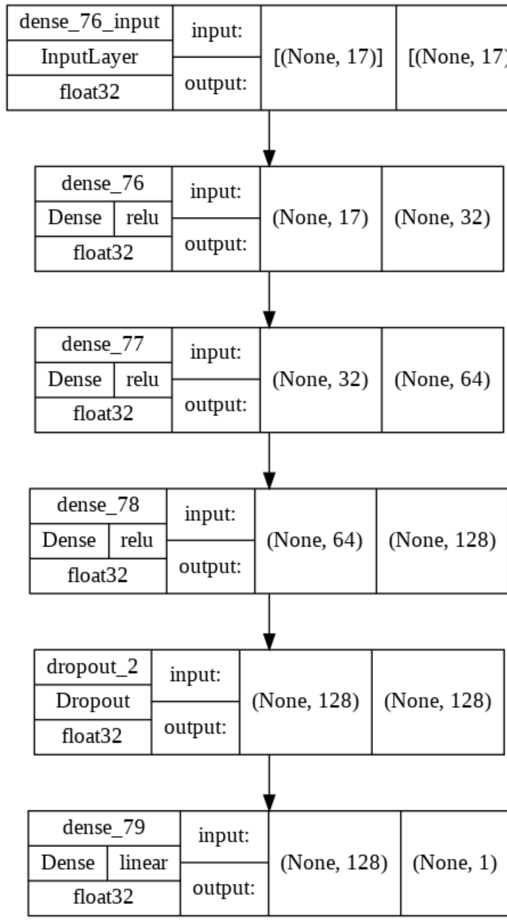


Figure B.3: R(Z-Z_{dr}) 2-Stage Deep Neural Network Architectural Structure

B.0.4 R(Z-Z_{dr}-K_{dp}) 2-Stage DNN Rain Rate Estimation Model

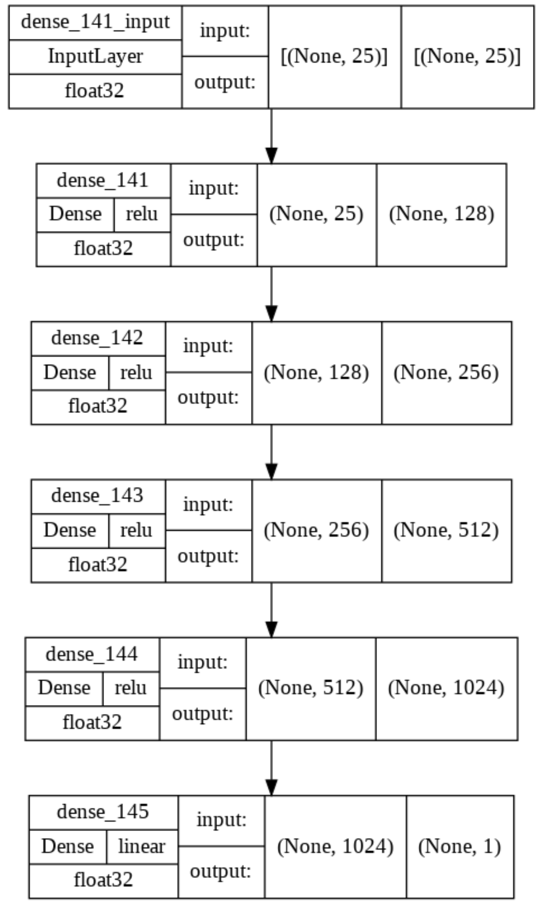


Figure B.4: R(Z-Z_{dr}-K_{dp}) 2-Stage Deep Neural Network Architectural Structure