

# Prediction of rain cell growth and decay using numerical weather prediction models

Youness Kabbouch

Technische Universiteit Delft





# Prediction of rain cell growth and decay using numerical weather prediction models

by

**Youness Kabbouch**

in partial fulfillment of the requirements for the degree of

**Master of Science**  
in Applied Physics

at the Delft University of Technology,  
to be defended publicly on Friday July 10, 2020 at 02:00 PM.

Student number: 4621808  
Supervisor: Dr. M.A. Schleiss  
Thesis committee: Prof. dr. ir. H.W.J. Russchenberg, TU Delft  
Dr. ir. J.A.E ten Veldhuis, TU Delft  
Dr. M.A. Schleiss, TU Delft  
Dr. Y. Dufournet, SkyEcho  
Dr. A.C.P. Oude Nijhuis, SkyEcho



# Acknowledgements

Obtaining my masters degree would not have been feasible without the support and guidance of many people. This thesis is a result of their help. I, therefore, would like to take this opportunity to express my gratitude and show my appreciation towards them. I would like to thank the following persons who made this possible.

**Dr. M.A. (Marc) Schleiss** has put a lot of his energy and time in guiding me through this thesis. I am very grateful for his insightful comments and suggestions and have learned a lot from him.

**SkyEcho** for making this thesis possible, giving me a place to do this thesis. I would like to especially thank **Dr. Y. (Yann) Dufournet** and **Dr. A.C.P. (Albert) Oude Nijhuis** from SkyEcho for sharing their knowledge and ideas. They have been very helpful and supportive. I wish them and SkyEcho good luck with the Rijnmond radar project and other future projects.

**The rest of the assessment committee** for taking their time to read and evaluate my thesis. Their ideas and feedback are appreciated and was of great value for the outcome of the thesis.

**Family and friends** for being supportive throughout the whole study. They were patient, listened and gave me advice if needed. My gratitude and appreciation goes out to them.

Before you lies my thesis and I hope you will read it with pleasure.

*Youness Kabbouch  
Rotterdam, July 2020*

This thesis is dedicated to the memory of **A. Kabbouch** and **M. Aazouz**, my grandfathers.



# Abstract

Radar rainfall nowcasting stands for the prediction of rainfall amounts and intensities over the next 6 hours by means of statistical extrapolation of radar measurements. It is the principal ingredient for modern flood forecasting and early warning systems. Radar forecasts are generated by identifying and tracking rainfall cells in radar images and extrapolating them along the main direction of motion based on the assumption of Lagrangian persistence. The latter means that the rainfall cells do not physically evolve over time. Therefore the key to improving predictions is to anticipate the growth and decay of rainfall cells during the next few hours. Studies have shown that estimating growth and decay based on past radar images is nearly impossible. Therefore, in this thesis, a new approach for predicting growth and decay based on physical guidance from a numerical weather prediction model is investigated. The idea is that numerical weather prediction models are better at anticipating changes in the atmosphere than radar and thus, should also contain information about the growth and decay of rain cells. To test this hypothesis, two simple machine learning models have been trained to learn the relationship between the observed growth and decay in radar images and the output parameters from the Dutch mesoscale numerical weather prediction model HARMONIE. The models were trained on data in the summer of 2019 around the city of Rotterdam for two area sizes,  $20 \times 20 \text{ km}^2$  and  $60 \times 60 \text{ km}^2$ . The first model is static and assumes that all predictions from HARMONIE are correct. Unfortunately, this turns out to be too optimistic as HARMONIE often places the rain cells at the wrong locations with the wrong intensities and trends. As a result, no meaningful relationship between HARMONIE outputs and growth/decay in radar could be learned. The dynamic model overcomes this issue with the help of an additional classifier. The classifier predicts whether the information from HARMONIE can be trusted or not. Then, a regression model predicts the magnitude of growth and decay (only for the trusted cases).

Out of 308 analysed cases, the classifier labelled 237 HARMONIE predictions as being untrustworthy, thereby removing a lot of bad cases. However, 98 of these were false negatives, meaning that they could have been used to predict growth and decay. On the other hand, only 71 cases were labelled as containing useful information, from which 27 cases were false positives. Despite these errors, the dynamic model was able to improve the root mean square error on the predicted growth/decay by 27% for these 71 cases. Also, the correlation between the predicted and actual growth/decay increased from near zero to 0.488. This is encouraging and clearly highlights the potential of this approach. Still, some important challenges remain. In particular, results show that the dynamic model often underestimates the magnitude of growth and decay. Also, when the entire validation data set is considered, including all unusable cases, the improvement with respect to the static model is only 6%. This is mainly caused by the poor performance of HARMONIE and the large number of unusable cases. However, performance could be improved further by building a better classifier capable of identifying all good cases with a low number of false negatives.



# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Abbreviations and Symbols</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 State of the art . . . . .	2
1.2 Problem definition . . . . .	4
1.3 Approach . . . . .	4
<b>2 Data</b>	<b>5</b>
2.1 Description of the radar data. . . . .	5
2.1.1 Radar observations. . . . .	5
2.1.2 Radar forecasts . . . . .	5
2.1.3 Conversion data to rain intensity . . . . .	6
2.2 Description of the NWP . . . . .	7
2.3 Data availability and quality . . . . .	7
2.3.1 Radar. . . . .	7
2.3.2 HARMONIE . . . . .	8
<b>3 Methods</b>	<b>11</b>
3.1 Study area. . . . .	11
3.2 Definition of growth and decay. . . . .	11
3.2.1 Tracking of precipitation parcel . . . . .	11
3.2.2 Formulation of growth and decay . . . . .	13
3.3 General approach . . . . .	14
3.3.1 Sliding window method . . . . .	15
3.3.2 Shift correction method . . . . .	15
3.4 Multi-layer perceptron. . . . .	16
3.4.1 Backpropagation . . . . .	17
3.4.2 Validation of the neural network. . . . .	18
3.5 Event selection . . . . .	18
3.6 Construction of the training and validation data set . . . . .	19
<b>4 Assessment of radar and HARMONIE forecasts</b>	<b>21</b>
4.1 Performance for selected events . . . . .	21
4.1.1 Event 1 - Growth . . . . .	21
4.1.2 Event 2 - No growth or decay. . . . .	23
4.1.3 Event 3 - Decay. . . . .	23
4.2 General performance. . . . .	27
4.2.1 Performance of radar nowcast with lead time . . . . .	28
4.2.2 Limitations of HARMONIE . . . . .	29
4.3 Correlation of HARMONIE features with precipitation rate . . . . .	30
4.4 Validation of the backtracking algorithm. . . . .	31
4.5 Performance regarding growth and decay . . . . .	32
4.5.1 Lagrangian versus Eulerian framework. . . . .	33
4.5.2 Comparison between radar and HARMONIE. . . . .	34

---

<b>5</b>	<b>Static model</b>	<b>35</b>
5.1	Model formulation . . . . .	35
5.2	Results for the static model. . . . .	36
<b>6</b>	<b>Dynamic model</b>	<b>41</b>
6.1	Input classifier . . . . .	41
6.2	Results for the dynamic model. . . . .	42
6.2.1	Input classifier . . . . .	42
6.2.2	Performance of the dynamic model . . . . .	43
6.2.3	Performance after applying the shift correction . . . . .	45
<b>7</b>	<b>Conclusion and recommendations</b>	<b>47</b>
7.1	Conclusion . . . . .	47
7.2	Recommendations . . . . .	49
<b>A</b>	<b>Accessing radar data with python</b>	<b>51</b>
<b>B</b>	<b>Accessing HARMONIE data with python</b>	<b>53</b>
	<b>Bibliography</b>	<b>55</b>

# List of Figures

2.1	Observed rain intensity on 2019-05-09 18:00 after conversion to precipitation rate $R$ (mm/h). . . . .	6
2.2	On the left the observed rain intensity by the radar and on the right the predicted rain intensity by the HARMONIE model for 2019-05-08 15:00. The HARMONIE model was run based on the latest data assimilated at 12 o'clock and the forecast time is thus 3 hours. . . . .	7
2.3	Precipitation rate observed on 2019-07-26 03:35 over the Netherlands. Artifacts in this observation are noticeable around the coast near Maasvlakte 2. . . . .	9
2.4	On the left the observed rain intensity by the radar and on the right the predicted rain intensity by the HARMONIE model for 2019-06-02 18:00. . . . .	9
3.1	The two areas ( $A^{20}$ and $A^{60}$ ) used for the prediction of growth and decay of rain cells. . . . .	12
3.2	Displacement terms $\alpha$ and $\beta$ used in the backtracking of rain cells. The solid line and dashed line represent the trajectory computed by the constant-vector scheme and semi-Lagrangian scheme respectively. . . . .	13
3.3	On the left: an example of a three-layer perceptron. On the right: a visual representation of how the output of a node is composed. . . . .	17
3.4	Process of selecting the events . . . . .	19
4.1	Time series of a rain cell passing the buffer zone $A^{60}$ on 2019-06-10 with intervals of 20 minutes. . . . .	22
4.2	Time series of the nowcast images on a rain cell passing the buffer zone $A^{60}$ on 2019-06-10 with intervals of 20 minutes. The lead time is one hour. . . . .	22
4.3	Time series of the HARMONIE forecasts on a rain cell passing the buffer zone $A^{60}$ on 2019-06-10. . . . .	23
4.4	Time series of a rain cell passing the buffer zone $A^{60}$ on 2019-07-27 with intervals of 20 minutes. . . . .	24
4.5	Time series of the nowcast images on a rain cell passing the buffer zone $A^{60}$ on 2019-07-27 with intervals of 20 minutes. The lead time is one hour. . . . .	24
4.6	Time series of the HARMONIE forecasts on a rain cell passing the buffer zone $A^{60}$ on 2019-07-27. . . . .	25
4.7	Time series of a rain cell passing the buffer zone $A^{60}$ on 2019-08-19 with intervals of 20 minutes. . . . .	25
4.8	Time series of the nowcast images on a rain cell passing the buffer zone $A^{60}$ on 2019-08-19 with intervals of 20 minutes. The lead time is one hour. . . . .	26
4.9	Time series of the HARMONIE forecasts on a rain cell passing the buffer zone $A^{60}$ on 2019-08-19. . . . .	26
4.10	From left to right, the observed precipitation rate, predicted precipitation rate from the radar nowcast with 1 hour lead time and predicted precipitation rate from HARMONIE respectively. . . . .	27
4.11	The observed average precipitation rate and the predicted average precipitation rate for different lead times from the radar nowcasts over the area $A^{20}$ . . . . .	28
4.12	The mean squared error between the KNMI nowcasts and observed precipitation rate computed over the entire data set on 5 min temporal resolution. The MSE in the plot on the left is over the whole data set and the MSE in the plot on the right is computed over points with observed average precipitation rate above 1 mm/h. As expected, the error increases with the lead time. . . . .	28

4.13	The correlation coefficient between the radar nowcasts and observed precipitation rate over the whole data set. As the lead time increases the error gets larger and the correlation decreases. . . . .	29
4.14	The observed average precipitation rate and the predicted average precipitation rate from the HARMONIE model over the areas $A^{20}$ and $A^{60}$ . . . . .	29
4.15	Polar histogram of the estimated flow direction. The flow directions are estimated using the optical flow method called Lucas–Kanade. . . . .	31
4.16	Left: Histogram of the growth and decay for the Eulerian and Lagrangian frameworks. Right: Scatterplot of the growth and decay in the Lagrangian framework against the growth and decay in the Eulerian framework. The growth and decay shown is averaged over area $A^{20}$ . . . . .	33
4.17	Scatterplot between the radar growth and decay and HARMONIE derived growth and decay for area $A^{20}$ . . . . .	34
5.1	The static neural network model . . . . .	36
5.2	The MSE of the training and validation data set after each epoch, i.e. after each update of the weights. The model was trained to estimate the average growth and decay for area $A^{60}$ . . . . .	36
5.3	Upper left: the predicted growth and decay against the observed growth and decay. Upper right: scatterplot of growth and decay against the growth and decay of HARMONIE. Lower left: scatterplot of growth and decay against the change in medium cloud cover. Lower right: scatterplot of growth and decay against the change in relative humidity. The growth and decay values are in logarithmic reflectivity and all points represent the average over $A^{60}$ . . . . .	38
5.4	The average growth and decay over $A^{60}$ of a rain event on 2019-07-26. The rain event is not predicted at the right time. Therefore, the growth and decay derived from HARMONIE does not match with the radar growth and decay. . . . .	39
5.5	The average growth and decay over $A^{60}$ of a rain event on 2019-08-01. The rain event is not predicted at the right place. Therefore, the growth and decay derived from HARMONIE does not match with the radar growth and decay. . . . .	39
6.1	Flowchart of the static and dynamic model. . . . .	42
6.2	Two scatterplots of the growth and decay against the normalized growth and decay of HARMONIE for area $A^{60}$ . The left scatterplot shows the points which were used to train the input classifier. The right scatterplot shows which points are predicted to be trusted or not by the input classifier. . . . .	43
6.3	Scatterplot showing the predicted growth and decay $\Delta\hat{z}_{t-60,t}$ against the observed growth and decay $\Delta z_{t-60,t}$ for area $A^{60}$ . . . . .	45
6.4	Scatterplots showing the HARMONIE growth and decay $\Delta inr$ against the observed growth and decay $\Delta z_{t-60,t}$ . Each scatterplot is corrected for a shift which is based on $L$ previous images. No shift correction is applied on the first scatterplot. . . . .	46

# List of Tables

2.1	Near surface parameters found in the data set from the KNMI HARMONIE model. The code values refers to the parameter ID in the GRIB Parameter Database. The level values show the height above ground, except for PMSL (height above sea) and BRT (Nominal top of atmosphere) . . . . .	8
4.1	The general performance of the radar nowcasts and HARMONIE. The lead time of the radar nowcasts is one hour and is computed on the entire data set on an hourly resolution.	27
4.2	Pearson’s correlation coefficient between the average precipitation rates $\bar{p}^{20}$ , $\bar{p}^{60}$ and the HARMONIE parameters averaged over area $A^{20}$ , sorted from high positive correlation to high negative correlation. . . . .	30
4.3	Pearson’s correlation coefficient between the average precipitation rates $\bar{p}^{20}$ , $\bar{p}^{60}$ and the HARMONIE parameters averaged over area $A^{60}$ , sorted from high positive correlation to high negative correlation. . . . .	31
4.4	Classification of the growth and decay in the area $A^{60}$ after applying the two different motion fields. . . . .	32
4.5	Performance of the KNMI radar nowcasts and HARMONIE in estimating growth and decay for 1h lead time. . . . .	33
5.1	Performance of the static model on the validation data set in estimating growth and decay for 1h lead time. . . . .	37
6.1	Confusion matrix of the input classifier for area $A^{60}$ , each row represents the cases in the actual class and each column represents the predicted class for these cases. . . .	43
6.2	Performance of the dynamic model on the validation data set in estimating growth and decay for 1h lead time. . . . .	44
6.3	Performance of the dynamic model in estimating growth and decay for 1h lead time. The performance metrics are computed only over the cases where the input classifier has predicted that HARMONIE is able to predict growth and decay. . . . .	44



# List of Abbreviations and Symbols

NWP	Numerical weather prediction
KNMI	Royal Netherlands Meteorological Institute
HARMONIE	HIRLAM ALADIN Research on Mesoscale Operational NWP in Euromed
MLP	Multi-layer perceptron
MSE	Mean squared error
MBE	Mean bias error
$A^l$	Area of a rectangle with side lengths of $l$ km
$\bar{p}^l$	Average precipitation rate over the area $A^l$
$\alpha$	The distance a pixel has traveled in the x-direction
$\beta$	The distance a pixel has traveled in the y-direction
$u(x, y)$	The velocity of a pixel in x-direction at location $(x, y)$
$v(x, y)$	The velocity of a pixel in y-direction at location $(x, y)$
$(x_{target}, y_{target})$	Location of a pixel at time $t$
$(x_{origin}, y_{origin})$	Location of a pixel at time $t - \tau$
$\bar{z}(t)$	Average logarithmic reflectivity
$\Delta z_{t-\tau, t}$	Change in average logarithmic reflectivity between time $t$ and $t - \tau$ , i.e. the average growth and decay
$\Delta \hat{z}_{t-\tau, t}$	Estimated change in average logarithmic reflectivity between time $t$ and $t - \tau$ , i.e. the estimated average growth and decay
$\rho$	Pearson's correlation coefficient
$\Delta inr$	The average change in precipitation rate after normalization from HARMONIE between time $t$ and $t - \tau$
$\Delta MCC$	The average change in medium cloud cover from HARMONIE between time $t$ and $t - \tau$
$\Delta RH$	The average change in relative humidity from HARMONIE between time $t$ and $t - \tau$



# 1

## Introduction

Nowcasting is the prediction of a certain phenomena over a short period of time. Precipitation nowcasting focuses on predicting rainfall mostly up to two hours. These nowcasts have increasingly become more accessible to the mass. Businesses, like airports, which are sensitive to changes in weather conditions require accurate predictions of precipitation. Also due to the urbanisation and climate change, there is an increasing demand in nowcasting for heavy rain and floods [1].

The conventional approach of nowcasting predicts precipitation based on the extrapolation of precipitation areas derived from radar maps. The extrapolation is done in various ways, but the fundamental principle is first to infer a motion field from a series of radar images and then the precipitation areas are extrapolated according to the estimated motion field. This approach is built on the assumption that the total intensity does not change over time, also called Lagrangian persistence [2].

The advection of rain cells usually accounts for a significant part of the variations in temporal domain for the first tens of minutes. However, one of the shortcomings of the conventional approach is the inability to predict new rain cells, because the nowcast depends only on what is observed at a certain time [3, 4]. Also growth or decay of a rain cell, both in aerial size and intensity, is not accounted for in this approach. Neglecting this growth and decay of rain cells causes the nowcasts lagging behind or ahead with respect to the real precipitation field. This time lag gets bigger when the lead times increases. This time lag is caused due to the size of the rain cell getting bigger or smaller and thus reaches an area faster or slower than predicted.

The disadvantages of extrapolating rain cells could be solved by approaching the nowcasting with a machine learning perspective. Several studies have already shown that machine learning techniques are able to outperform the traditional nowcasting techniques, by capturing correlations in both the spatial and temporal domain and recognizing patterns in successive radar images [5–7]. However, the use of machine learning in precipitation nowcasting is still in an early stage of development [6].

The predicted rainfall contains errors mainly caused by growth and decay of rain cells [8]. The problem lies in the chaotic nature of rainfall caused by complex and dynamical processes. Numerical weather predictions (NWP) use a variety of information, like air pressure, wind, temperature and humidity, to predict how the weather will change and/or look in the future. Because numerical weather predictions are based on physical models, they better account for the dynamical behaviour of rain cells in contrast to radar nowcasts which do not account for any change in the rainfall field. The downside of using these predictions is that the spatial resolution is relatively coarse compared to the radar nowcasts and the temporal resolution is far lower. In this thesis, we investigate whether the information from numerical weather prediction models could be combined with the conventional extrapolation approach using a machine learning technique to account for the growth and decay of rain cells.

## 1.1. State of the art

Techniques for extrapolating radar echos can be divided into two main groups. The first group tries to fit a sequence of images and extrapolates the image based on the derived motion field. This is done by either maximizing the correlation between the images [9] or by solving the optical flow equations [10]. Different techniques exist, but the underlying basis remains the same. It performs well for large scale rain cells which do not move fast and also do not grow as fast as convective rain cells. As this extrapolates the whole area, it is widely used in nowcasting systems, such as Nowcasting and Initialization for Modeling Using Regional Observation Data System (NIMROD) [11], Auto-Nowcast System (ANC) [12], Continuity of TREC Vectors (COTREC) [13], McGill Algorithm for Precipitation Nowcasting by Lagrangian Extrapolation (MAPLE) [14], Short-Term Ensemble Prediction System (STEPS) [15] and Short-range Warning of Intense Rainstorms in Localized Systems (SWIRLS) [16]. The second group is mostly used in forecasting storm cells. It detects and tracks the movement of individual rain cells. Depending on the used technique certain parameters, like the centroid or borders, are tracked and extrapolated [17, 18]. For example, Thunderstorm Identification, Tracking, Analysis and Nowcasting (TITAN) [18] identifies individual storm cells.

There are many uncertainties in the process of extrapolating radar echos. The first one are the uncertainties in the measurement. The main sources of error in the measurement can be found in signal attenuation, clutter and anomalous propagation, beam blockage, variability of the Z-R relation [19]. The model may also contain errors in the derived motion field [20]. The second source of uncertainties is related to the dynamics of the rain cells and atmosphere and is linked to the assumption of Lagrangian persistence. Most of the predictive skill in radar nowcasts therefore tends to be lost after 60-180 min [21]. In many cases, the intensity of the rain field will vary throughout time and also the size of the individual rain cells will change over time. As aforementioned, the biggest error source in predicting rainfall after tens of minutes is caused by this change in intensity and size (growth and decay) [8]. Note that this also depends on the type of rain cell. Convective rain cells tend to have a higher variability than stratiform rain cells and thus the error is much larger for convective cases. A recent study [22] attempted to determine the predictability of growth and decay of precipitation at different scales. According to the study, the predictability of both rainfall and growth and decay increases when eliminating small scale rain cells. Small scale rain cells (convective cells) change very rapidly and their lifetime is just a few tens of minutes. And hence the predictability of these small scale rain cells and growth and decay is considerably low [22]. Larger cells, however, can persist for many hours and remain predictable for much longer.

The simplest form of nowcasting is to just extrapolate the rainfall fields and not take the growth and decay into account. One might also think that trends in intensity and echo size in radar images could be used for the estimation of the growth and decay. Tsonis and Austin were the first ones to study this. They concluded that including these trends do not yield any significant improvements. Their method was unable to predict when cells would start decay and in general, the variability of the rain cells dominated the weakly defined trend [23]. The study by Wilson came to the same conclusion [8]. Both studies tracked individual cells and then estimated the trends. However, a different study showed that trending could provide some improvements [24]. This study used a radius of influence around each pixel and then estimated the trends.

All of the nowcasting systems above were based on radar images. However, there are nowcasting systems which use more complex algorithms to improve the predictions of rainfall fields. These are called expert systems and make decisions based on different models, thresholds and so on [25]. Thunderstorm Identification, Tracking, Analysis and Nowcasting (TITAN) [18] uses thresholds to identify precipitation areas and then track these so-called objects. Other systems also try to incorporate some of the growth and decay by using boundary layer convergence line and fuzzy logic. The Auto-Nowcast System [12] produces for example a convective likelihood field which is obtained from fuzzy logic rules. These rules are based on systematic and persistent changes in the rainfall fields.

NWP models uses physical atmospheric models to simulate and predict the conditions of the weather. Because of this, the NWP models are very computational expensive. The models require information of the initial state in order to produce a good prediction. This is done using data assimilation in the

NWP models. There is a variety of data assimilation methods, like Three-Dimensional Variational Data Assimilation (3DVAR) [26] and Four-dimensional variational data assimilation (4DVAR) [27]. The most commonly used method is 3DVAR, because of its computational efficiency [28, 29]. These methods try to produce the best initial state of the model using the given information.

As the NWP models move to higher resolutions, it will require the same high resolution observation data for initialisation. Radar networks can provide these data [30]. The assimilation of high resolution observation data benefits the prediction of convective precipitation [31, 32]. Convective precipitation are found on a smaller scale and by having more dense observation these rain cells are better described in the initial state of the model [33]. A recent development is the assimilation of radar data into NWP. Studies have tried assimilating radar data into NWP and concluded that this improves the forecasts of convective rain cells. Assimilation of the radar improved the mesoscale dynamic field in terms of better location and structure prediction and also improves the predictions of convective rain cells [31, 33, 34]. However, even after assimilation the prediction could still be off in time and position [31]. The assimilation of radar data into NWP can be done in various ways, but all have the same issue that the data cannot be directly used. Precipitation rate is computed from other parameters and the observed precipitation rates can therefore not be directly assimilated into an NWP. However, information can be inferred from the radar data about other variables, like moisture and temperature. This information can be used to adjust the variables of the NWP so that it fits more with the radar data [35]. Multiple methods for radar data assimilation are described in the review paper of Gustafsson et al [32]. This paper concluded also that the more advanced methods, like 4DVar, yield more improvements over the simpler methods.

The disadvantage of assimilating radar data is that radar observations contain noise from artifacts, clutter and radio interference. NWP does not contain any noise and by assimilating radar into NWP, noise can be introduced even though clutter removal techniques are applied beforehand. Another issue for radar data assimilation is the large difference in spatial and temporal resolution between the radar data and NWP [32]. It is therefore necessary to develop methods which can handle these discrepancies.

Attempts have been made to use NWP models for nowcasting purposes. A lot of progress have been made but there are also challenges which has not been overcome yet. Recent development is the blending of radar extrapolation techniques with NWP. This blending is simply a weighted average between the extrapolation forecast and NWP and allows therefore for a seamless 0-6 hour forecast. Resulting in a better forecast at higher lead times (2-6 hours), as NWP produces better forecasts at higher lead times. NIMROD uses this blending to allow for longer nowcasts [11]. The blending gives a certain weight to the extrapolation and NWP models depending on the lead time. For the first hour, the extrapolation of the rainfall field is given full weight. This weight gradually decreases over lead time and the weight for NWP increases over lead time. The blending techniques requires a high resolution (<4 km) convection-permitting NWP model [25]. These NWP models can reproduce the initiation and evolution of small scale convective rain cells. They however lack in accuracy in both spatial and temporal domain and are therefore not suited for nowcasting purposes [36].

## 1.2. Problem definition

The goal of this thesis is to improve radar nowcasts by estimating growth and decay of rain cells with the help of NWP models. NWP models contain some information about the future behaviour of rain cells. However, NWPs are limited in both spatial and temporal resolution and also lack accuracy compared to radar. The information from these models can still be helpful though, because NWPs incorporates physics. Using machine learning techniques, the relation between the NWP prediction and the true rainfall field as seen by radar can be learned. The research question is therefore formulated as follows:

*Is it possible to predict the growth and decay of rain cells by adding information obtained from numerical weather prediction models in radar nowcasts using machine learning?*

In order to answer this question, multiple sub-questions have been defined:

1. How accurate are current radar nowcasts and NWP predictions, for example the KNMI nowcasts?
2. How can we use machine learning to extract useful information about growth and decay from the numerical weather forecasts?
3. What is the gain in performance when the NWP predictions of growth and decay are combined with radar?

## 1.3. Approach

To answer the research questions above, information about rain cell growth and decay from the Dutch national C-band radar network and the numerical weather prediction model HARMONIE will be used. More specifically, the following tasks have been defined:

1. Assess the errors due to growth and decay in Lagrangian radar nowcasts.
2. Train a static model for predicting average growth and decay in radar images based on different features extracted from the HARMONIE weather prediction model. This model assumes that all the predictions from the numerical weather prediction model HARMONIE are good and can be trusted.
3. Train an adaptive, dynamical model for predicting growth and decay based on the most recent performances of the numerical weather prediction model HARMONIE over a moving window. The dynamic model uses only HARMONIE predictions which it trusts to predict growth and decay. The disadvantage is that the model is not always able to predict growth and decay, but could perform better than the static model.
4. Quantify the performance of each model and determine which combination of features and approach yields the most promising results.
5. Propose ideas for future improvements.

# 2

## Data

In this chapter the two data sets are described. It is important to know how the data sets are structured and how to access them. This chapter contains three parts. The first part describes the radar data set and how the radar images are converted to rain intensities. The second part gives a description of the NWP model used in this thesis. The parameters found in the model are also shown here. The last part describes the availability and quality of both data sets.

### 2.1. Description of the radar data

The radar data used in this thesis comes from the Royal Netherlands Meteorological Institute (KNMI). It contains forecasts of radar reflectivity composites at a height of 1500 meters over the Netherlands and surrounding areas. The radar reflectivity composites are forecasted every five minutes up to a lead time of two hours. The radar data used in this study span the summer season of 2019, namely May, June, July and August. A total of 2721 hours of data were collected.

#### 2.1.1. Radar observations

The first radar reflectivity composite is not predicted but observed by two radars. One radar is installed in Herwijnen (51.8369 N, 5.1381 E) and the other one in Den Helder (52.9533 N, 4.7900 E). Both radars measure radar reflectivity at four different elevation angles (0.3, 1.1, 2.0, 3.0 degrees) and have a maximum observation range of 320 km. The measurements at different elevation angles, also called scans, are then merged into one radar image for each radar, containing radar reflectivities at 1500 m. The radar image is generated by linearly interpolating the radar reflectivity at different elevation angles. The two radar images are then combined by computing the weighted average of the radar reflectivities. The weights are defined based on the distance from a given point to the radar. This results in a radar reflectivity composite of pseudo constant altitude plan position indicator (PseudoCAPPI) with pixels representing an area of 1 by 1 squared kilometer. More information about how KNMI produces these reflectivity composites can be found in [37].

#### 2.1.2. Radar forecasts

Each data file contains one observed reflectivity composite and 24 forecasts based on this observation. These forecasts are all based on the first radar reflectivity composite, which is observed by the radar. The forecasts contain a prediction of the precipitation for every five minutes with a lead time up to two hours. This is done by assuming Lagrangian persistence, meaning that the precipitation rate does not change over time. This assumption neglects thus any growth and decay. The method of estimating the motion field by KNMI is not published and the details remains unknown. But the estimates are very similar to those obtained by optical flow. The motion field is then used for the advection of the observed radar reflectivity composite. The 1h lead time will be the main focus in this thesis, because most of the predictive skill is then lost due to growth and decay of rain cells.

### 2.1.3. Conversion data to rain intensity

The radar data is stored in an HDF5 file format and can easily be accessed in python with the h5py package. Appendix A shows how to access the radar files using h5py. The reflectivities are saved as 8 bit integers, ranging from 0 to 255. Pixels with values of 0 and 255 represent missing or out of range data, respectively. The conversion from 8 bit integer to reflectivity  $Z$  ( $mm^6/m^3$ ) is done in the following manner:

$$z = 0.5(px - 64) \quad (2.1)$$

$$Z = 10^{z/10} \quad (2.2)$$

where  $px$  represents the pixel value and  $z$  is the logarithmic reflectivity in  $dBZ$ . The Z-R relationship, found in Eq. 2.3, is used to convert from reflectivity  $Z$  ( $mm^6/m^3$ ) to rainfall intensity  $R$  ( $mm/h$ ). KNMI uses the standard Z-R relationship, also known as Marshall Palmer equation [38], with coefficient  $a = 200$  and  $b = 1.6$ . The same coefficients are used in this thesis.

$$Z = 200R^{1.6} \quad (2.3)$$

$$R = (Z/200)^{1/1.6} \quad (2.4)$$

Figure 2.1 shows an example of a radar reflectivity composite which is converted to precipitation rate. Thanks to the 1000 meter resolution grid, also smaller rain cells can be detected with enough detail. The high resolution also makes it possible to accurately estimate the motion field. The composite uses a stereographic projection and can be found inside each data file as a proj.4 definition. The coordinates of each pixel are converted to WGS84 in order to work on one projection for both radar and NWP model.

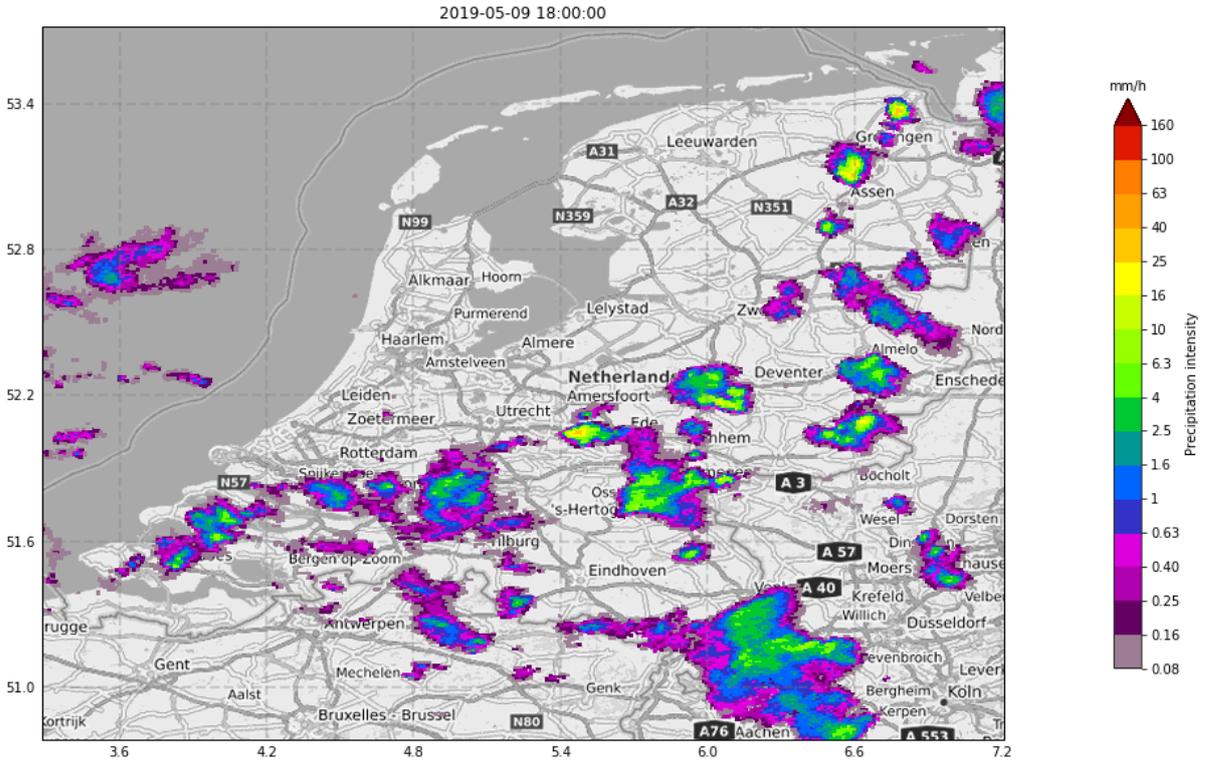


Figure 2.1: Observed rain intensity on 2019-05-09 18:00 after conversion to precipitation rate  $R$  ( $mm/h$ ).

## 2.2. Description of the NWP

As mentioned in chapter 1, NWP models predicts the state of the atmosphere ahead of time based on mathematical models using the current atmospheric conditions. KNMI has developed, in collaboration with other national meteorological services, two models, namely the High Resolution Limited Area Model (HIRLAM) and HARMONIE (HIRLAM ALADIN Research on Mesoscale Operational NWP in Euromed). Both of these models are specifically designed for short term weather predictions, with HARMONIE having a higher resolution in the spatial domain. KNMI computes HIRLAM and HARMONIE on a grid with resolutions of 11 km and 2,5 km respectively.

KNMI provides HARMONIE data with hourly predictions up to 48 hours into the future every 6 hours with a spatial resolution of 2,5 km. It has been available since 2012 and is run four times a day at 00, 06, 12, and 18 o'clock. Each run takes around 2.5 to 3 hours to complete. From that, surface parameters are produced and distributed through the KNMI datacenter. Data are only stored for 24 hours after which they are overwritten. In this thesis, we decided to use HARMONIE because of 1) data availability, 2) the relatively high spatial resolution of 4 km compared with other weather models and 3) its ability to predict mesoscale phenomena such as thunderstorms. Note that only the surface parameters are available for free on the KNMI data center. However, in the future, the full NWP output could be used. This would provide more information about the vertical variability.

The data set has been, just as the radar data set, manually downloaded every day and overlaps the radar data for the biggest part. The data set is stored in multiple GRIB files, with each file representing a forecast with respect to the initial start time. Accessing the data is not as easy as with the radar data, the package `cfrib` has been used together with some modifications to the default configuration of this package. Appendix B shows which steps has to be taken and how to access the data. The coordinates are stored as latitude and longitude pairs in the coordinate system WGS 84.

Each file contains multiple parameters, such as temperature and rain intensity at a certain height level (see Figure 2.2). All parameters are shown in Table 2.1. PMSL and BRT are not accessible because they are on a different height system than the other parameters. Also, accumulated parameters are not present in the zero hour prediction files. These parameters are therefore not used and removed from the data set.

Figure 2.2 shows on the left the observed rain intensity and on the right the predicted rain intensity by the HARMONIE model. The prediction is a 3 hour forecast, but the rain event is still clearly visible. The lower resolution of HARMONIE is also noticeable and the precipitation is smoother.

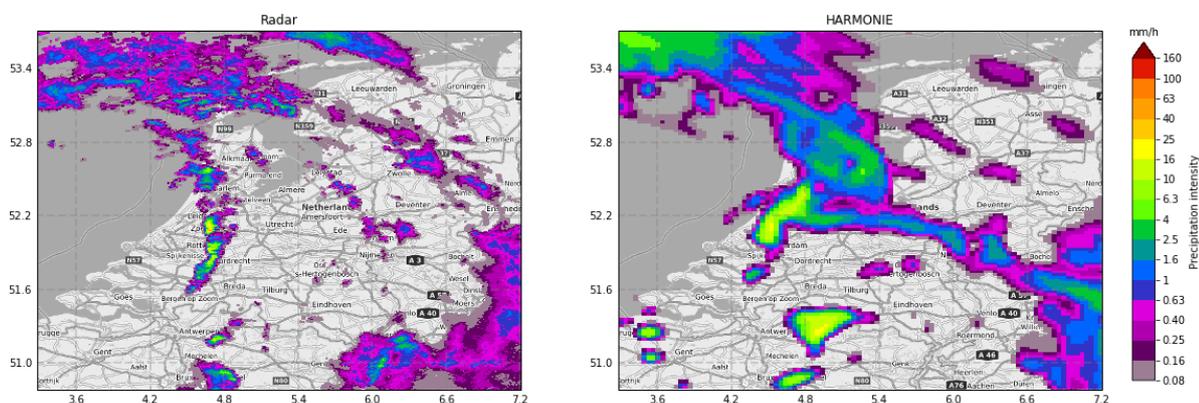


Figure 2.2: On the left the observed rain intensity by the radar and on the right the predicted rain intensity by the HARMONIE model for 2019-05-08 15:00. The HARMONIE model was run based on the latest data assimilated at 12 o'clock and the forecast time is thus 3 hours.

## 2.3. Data availability and quality

### 2.3.1. Radar

The radar data from the KNMI is not perfect. The access to the 5-minute forecasted radar reflectivity composites is limited and only the last 24 hours of observations can be downloaded. This has led to some missing observations in the data set and also to some corrupted files, which were removed from

Table 2.1: Near surface parameters found in the data set from the KNMI HARMONIE model. The code values refers to the parameter ID in the GRIB Parameter Database<sup>1</sup>. The level values show the height above ground, except for PMSL (height above sea) and BRT (Nominal top of atmosphere).

Parameter	Description	Code	Level	Cumulative	Units
PMSL	Pressure on mean sea level	001	0	-	<i>Pa</i>
U10	U-wind component at 10 m	033	10	-	<i>m s<sup>-1</sup></i>
V10	V-wind component at 10 m	034	10	-	<i>m s<sup>-1</sup></i>
2T	Temperature of air at 2 m	011	2	-	<i>K</i>
2RH	Relative humidity at 2 m	052	2	-	%
TCC	Total cloud cover	071	0	-	%
LCC	Low Cloud Cover (surface to 748 hPa)	073	0	-	%
MCC	Medium Cloud Cover (748 to 424 hPa)	074	0	-	%
HCC	High Cloud Cover (above 424 hPa)	075	0	-	%
CR	Accumulated rain from start runtime	061	457	X	<i>kg m<sup>-2</sup></i>
CS	Accumulated snow from start runtime	062	457	X	<i>kg m<sup>-2</sup></i>
CG	Accumulated graupel from start runtime	063	457	X	<i>kg m<sup>-2</sup></i>
INR	Rain precipitation rate	061	456	-	<i>kg m<sup>-2</sup> s<sup>-1</sup></i>
INS	Snow precipitation rate	062	456	-	<i>kg m<sup>-2</sup> s<sup>-1</sup></i>
ING	Graupel precipitation rate	063	456	-	<i>kg m<sup>-2</sup> s<sup>-1</sup></i>
SC	Snowcover	066	0	-	
BLH	Boundary layer height	067	0	-	<i>m</i>
GR	Global Radiation	117	0	X	<i>J m<sup>-2</sup></i>
SWR	Net short-wave radiation	111	0	X	<i>J m<sup>-2</sup></i>
LWR	Net long-wave radiation	112	0	X	<i>J m<sup>-2</sup></i>
LH	Latent heat flux	121	0	X	<i>J m<sup>-2</sup></i>
SH	Sensible heat flux	122	0	X	<i>J m<sup>-2</sup></i>
BRT	Brightness temperature	118	39680	-	<i>J m<sup>-2</sup> s<sup>-1</sup></i>
CLB	Cloud base	135	0	-	<i>m</i>
CIG	Column integrated graupel	201	458	-	<i>kg m<sup>-2</sup></i>
UWG	U-component maximum gust at 10 m	162	10	-	<i>m s<sup>-1</sup></i>
VWG	V-component maximum gust at 10 m	163	10	-	<i>m s<sup>-1</sup></i>

the data set. The radar data set also contains artifacts. These artifacts can be caused by a large variety of objects. A radar transmits electromagnetic pulses and then measures the power after the has been scattered back by an object. In the case of a weather radar this back scattering is preferably caused by precipitation, however backscatter can also be caused by other objects like buildings, windmills, wind turbines and birds or insects. Most of the clutter in radar images can be found above sea and also at above 200 km distance from the radars. An example of this clutter is shown in Figure 2.3.

### 2.3.2. HARMONIE

Because HARMONIE is physical model, it may contain some information about the dynamical behaviour of rain cells. This information could be useful for determining growth and decay of rain cells. However, like any other NWP model, HARMONIE lacks accuracy in both spatial and temporal domain. Rain cells could be predicted at a certain location but could be observed at a different location in the radar. Also, HARMONIE does not have the ability to predict rain cells with a lifetime of tens of minutes, due to the hourly resolution. The lifecycle of a convective rain cell is too short to be resolved by the model. Figure 2.4 shows a good example of the lack in spatial accuracy and the inability to predict smaller scale rain cells. Nonetheless, large scale trends over a few hours and hundreds of kilometers can still be captured, as shown in Figure 2.2.

HARMONIE also does not contain any artifacts or clutter unlike radar.

<sup>1</sup>The database can be found at <https://apps.ecmwf.int/codes/grib/param-db/>

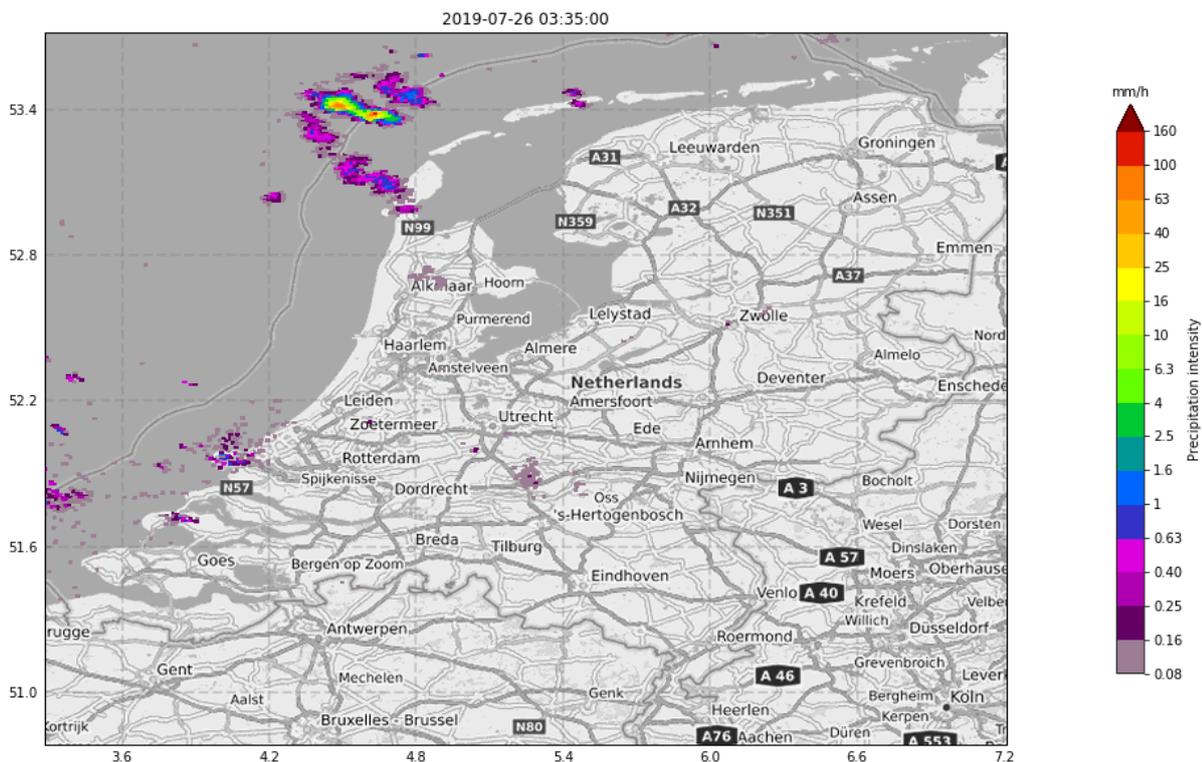


Figure 2.3: Precipitation rate observed on 2019-07-26 03:35 over the Netherlands. Artifacts in this observation are noticeable around the coast near Maasvlakte 2.

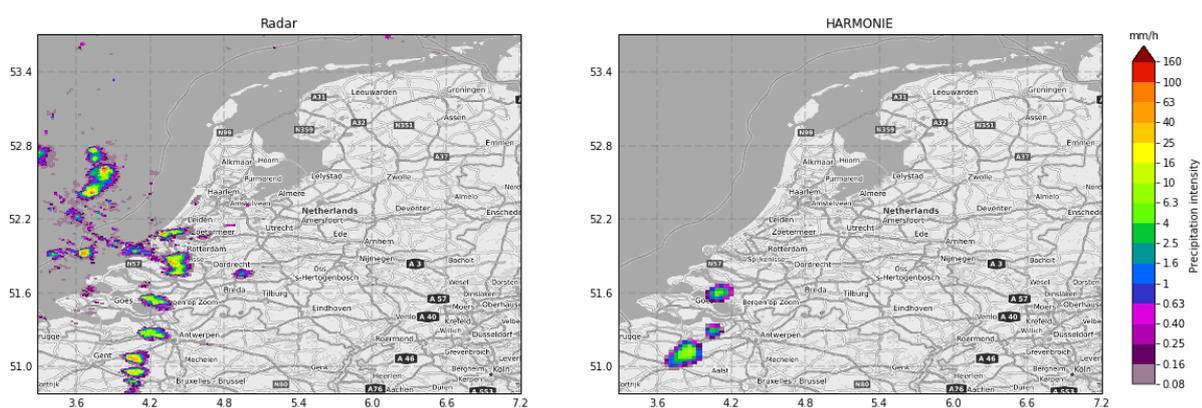


Figure 2.4: On the left the observed rain intensity by the radar and on the right the predicted rain intensity by the HARMONIE model for 2019-06-02 18:00.



# 3

## Methods

The notion of growth and decay of rain cells is very broad and can be defined in multiple ways. In the first part of this chapter, the formal definition of growth and decay used in this thesis as well as its mathematical formulation are given. The procedure of computing the growth and decay involves adapting the Lagrangian framework in order to follow a precipitation parcel along its trajectory. This enables the tracking of change in precipitation rate over time. The second part of this chapter, introduces the multi-layer perceptron (MLP). This is a machine learning technique which is very versatile in its use as it can be used for both classification and regression problems. MLP will be used in this thesis to predict the growth and decay. Not all of the available data will be used for this. From the available data only days with rain will be used. The last paragraph explains step-by-step how the data are filtered.

### 3.1. Study area

In order to reduce the amount of data and the computational time, only a small area of the Netherlands will be taken into consideration. The area of interest over which the growth and decay will be predicted is a  $20 \times 20 \text{ km}^2$  square over the city of Rotterdam centered around the location of the Rijnmond X-band radar on top of the Delftse Poort at the coordinates 51.923630 N, 4.471657 E. This area will be called the target area and denoted as  $A^{20}$ . A larger area surrounding  $A^{20}$  and acting as a buffer zone will also be used. This is important as rain cells move around and information could be found around the target area. The larger area covers an area of  $60 \times 60 \text{ km}^2$ , overlapping the observation area of the Rijnmond radar and will be denoted as the buffer area  $A^{60}$ . The size of the buffer zone is chosen to be  $60 \times 60 \text{ km}^2$ , because the Rijnmond radar, once it is operational, will have a range of up to 30 km over the same area. Also, the average wind speed in the summer is around  $20 \text{ km h}^{-1}$  and rain cells will still be inside the buffer area after 1 hour has passed by. Figure 3.1 shows the two areas. To minimize the confusion about which area is used, superscripts are used to denote the area. For example,  $\bar{P}^{20}$  and  $\bar{P}^{60}$  stand for the average precipitation rates over the  $20 \times 20 \text{ km}^2$  area and  $60 \times 60 \text{ km}^2$  area respectively.

### 3.2. Definition of growth and decay

Rain cells grow and decay in various ways. They can grow and decay in terms of size, meaning that a rain cell grows when it covers a larger area at time  $t$  with respect to time  $t - \tau$ . Conversely, decay means that the rainy area becomes smaller. Growth and decay can also refer to the strengthening or weakening of a rain cell. Growth (decay) is then linked to the increase (decrease) of precipitation rates. This thesis only focuses on the latter definition, i.e., the strengthening or weakening of the average rain rate over an area.

#### 3.2.1. Tracking of precipitation parcel

Due to motion, precipitation parcels observed at time  $t - \tau$  will be at a different location at time  $t$ . The Lagrangian framework allows for the physical properties to be studied as we follow each precipitation

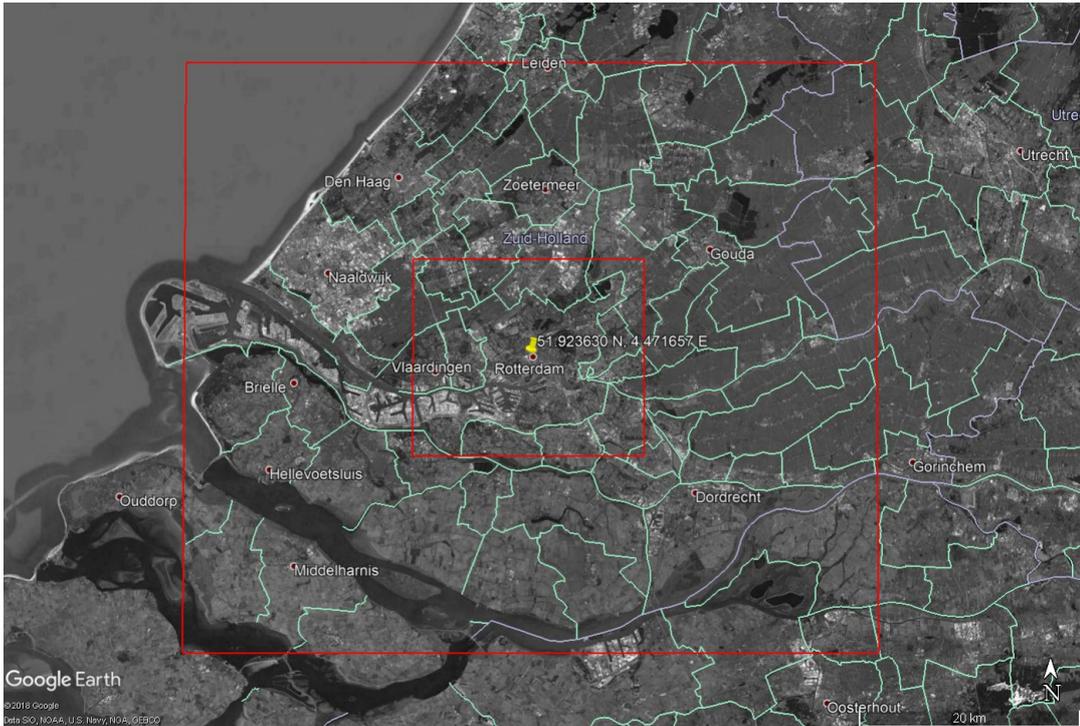


Figure 3.1: The two areas ( $A^{20}$  and  $A^{60}$ ) used for the prediction of growth and decay of rain cells.

parcel along its trajectory. The Lagrangian framework is adopted by first estimating the motion field over the area between time  $t - \tau$  and  $t$  and then computing the trajectory of each parcel. A backtracking algorithm is then used to estimate the original location of each parcel at time  $t - \tau$ , see Figure 3.2.

#### Estimation of the motion field

The motion field describes the apparent motion of a parcel at any location in the observation field. The motion field is estimated using the Lucas-Kanade optical flow method provided by the python package pySTEPS [39, 40]. This is different from the object-tracking method used by KNMI but our analyses show that both approaches produce very similar motion estimates. The Lucas-Kanade method assumes that the pixels inside a window are moving in one direction with the same speed. This is based on the assumption that the moved distance of an object inside the image is small. This means that the optical flow equation must hold within a window for all pixels, resulting in a linear system of equations. The optical flow method only estimates the motion field around rain cells as it detects corners of objects inside an image. This is to reduce the computational time and noise.

Two consecutive radar fields are used to estimate the motion field. The first is the reflectivity field at time  $t - \tau$  and the second is the reflectivity field at time  $t$ . The motion field is therefore the motion from time  $t - \tau$  to  $t$ . Normally in radar nowcasting, the motion field is based on radar images until time  $t - \tau$ , because the radar image at time  $t$  is not observed yet. This introduces additional uncertainties, because similarly to the intensity, the motion field is not constant and can vary over time. In this thesis, these additional uncertainties are avoided by including the radar image at time  $t$ , therefore leading to the best possible estimate of the motion field.

### Backtracking

The motion field is used to compute the displacement terms  $\alpha$  and  $\beta$ , representing the distance traveled in the  $x$  and  $y$ -direction respectively. These can be computed in two ways. The first way is called the constant-vector scheme. The displacement terms are then computed as:

$$\begin{aligned}\alpha &= \tau u(x, y) \\ \beta &= \tau v(x, y)\end{aligned}\quad (3.1)$$

where  $u(x, y)$  and  $v(x, y)$  represent the estimated motion between time  $t - \tau$  and  $t$  in  $x$  and  $y$ -direction respectively at location  $(x, y)$ . The constant-vector scheme translates each grid point only once and does not allow for rotation. The second way to compute the displacement terms is by using a semi-Lagrangian scheme. The time lag  $\tau$  is divided into smaller time steps and the displacement is computed recursively, in the same manner as in the constant-vector scheme except that the location is updated at every time step. Depending on the amount of time steps and how quickly the motion field changes, this semi-Lagrangian scheme can account for rotation up to a certain degree. To simplify the analysis, the displacement terms in this thesis are computed with the constant-vector scheme. The error introduced by not accounting for the rotation is small as the parameters are averaged over a relatively large area anyway.

By convention, the motion field goes forward in time and therefore the displacement terms points away from the origin  $(x_{origin}, y_{origin})$  to the target location of the rain cell  $(x_{target}, y_{target})$  at time  $t$ . The origin of the rain cell at time  $t - \tau$  is thus found by subtracting the displacement terms, see Eq. 3.2. Figure 3.2 shows the displacement terms, which can be computed by the two schemes.

$$\begin{aligned}x_{origin} &= x_{target} - \alpha \\ y_{origin} &= y_{target} - \beta\end{aligned}\quad (3.2)$$

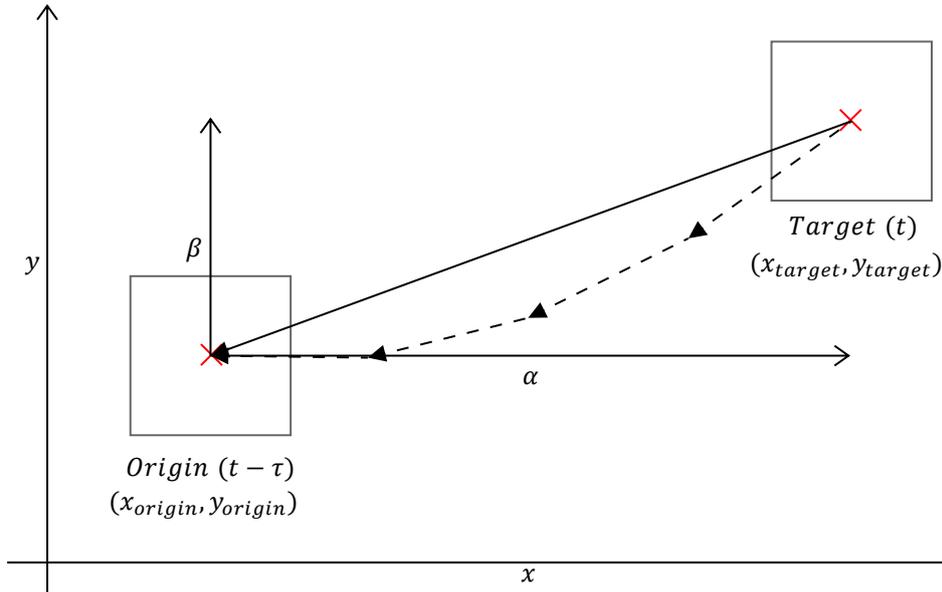


Figure 3.2: Displacement terms  $\alpha$  and  $\beta$  used in the backtracking of rain cells. The solid line and dashed line represent the trajectory computed by the constant-vector scheme and semi-Lagrangian scheme respectively.

### 3.2.2. Formulation of growth and decay

The growth and decay can be mathematically defined in two ways. A simple way to formulate growth and decay is with the help of a multiplicative model, shown in Eq. 3.3. The growth and decay  $C^{mult}$  between time  $t - \tau$  and  $t$  is formulated as the ratio between the average precipitation rates  $\bar{P}$  at time  $t - \tau$  and  $t$ . The movement of the precipitation parcels are accounted for by adopting the Lagrangian

framework as described in section 3.2.1. This means that  $\bar{P}(t)$  is computed over the target area and  $\bar{P}(t-\tau)$  over the backtracked area, i.e. the origin in Figure 3.2. The multiplicative model always outputs a positive value ranging between 0 and infinity. Growth occurs when  $C^{mult}$  is larger than 1 and when the average precipitation rate decreases the ratio  $C^{mult}$  will be less than 1. One major drawback for this model is that it can only be used when there is already precipitation. The average precipitation rate  $\bar{P}(t-\tau)$  should be always greater than 0. This means that initiation of rain cells can not be predicted with the multiplicative model.

$$\bar{P}(t) = \bar{P}(t-\tau)C^{mult}(t) \quad (3.3)$$

Growth and decay can also be formulated as an additive model, see Eq. 3.4. The growth and decay  $C^{add}$  shows the difference between the average precipitation rate  $\bar{P}$  between time  $t-\tau$  and  $t$ . This formulation of the growth and decay is easy to understand as it represents the actual change in precipitation rate in mm/h. Positive values stand for growth and negative values for decay. The additive model can also be used to study the initiation of rain cells.

$$\bar{P}(t) = \bar{P}(t-\tau) + C^{add}(t) \quad (3.4)$$

In this thesis we will use the additive model for the estimation of the growth and decay. However, logarithmic reflectivity  $z$  (dBZ) will be used instead of rainfall rates (mm/h), because reflectivity fields are directly measured by the radar and have a more Gaussian distribution. Also, the impact of an error in the lower precipitation rates is higher compared to the same error at a higher precipitation rate. Even though the absolute error is the same, the impact of this error is different at different scales of precipitation. A difference of 5 mm/h has more impact for the precipitation rates between 1 – 10 mm/h than for 20 – 30 mm/h. The relative error is therefore more important than the absolute error. The symbol for average precipitation rate  $\bar{P}$  thus changes to average logarithmic reflectivity  $\bar{z}$ :

$$\bar{z}(t) = \bar{z}(t-\tau) + \Delta z_{t-\tau,t} \quad (3.5)$$

The radar observations from the KNMI are used for the reflectivity field  $z$  at time  $t$  over the Netherlands. The precipitation parcels are then traced back to its original position at time  $t-\tau$  using the constant-vector scheme. Because we are interested in the mean growth or decay, the reflectivity fields are first averaged over the area  $A^{20}$  and then the mean growth and decay is computed. Resulting in Eq. 3.5 which is used to compute average growth and decay over area  $A^{20}$ , where  $\Delta z_{t-\tau,t}$  is the average growth and decay.

Due to the temporal resolution of HARMONIE, this thesis will focus on predicting growth and decay for a lead time  $\tau = 60$  minutes.

### 3.3. General approach

The idea behind using a numerical weather prediction model to improve the radar nowcasts is simple. There exists a tradeoff between accuracy in time and space and the dynamical part of rain cells. Radar nowcasts are able to predict the location of a rain cell fairly accurate at the right time. However the shortcomings of radar nowcasts is that they do not account for any change in size or intensity of rain cells. In contrast to radar nowcasts, numerical weather prediction models are less accurate when it comes to predicting rain cell at the right location and time, but are able to predict the dynamics of rain cells. Examples can be found in Chapter 4.

Information about the dynamics may be present directly in the different parameters at time  $t$ , the time we want to predict the growth and decay. Since we are only interested in the areal average growth and decay, looking at the trend or change of the parameters would be the better choice. Therefore, only the trend of the parameters from HARMONIE are used. The trends are computed, just like the growth and decay, using the additive model, see Eq. 3.5. The parameters from HARMONIE are backtracked, averaged and then the trend is computed by subtraction.

The information from the trends however is still not directly usable to predict growth and decay. For this, a machine learning technique needs to be used to make sense out of the information from HARMONIE and to predict growth and decay. The multi-layer perceptron (MLP) will be used for the prediction of growth and decay. Other machine learning techniques could be used, but MLP is very flexible as it can be used for both classification and regression problems. Also MLP can be used deal

with non linear problems and is able to produce predictions very fast, once a model is trained. More information about MLP and how it is used can be found in chapter 3.4.

### 3.3.1. Sliding window method

It should be noted that the performance of the MLP depends on the information it gets. Too much contradictory data or too much noise in the data will result in the MLP not being able to make sense out of the data. This could be also the case with the HARMONIE data, because the data itself is a prediction and is not 100% accurate. Therefore trusting all of the information from HARMONIE might be a bad idea. A better option might be to only predict growth and decay for cases where the accuracy of the information from HARMONIE is high enough. In the remainder of the cases, it is better to assume Lagrangian persistence, i.e. no growth and decay.

The accuracy for each prediction could be measured by comparing the growth and decay from HARMONIE at time  $t$  with the radar observation at time  $t$ . This is however not possible as the radar observation at time  $t$  is not present. An alternative option is to use the historical performance of HARMONIE. The performance of previous predictions can be used as an indication of how good the information from HARMONIE at time  $t$  is. If the previous predictions are consistent with the corresponding observations, chances are high that the image at time  $t$  will also be consistent. For this purpose, the sliding window method is used to move over the data set and to compute the historical performance of HARMONIE for each prediction. The sliding window method uses a window, with a certain length  $L$ , which moves over each prediction and computes the performance of the previous  $L$  predictions.

The historical performance could be computed in two different ways. First option is to compute the correlation between growth and decay from radar and HARMONIE. The correlation is computed over the points inside the window. Correlation shows how similar the two time series are. The downside however is that it does not show how good the agreement is between radar and HARMONIE. So HARMONIE and radar could both be contradicting each other regarding growth and decay but the correlation could still be high. The second option is to count how many times HARMONIE agrees with radar within the window. HARMONIE agrees with radar if both show growth or decay. A high number of agreements means that there is high chance that the prediction for time  $t$  will also be correct. Both of the performance quantities will be used to assess the performance of HARMONIE. How the cases where HARMONIE could be used to predict growth and decay will be chosen is through MLP and is explained in chapter 5.

### 3.3.2. Shift correction method

Not only limiting the amount of data given to the model could improve the results. HARMONIE does not always predict the rain cells at the right location at the right time. Correcting these inconsistencies could lead to an increase in accuracy of the information from HARMONIE. This can be accomplished by aligning the precipitation images from HARMONIE with the radar observation. This will reduce the chance of the rain cell being at the wrong location.

Aligning the images can be achieved by computing the correlation between two images for all possible shifts and looking for the maximum correlation, this method is also called cross correlation. Computing the cross correlation directly is computationally expensive and takes a considerable amount of time as each radar image contains  $700 * 765$  pixels. A method called Phase correlation [41, 42] uses the properties of the Fourier Transform in order to efficiently compute the cross correlation between two images. Directly computing the cross correlation involves a convolution operation making it very expensive for a large amount of pixels. However, the convolution operation in the spatial domain is equivalent to a multiplication in the frequency domain. Transforming the images to the frequency domain using the Fast Fourier Transform reduces the computational time significantly. Phase correlation uses this property to efficiently compute the correlation surface  $c_{I_1, I_2}$ :

$$c_{I_1, I_2}(x, y) = \mathcal{F}^{-1}(\mathcal{F}(I_1)\mathcal{F}(I_2)^*) \quad (3.6)$$

where  $\mathcal{F}(I_1)$  and  $\mathcal{F}(I_2)$  are respectively the Fourier Transform of the images  $I_1$  and  $I_2$ ,  $\mathcal{F}^{-1}$  denotes the inverse Fourier Transform and  $\mathcal{F}(I_2)^*$  stands for the complex conjugate of the Fourier Transform of image  $I_2$ . The correlation  $c_{I_1, I_2}(k, l)$  at coordinates  $(k, l)$  shows the correlation between the two images corresponding to shifting image  $I_2$  by the difference between the center of the surface and coordinates

$(k, l)$ . The coordinates with the maximum correlation are therefore the best estimate for translation shift.

#### Phase correlation algorithm

Multiple steps have to be taken to find the shift between the radar precipitation rate and precipitation rate from HARMONIE. The computation of the method consists of the following steps:

1. Ensure that both precipitation rate images are in the same units ( $mm/h$ ).
2. Convert the HARMONIE coordinates to the radar projection and resample the HARMONIE image so that both images have the same grid and dimensions. Using the python package `scipy`, the HARMONIE images are resampled to the radar grid. Resampling is done with the function `scipy.ndimage.map_coordinates()` which maps the image to a new grid by cubic spline interpolation.
3. Compute the correlation surface  $c_{I_{radar}, I_{HARM}}$ 
  - (a) Compute the Fourier Transform of both images  $\mathcal{F}(I_{radar})$  and  $\mathcal{F}(I_{HARM})$ . The 2-D Fourier Transform is computed using the function `np.fft.fft2()` from the python package `numpy`.
  - (b) Compute the cross power spectrum,  $\mathcal{F}(I_{radar})\mathcal{F}(I_{HARM})^*$ .
  - (c) Take the inverse Fourier Transform of the cross power spectrum and rearrange this by shifting the zero component to the center of the surface. The correlation surface  $c_{I_{radar}, I_{HARM}}$  needs to be rearranged because the function saves the negative terms at the end of the surface. The function `np.fft.fftshift()` is used to rearrange the components.
4. Find the translation shifts  $(\Delta x, \Delta y)$ .
  - (a) Find coordinates of maximum correlation  $(x_{max}, y_{max})$ .
  - (b) Compute the translation shifts. The shift is computed as the difference between the center of the surface  $(x_c, y_c)$  and  $(x_{max}, y_{max})$ , i.e.  $(\Delta x = x_{max} - x_c, \Delta y = y_{max} - y_c)$ . A negative  $\Delta x$  means that HARMONIE should be moved to the west by  $|\Delta x|$  km, and vice versa. A negative  $\Delta y$  means that HARMONIE should be moved to the north by  $|\Delta y|$  km, and vice versa.

The same problem, as for measure the performance of HARMONIE, arises that there is no radar observation at time  $t$ , so the HARMONIE image cannot be aligned with the corresponding radar observation. Assuming that the best shift for time  $t$  does not deviate much from the shift for time  $t - 60$ . We could solve this issue by estimating the best shift for the previous image at time  $t - 60$  and applying this shift to the image at time  $t$ . Also, it is possible to extend the phase correlation algorithm to estimate the best shift for a series of images. This thesis will cover both cases.

### 3.4. Multi-layer perceptron

In this thesis a MLP (multi-layer perceptron) is used for the prediction of growth and decay. MLP is also called a feedforward neural network [43]. It is comprised of multiple layers, each of which containing a certain number of neurons(nodes). Figure 3.3 gives an example of a three-layer perceptron (3LP). The number of neurons within the  $i$ th layer is denoted as  $N_i$ . The first layer is called the input layer. The features or data are directly fed into this layer and the number of neurons is the same as the number of features. The last layer is the output layer. It can contain multiple neurons depending on the desired output. Since the goal is to predict the growth and decay at a certain moment, the output layer only has one neuron. The layers between the first and the last layer are called hidden layers. These layers are used to map the input (features) to the desired output. The output  $a_{i,j}$  of a node  $j$  within a hidden layer  $i$  is the weighted sum of each neuron in layer  $i - 1$ . A non linear activation function  $h$  is then applied to the weighted summation, making the MLP non linear. In this thesis, the hyperbolic tangent function is used as the activation function  $h$ . This is a natural choice for modeling growth and decay as the range is between  $-1$  and  $1$  and the function is centered around zero. This ensures the MLP deals with both negative and positive values in the same manner. Figure 3.3 and Eq. 3.7 shows the output of a node  $j$  in layer  $i$ . The sum is over all neurons  $k$  in layer  $i - 1$ .

$$\begin{aligned}
 z_{i,j} &= \sum_k a_{i-1,k} w_{i,j,k} \\
 a_{i,j} &= h(z_{i,j}) \\
 h(x) &= \tanh(x)
 \end{aligned}
 \tag{3.7}$$

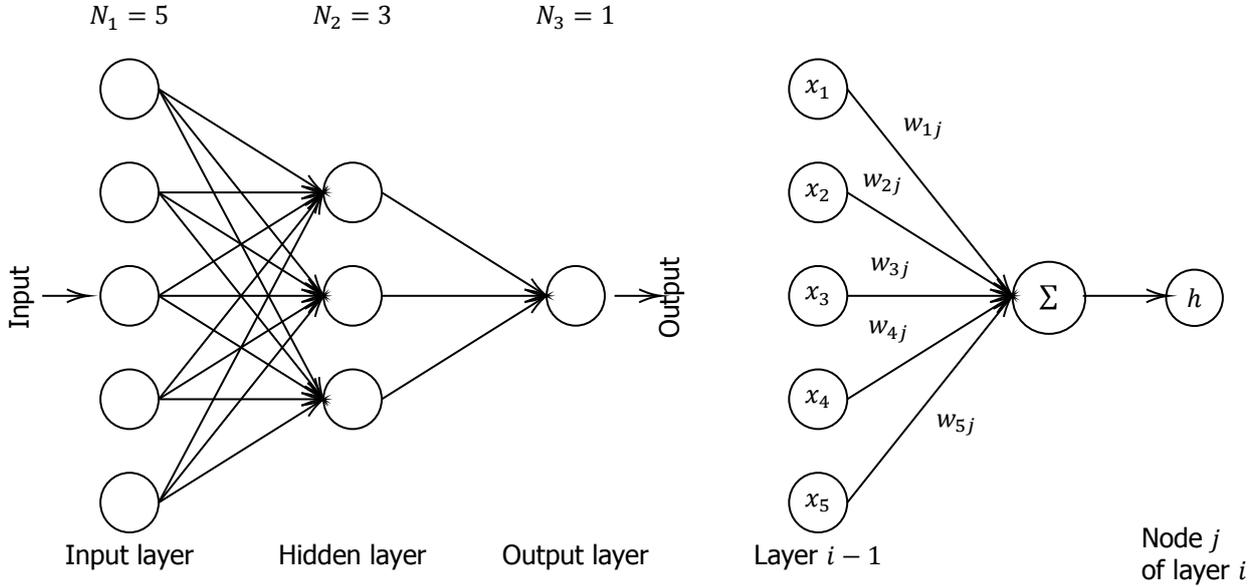


Figure 3.3: On the left: an example of a three-layer perceptron. On the right: a visual representation of how the output of a node is composed.

### 3.4.1. Backpropagation

The weights found in Eq. 3.7 are estimated by backpropagation. They are updated iteratively according to the stochastic gradient descent algorithm Adam [44]. One iteration is also called an epoch. This algorithm is expected to work well with noisy data. It uses the exponential moving averages of the first and second moments of the gradients. These two averages are then used to update the weights of the neural network model. For a certain weight  $w$ , the algorithm consists of the following steps:

$$\begin{aligned}
 g_t &= \frac{\partial J}{\partial w_t} \\
 m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
 v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
 \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\
 \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\
 w_t &= w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
 \end{aligned}
 \tag{3.8}$$

where  $J$  represents the cost function, which is minimized with the Adam algorithm [44]. The exponential moving average of the first and second moments are formulated by  $m_t$  and  $v_t$  at iteration  $t$ , respectively. These are then corrected for the bias. The moving averages use  $\beta_1$  and  $\beta_2$  as the exponential decay rate for the first and second moment, respectively and  $\epsilon$  is used to prevent division by zero. The settings recommended by [44] are used,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 10^{-8}$ . The learning

rate  $\alpha$  depends on the input and the size of the model and is changed for each model. The vanishing gradient problem is mitigated by keeping the number of hidden layers as low as possible.

The neural network models are created and trained using the python package Tensorflow [45], which is an open-source machine learning package from Google. Also the stochastic gradient descent algorithm Adam was already implemented in the package.

### 3.4.2. Validation of the neural network

The cost function  $J$  used in the backpropagation is chosen to be the mean squared error (MSE). The mean squared error measures how good the model fits the data in a squared sense. Larger errors will therefore be penalized more than smaller ones.

The issue of under and overfitting is minimized by keeping the size and complexity of the neural network models to a minimum. Only 80% of the training data set will be used to train the model. The remaining part is used to evaluate the behaviour of the model. The MSE of this part is computed after every epoch and is used to avoid overfitting the model. The training of the model will be stopped if there is no improvement in the MSE after five epochs. This method is called early stopping.

To evaluate the models, mean squared error (MSE), mean bias error (MBE) and Pearson's correlation coefficient ( $\rho$ ) will be computed over the validation data set. MBE represents the systematic error of the model to under or overestimate the growth and decay. A perfect model would result in a systematic error of  $MBE = 0$ , i.e., no under or overestimation of the growth and decay. The Pearson's correlation coefficient measures the linear relationship between the growth/decay and the predicted growth/decay. The correlation for a perfect model should be  $\rho = 1$ . The performance metrics are defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (3.9)$$

$$MBE = \frac{1}{N} \sum_{i=1}^N \hat{y}_i - y_i \quad (3.10)$$

$$\rho = \frac{\sigma_{\hat{y}y}}{\sigma_{\hat{y}}\sigma_y} \quad (3.11)$$

where  $y_i$  and  $\hat{y}_i$  represents the true and estimated value of data point  $i$  respectively.

## 3.5. Event selection

The radar archive contains a lot of time steps where no rain was observed in area  $A^{20}$ . A flowchart showing the procedure used to select the events is shown in Figure 3.4. The first step is to select a set of potential events. This is done by using two criteria:

1. The fraction of wet pixels in the target area  $A^{20}$  is greater than 15%. A pixel is considered wet when the precipitation rate (reflectivity) is higher than 0.2 mm/h (11.827 dBZ).
2. The maximum precipitation rate (reflectivity) in the target area  $A^{20}$  is larger than 10 mm/h (39.010 dBZ).

The threshold of 0.2 mm/h is used to minimize the noise found in the observations. If a radar observation meets both criteria, then all of the observations of that day are added to the data set. This is to ensure that the growth or decay is present in the observations, but this is not always the case as rain cells are not fixed to a specific location and move around. The HARMONIE data set is then added to the data set. Note that some files were missing or mislabeled, due to the way the two data sets have been gathered. The records which depend on a file which is missing, misnamed or corrupted were removed. The data set with the remaining records were then visually checked for artifacts. All suspect cases were removed.

In total, there were 39 days in the radar data set that met both criteria. However, 11 days of radar data were removed after checking for artifacts. As a result 28 days of data in 2019 were used in this thesis. The month of May had the least amount of rain events, with only four days of rainfall. June

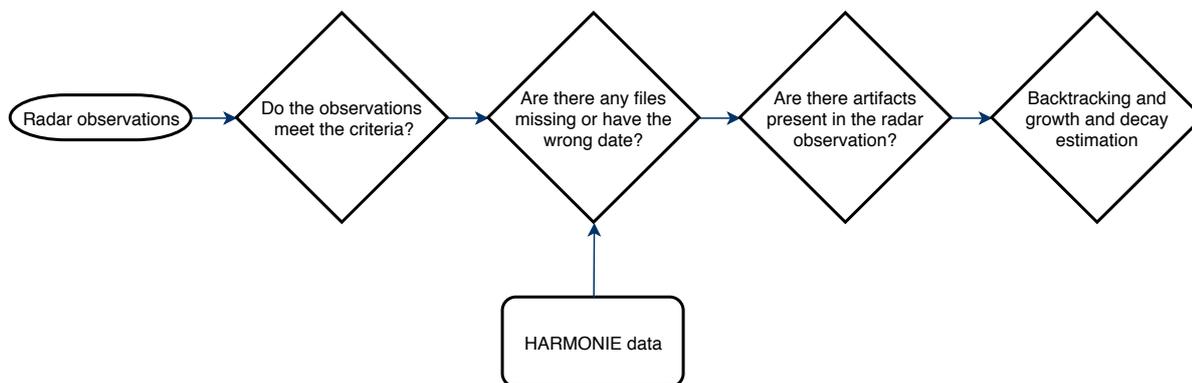


Figure 3.4: Process of selecting the events

however started with warm days and little rain. From the 10th of June, rain events started to occur almost daily up until the 19th. The 25th of June 2019 had met the criteria, but was removed from the data set as there were a lot artifacts in the observations. Four days in the month of July were also removed for the same reason. Rain started to fall again at the end of July and continued into August. August had the largest number of rain events, with a total of 13 days of rain.

### 3.6. Construction of the training and validation data set

As aforementioned, the data set exists only of days which include a rain event. This data set is split into two data sets, a training and validation data set. The training data set is used for the training of the machine learning technique by optimizing the weight for each neuron. To tune the models hyper-parameters, such as the number of hidden layers and neurons, a k-fold cross validation is used only on the training data. The validation data set is used for validating the model.

The data set is split based on the date of the observations in order to avoid information leaking into the model. The observations from the first three months, May, June and July, is assigned to the training data set and August will be used as validation. The data sets only have records which depend on the observations made on the hour. This limitation is due to the temporal resolution of the HARMONIE model. Therefore the time lag  $\tau$  is set to 60 minutes. The training data set has 355 observations (14.8 days) and the validation data set has 309 observations (12.9 days).



# 4

## Assessment of radar and HARMONIE forecasts

### 4.1. Performance for selected events

As mentioned in chapter 3, there are 28 days in the four months with rainfall. Of these 28 days three events were selected to give a sense of how good the radar nowcasts and HARMONIE perform. These three events either have a growing, decaying or little to no growth or decay behaviour. The differences between the radar nowcasts and HARMONIE will also become more apparent by showing these events.

#### 4.1.1. Event 1 - Growth

The first selected event occurred on 2019-06-10. Figure 4.1 shows a time series of radar images of this event. A rain cell passes through area  $A^{60}$  from the south to the north side. It is clearly noticeable that the area of the rain cell gets smaller as the southern part is vanishing. At the same time, the front of the rain cell is growing both in intensity and in size. The Lagrangian extrapolation method does not account for this growth and decay, causing the radar nowcast being off in time. The combination of growth and decay of this rain cell results in an artificial acceleration. The rain cell passes through the buffer zone earlier than is predicted by the KNMI, due to this artificial acceleration. Figure 4.2 shows the nowcast product from the KNMI for this event. At 13:40 the rain cell has reached the buffer zone, but the nowcasts shows this happening at 14:00. The radar nowcast lags approximately 20 mins behind. Also, the area of the rain cell inside the buffer zone is bigger in the nowcast at 15:00 than in the observations. The average predicted precipitation rate in the area will therefore be overestimated. The nowcasts for this time frame are therefore both off in timing and average intensity.

The time series of the HARMONIE forecasts are shown in figure 4.3. HARMONIE has a harder time in predicting the rain cell. The rain cell is predicted to pass through the area at 16:00, which is almost two hours too late. However, it is most likely that the rain cell is predicted at the wrong location. At 14:00 HARMONIE predicts a rain cell south of  $A^{60}$  and continues moving north. These inaccuracies result in HARMONIE performing bad for this event, unless the HARMONIE is corrected for these discrepancies in time and space.

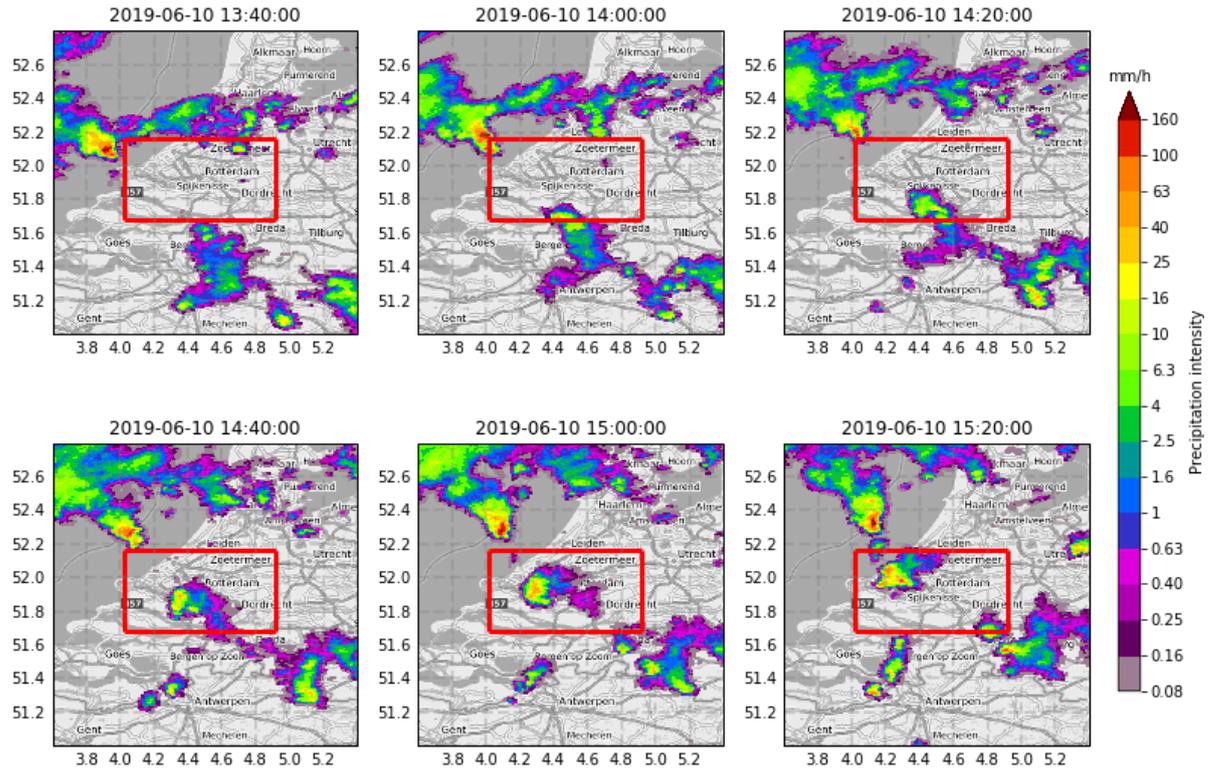


Figure 4.1: Time series of a rain cell passing the buffer zone  $A^{60}$  on 2019-06-10 with intervals of 20 minutes.

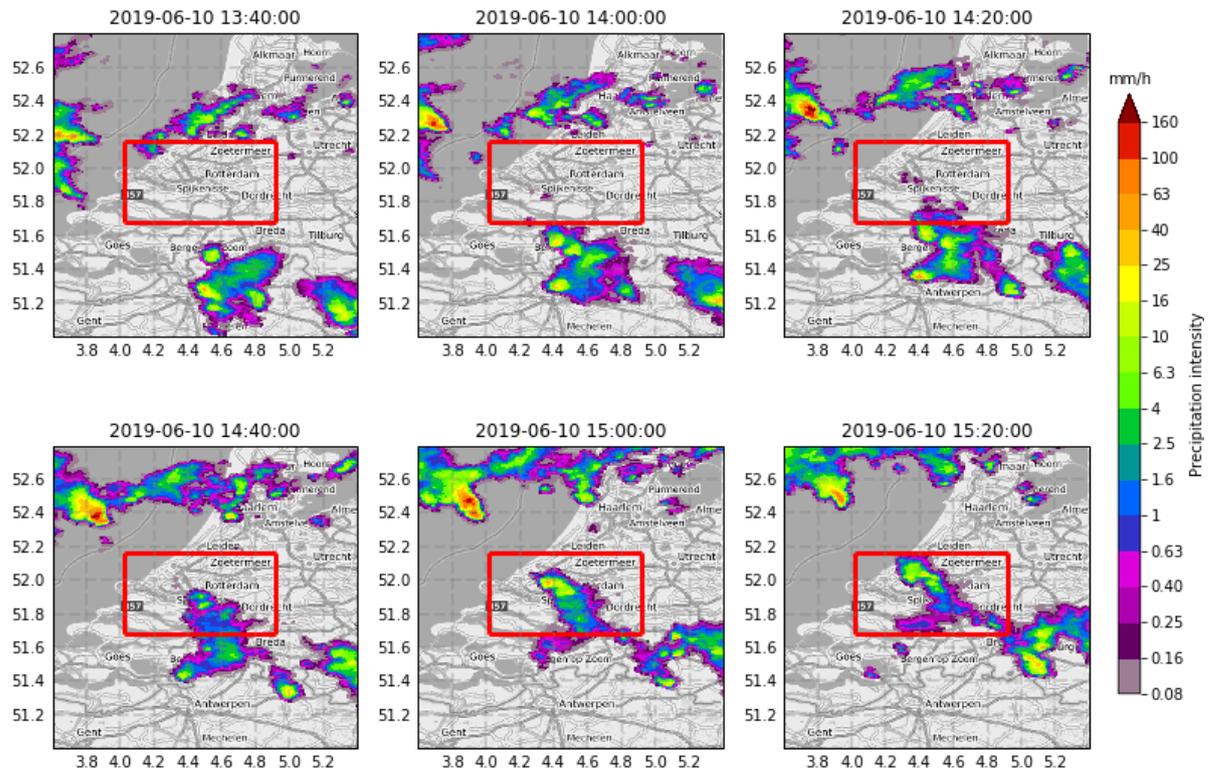


Figure 4.2: Time series of the nowcast images on a rain cell passing the buffer zone  $A^{60}$  on 2019-06-10 with intervals of 20 minutes. The lead time is one hour.

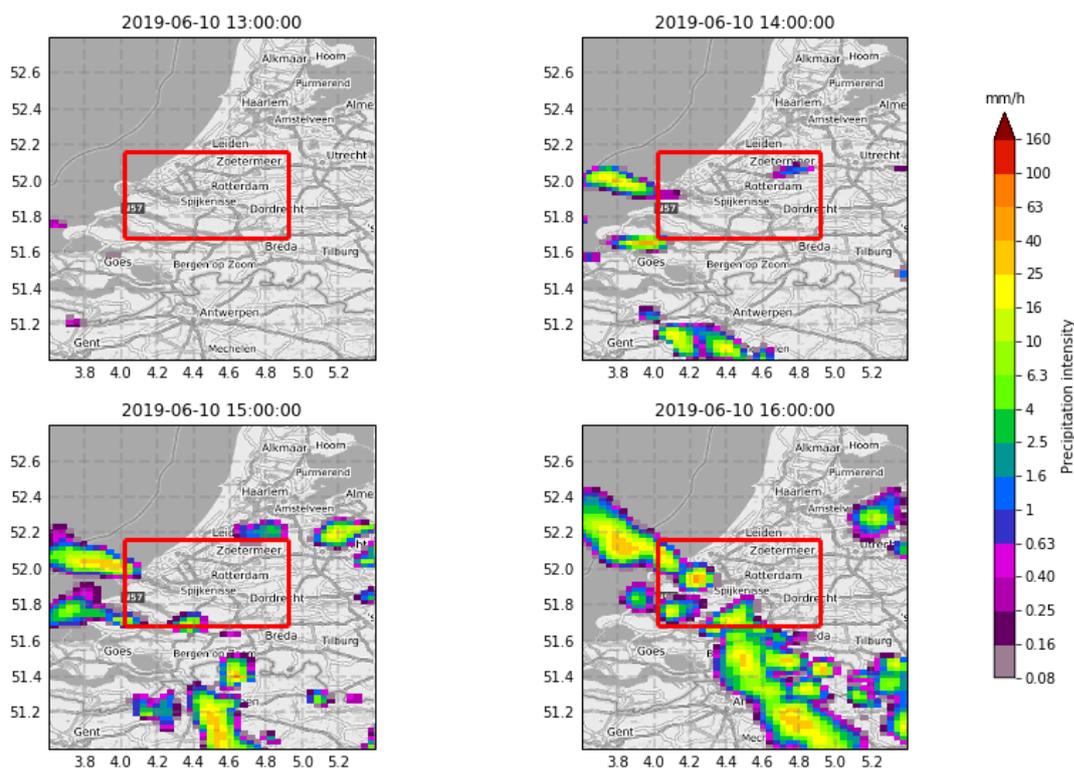


Figure 4.3: Time series of the HARMONIE forecasts on a rain cell passing the buffer zone  $A^{60}$  on 2019-06-10.

#### 4.1.2. Event 2 - No growth or decay

Event 2 is an event for which the radar nowcast is very similar to the observed radar images. This event took place on 2019-07-27 and is shown in Figure 4.4. At first, between 03:40 and 04:00, the rain cell grows rapidly. The rain cell then stops growing significantly from 04:00 and onward. Because the rain cell does not change much in intensity or size, we would expect that the radar nowcast will be close to the real truth. Figure 4.5 shows the radar nowcast with a lead time of one hour. The nowcast is comparable to the radar observations as there is little change in the shape of the rain cell.

HARMONIE, on the other hand, is clearly wrong in predicting the rain cell. Figure 4.6 shows the prediction of HARMONIE and it looks like the rain is not predicted at all. However the rain could also be predicted at the wrong location. HARMONIE predicted rain south of the Netherlands and could be predicted at the wrong location and actually correspond to the rain cell passing over Rotterdam. This proves that HARMONIE is not always reliable.

#### 4.1.3. Event 3 - Decay

The third event is shown in Figure 4.7 and is an example of a rain cell decay/vanishing. Two rain cells are present near the buffer zone at 08:00 and move towards the northeast direction. One rain cell starts decaying until it is vanished completely at around 08:40. The other rain cell does not change very much. As mentioned, the nowcast does not take any growth or decay into account and thus still shows the vanished rain cell in the prediction for 09:00, see Figure 4.8. According to the prediction the rain cells will also take longer to leave the area. Just like event 1, the decaying leads to the nowcast being off in timing and average precipitation rate.

Figure 4.9 shows that HARMONIE predicts one rain cell instead of two at 08:00. This rain cell is decaying and disappears at 09:00. This might be caused by the lower resolution of HARMONIE and also the inability of HARMONIE to predict small rain cells. Also, it could be that only the vanishing rain cell is predicted. But, on the bright side, the dynamical part of this event is predicted correctly by HARMONIE and could be used for improving the radar nowcast.

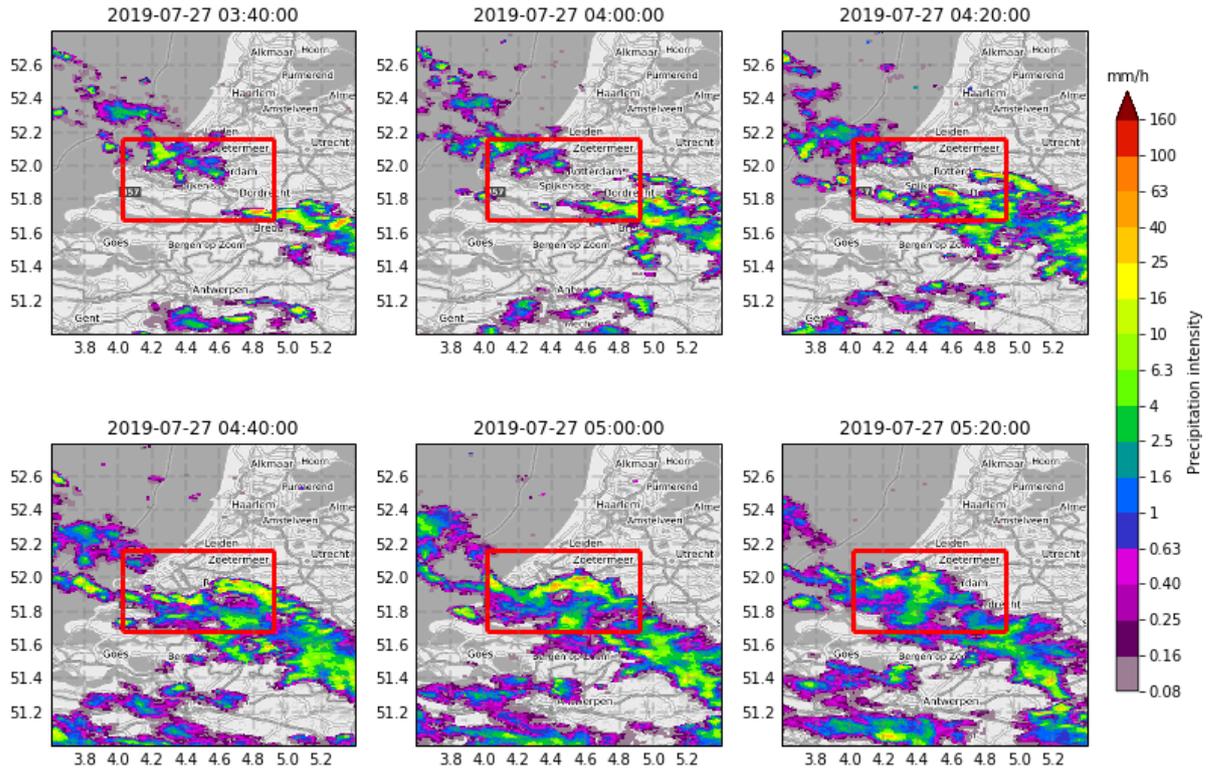


Figure 4.4: Time series of a rain cell passing the buffer zone  $A^{60}$  on 2019-07-27 with intervals of 20 minutes.

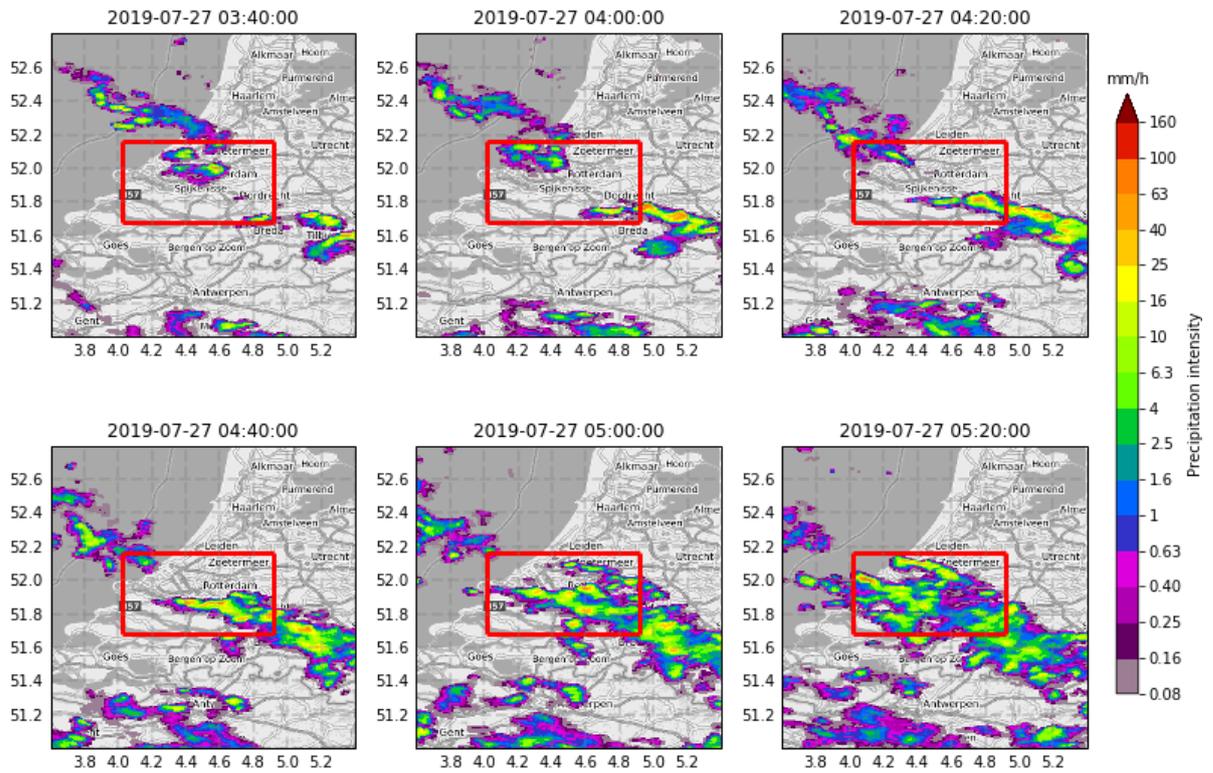


Figure 4.5: Time series of the nowcast images on a rain cell passing the buffer zone  $A^{60}$  on 2019-07-27 with intervals of 20 minutes. The lead time is one hour.

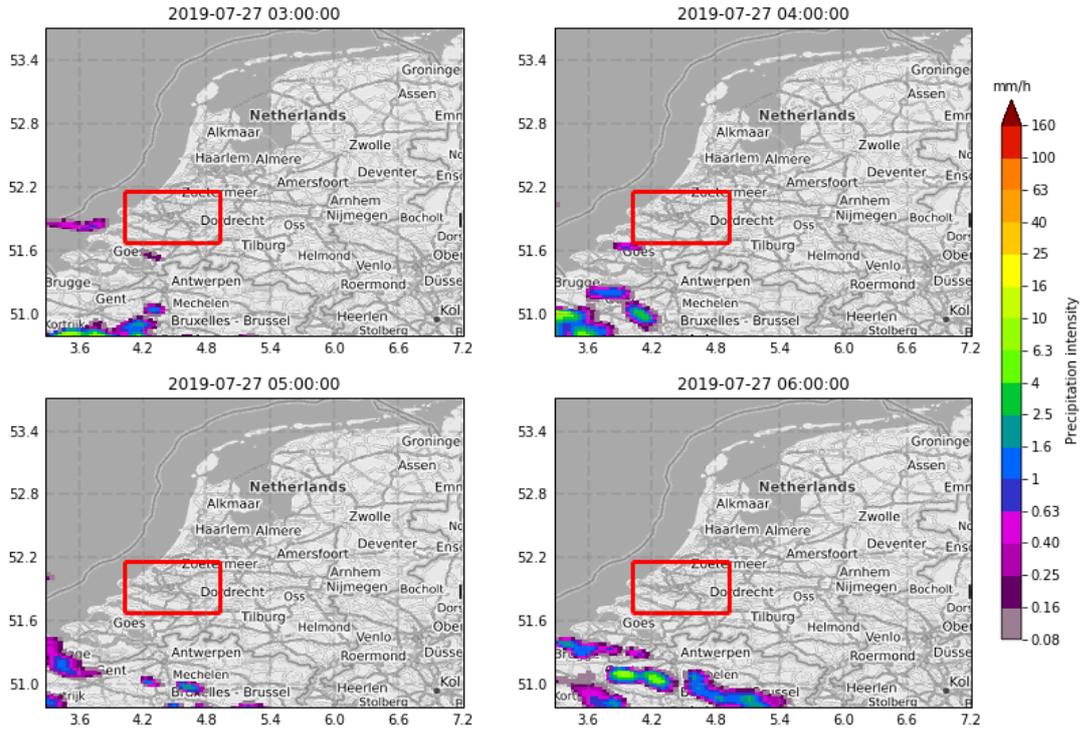


Figure 4.6: Time series of the HARMONIE forecasts on a rain cell passing the buffer zone  $A^{60}$  on 2019-07-27.

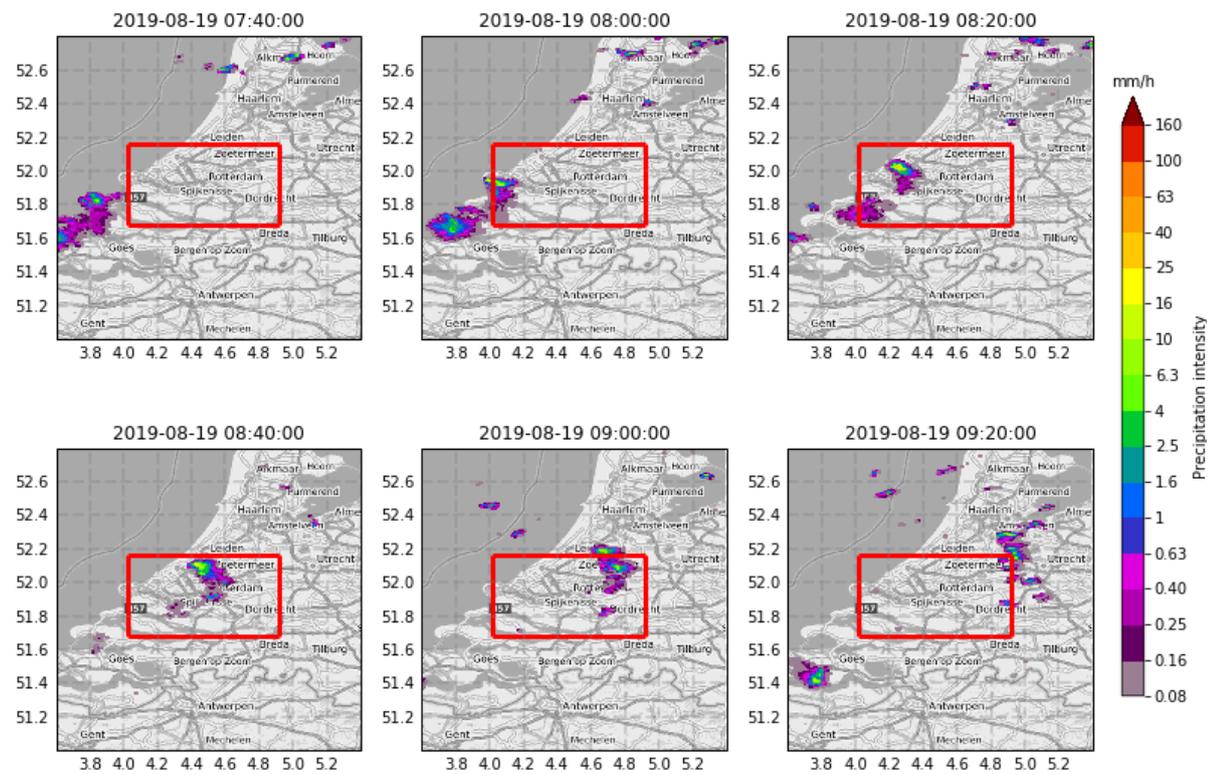


Figure 4.7: Time series of a rain cell passing the buffer zone  $A^{60}$  on 2019-08-19 with intervals of 20 minutes.

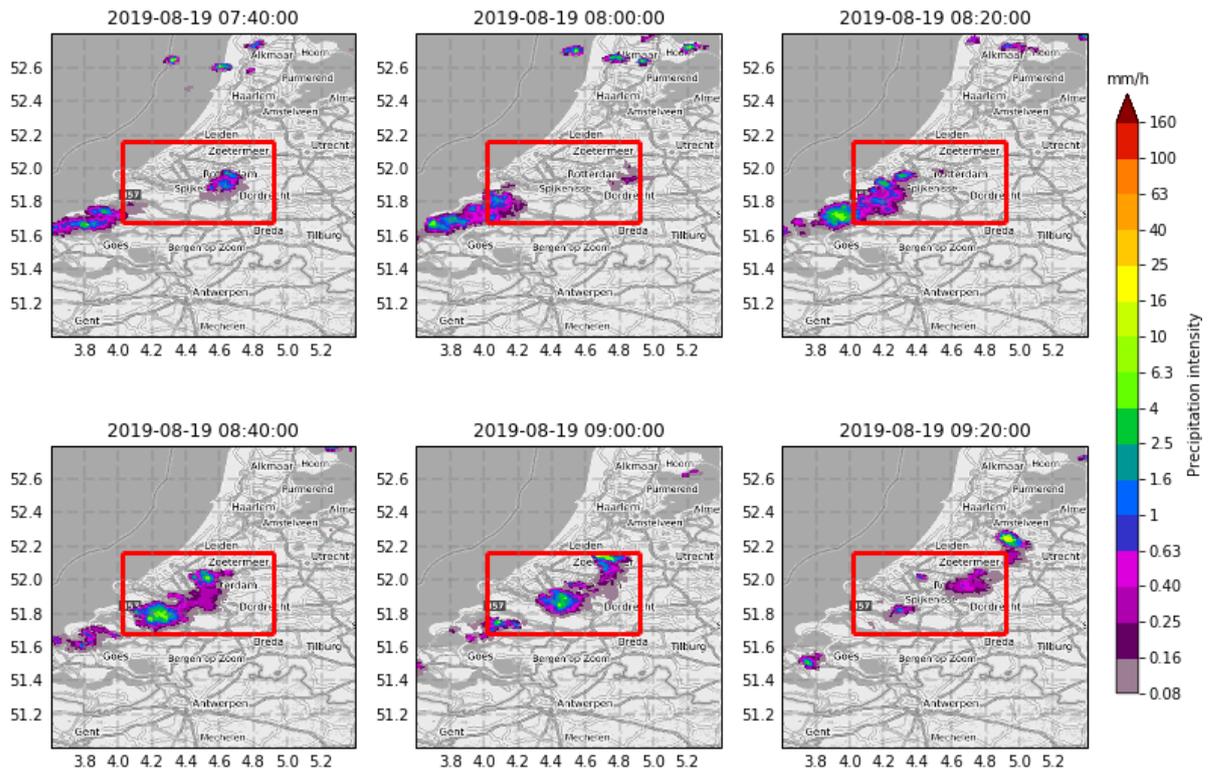


Figure 4.8: Time series of the nowcast images on a rain cell passing the buffer zone  $A^{60}$  on 2019-08-19 with intervals of 20 minutes. The lead time is one hour.

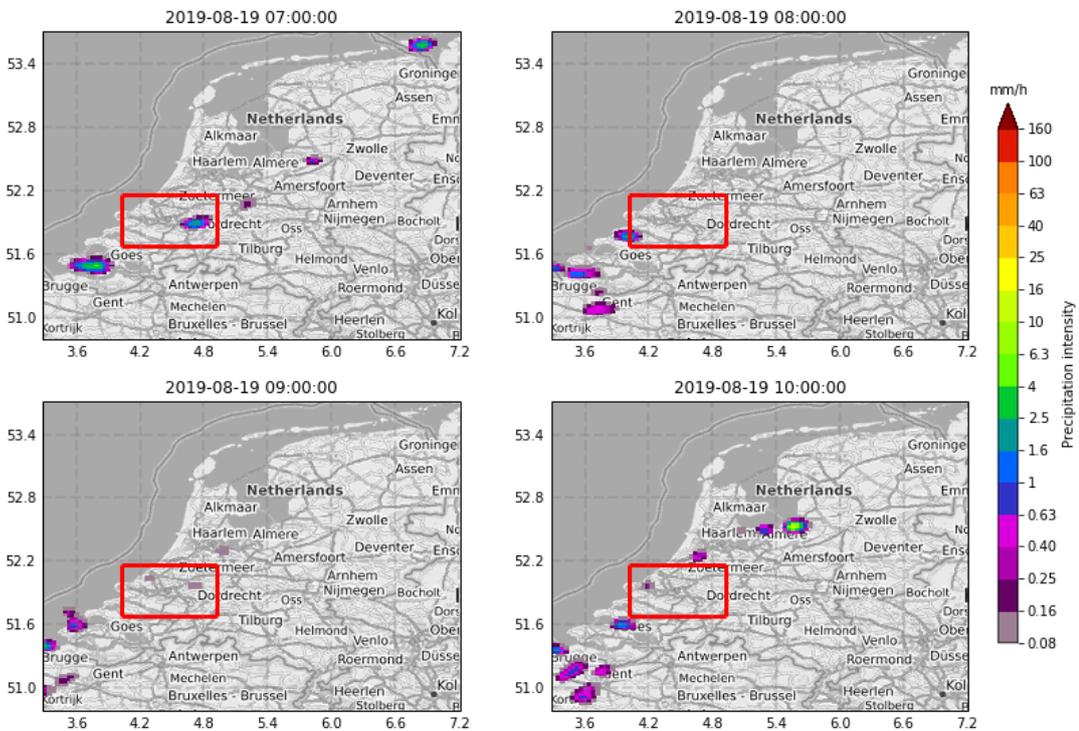


Figure 4.9: Time series of the HARMONIE forecasts on a rain cell passing the buffer zone  $A^{60}$  on 2019-08-19.

## 4.2. General performance

The radar nowcasts and HARMONIE predictions are both not perfect. The three events shown before proved that the performance of both product depend on the characteristics of the rain cell itself. The growing or decaying part of a rain cell is not captured with the radar nowcasting method, but the position of the rain cell is still predicted very well. This is in contrast with HARMONIE. HARMONIE is capable of predicting growth and decay of rain cells, but has a hard time predicting the position of the rain cell. There is a trade off between the accuracy of position/timing and dynamical behaviour.

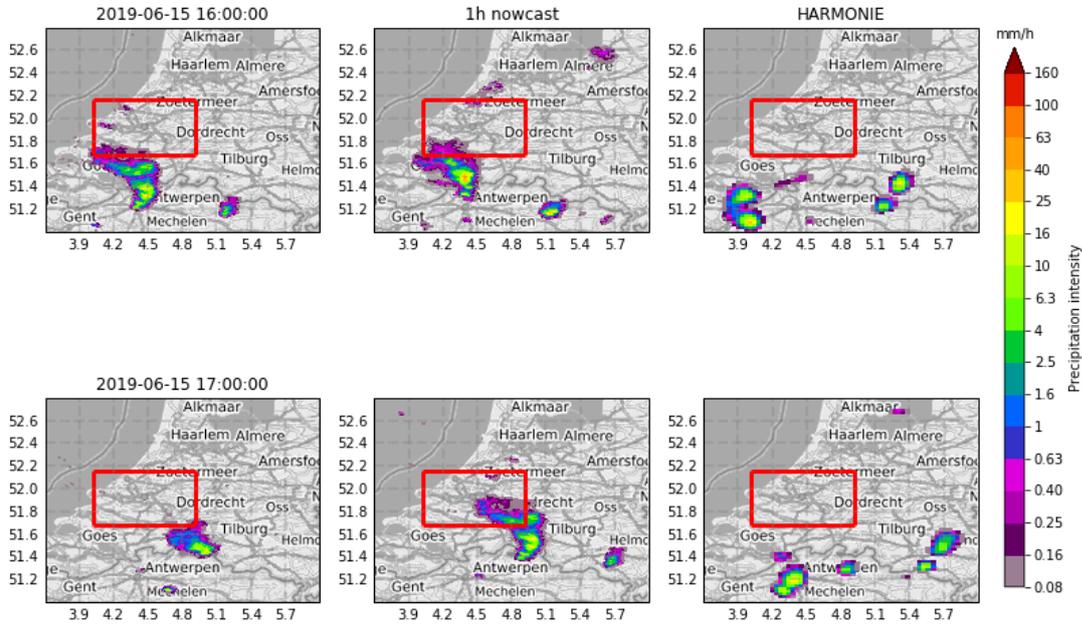


Figure 4.10: From left to right, the observed precipitation rate, predicted precipitation rate from the radar nowcast with 1 hour lead time and predicted precipitation rate from HARMONIE respectively.

The difference between the KNMI products can be clearly seen in Figure 4.10. HARMONIE captures the decay of the rain cell very well and the radar nowcasts predicts the rain cell at the right location even though the prediction does not account for the decay of the rain cell. The performance of these products are therefore assessed in terms of mean squared error (MSE), mean biased error (MBE) and correlation. The performance metrics are defined in chapter 3.4.2. The performance metrics are computed over the average precipitation rate  $\bar{P}$  over area  $A^{60}$  and  $A^{20}$ . The performance of HARMONIE is only computed for area  $A^{60}$ , due to the low resolution and HARMONIE not predicting the rain cells at the right location.

Table 4.1: The general performance of the radar nowcasts and HARMONIE. The lead time of the radar nowcasts is one hour and is computed on the entire data set on an hourly resolution.

	Radar nowcast averaged over $A^{60}$	Radar nowcast averaged over $A^{20}$	Harmonie averaged over $A^{60}$
MSE [ $mm^2/h^2$ ]	0.100	0.621	0.858
MBE [ $mm/h$ ]	-0.009	0.004	0.008
$\rho$ [-]	0.850	0.564	0.345

Table 4.1 shows how well both products perform. The MSE over area  $A^{60}$  is lower than for area  $A^{20}$ . Going from a larger area to a smaller area means that predicting the rain cell at the correct position will become more important and thus the MSE will be higher. Also, the dynamical behaviour will become more important for smaller areas as the average is computed over a smaller area. A smaller area means less pixels that are getting averaged and thus the growth and decay will become more apparent. As expected, HARMONIE is less accurate than the radar nowcast. This is mostly because the rain cells are being predicted at the wrong location or at the wrong time.

### 4.2.1. Performance of radar nowcast with lead time

As mentioned in chapter 1, the nowcasts perform well whenever the dominant source of variability is due to the motion of the rain cells rather than growth and decay. The growth and decay in events 1 and 3 implies that the nowcasts were less accurate than for event 2. Event 2 had little growth, resulting in the nowcast being more accurate. The same conclusion can also be drawn with longer lead times. By increasing the lead time, growth and decay will play a more evident and important role. The accuracy of the nowcasts decreases as a result of longer lead times. To visualize this phenomena the observed average precipitation rate  $\bar{P}^{20}$  of an event is shown together with the predicted average precipitation rate  $\hat{P}_\tau^{20}$  for different lead times  $\tau = 5 \text{ min}$ ,  $60 \text{ min}$  and  $120 \text{ min}$  in Figure 4.11.

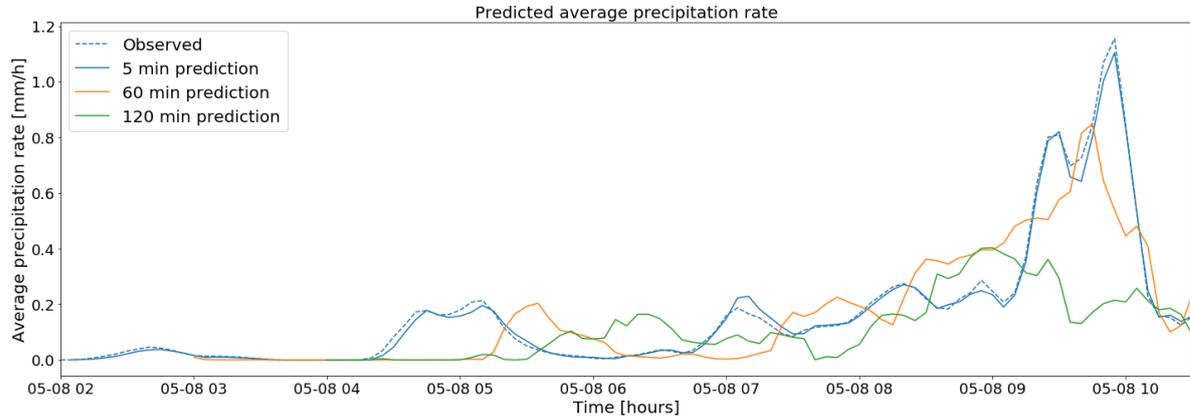


Figure 4.11: The observed average precipitation rate and the predicted average precipitation rate for different lead times from the radar nowcasts over the area  $A^{20}$ .

It is clearly visible that the accuracy decreases when the lead time increases. The predicted average precipitation rate with 5 min lead time follows the observed precipitation rate very well. Increasing this lead time to 60 minutes in a prediction which lags behind of the observation. Also the predicted precipitation rate does not follow the observation very well anymore. Increasing the lead time even further results in a bigger time lag, so the rain cell passes the area almost an hour too late. Also the predicted precipitation rate is lower. The growth of this rain cell is not captured with extrapolation. Therefore this is noticeable in all of the nowcasts and the lag increases with lead time. This is also evident when looking at the MSE in Figure 4.12. As the lead time increases the error gets larger. Up to around 45 minutes the mse shows a linear increase. The linearity disappears after this. This change is probably due to the growth and decay becoming the dominant source of variability. The MSE is even higher when only looking at average precipitation rates above 1 mm/h.

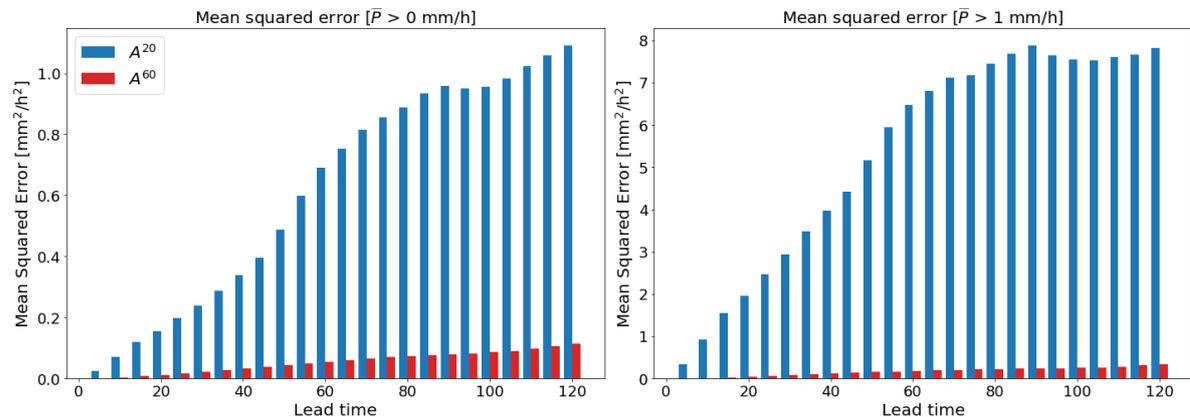


Figure 4.12: The mean squared error between the KNMI nowcasts and observed precipitation rate computed over the entire data set on 5 min temporal resolution. The MSE in the plot on the left is over the whole data set and the MSE in the plot on the right is computed over points with observed average precipitation rate above 1 mm/h. As expected, the error increases with the lead time.

The correlation coefficient is shown in figure 4.13 and shows the same behaviour. This correlation coefficient measures the strength of the linear relationship between two variables and if this is correlation is positive or negative. If two variables are perfectly positive correlated the correlation coefficient will be 1 and if it is perfectly negative the correlation coefficient will be  $-1$ . If there is no correlation the correlation will be 0. The difference between the correlation of the areas  $A^{20}$  and  $A^{60}$  is big. The cause can be found in the difference in size. Averaging over a larger area tends to smooth out the smaller, more localized, rain cells and also the growth and decay of rain cells is less. Surprisingly, the correlation coefficient of  $A^{60} [\bar{P} > 1 \text{ mm/h}]$  seems to not change very little for lead times between 80 and 105 minutes. It is not clear why this happens.

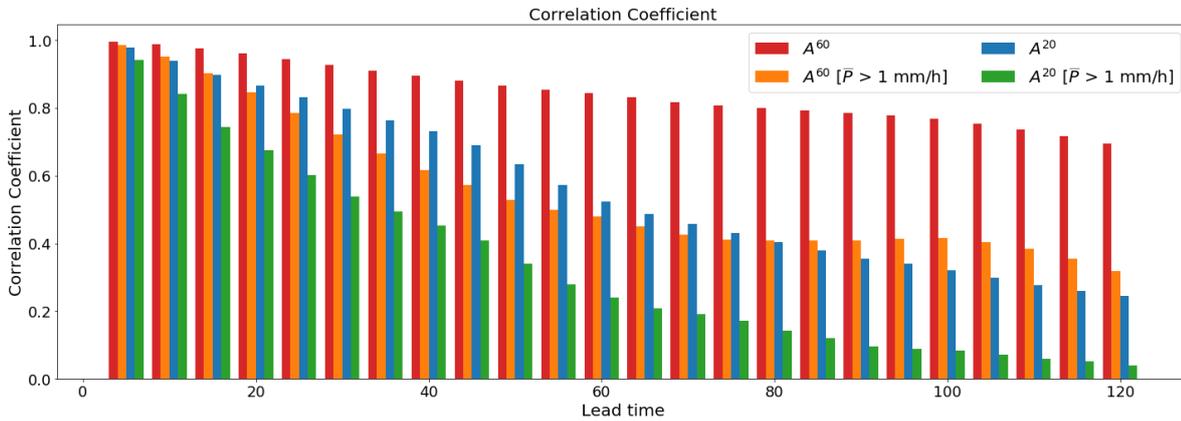


Figure 4.13: The correlation coefficient between the radar nowcasts and observed precipitation rate over the whole data set. As the lead time increases the error gets larger and the correlation decreases.

#### 4.2.2. Limitations of HARMONIE

The estimated rain precipitation rate (INR) from HARMONIE is expected to be a useful parameter. HARMONIE does a decent job at predicting precipitation rate. But its timing and accuracy are far from perfect. Some rain events last much longer than is predicted by the model and on some occasions the rain events are predicted too early or too late. This can go up to 3 hours too early/late. Four events are not predicted at all and when rain is predicted, the average intensity is most of the time an order of magnitude smaller than in the radar. This could be caused by the fact that HARMONIE predicts smaller rain cells and thus has a lower average precipitation rate. Also the grid size of HARMONIE is larger than the radar, causing the rain rates to be smoothed out over a larger area. By working in the logarithmic domain, the change in low average precipitation rate will have more impact.

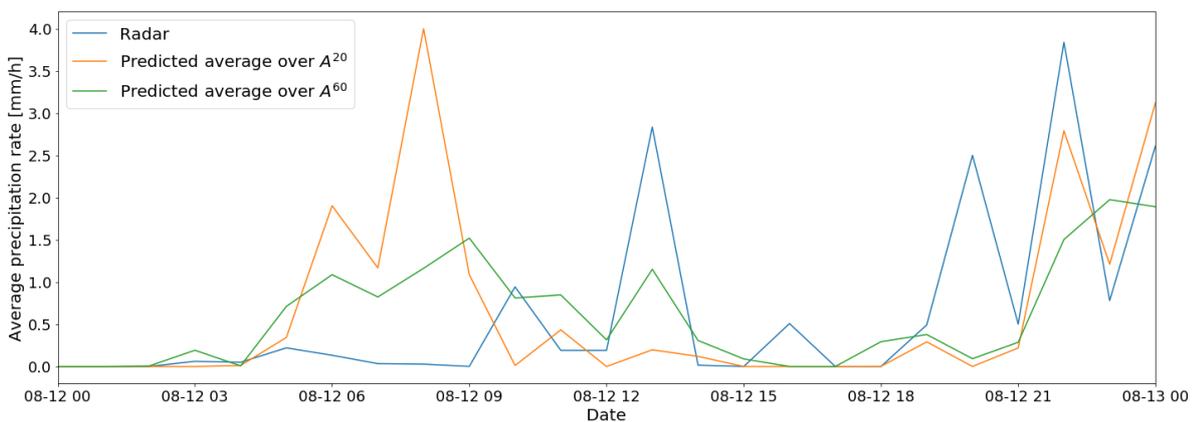


Figure 4.14: The observed average precipitation rate and the predicted average precipitation rate from the HARMONIE model over the areas  $A^{20}$  and  $A^{60}$ .

Figure 4.14 shows the predicted average precipitation rate over the areas  $A^{20}$  and  $A^{60}$ . It looks

like the peak at 08:00 is predicted too early by HARMONIE. At 22:00 the average precipitation rate is predicted correctly. Thus the predictions do not have a constant time lag but this varies over time. HARMONIE also does not always predict the rain cells at the right location. A lot of these time and spatial discrepancies are present in the HARMONIE data.

### 4.3. Correlation of HARMONIE features with precipitation rate

The HARMONIE data set contains a variety of parameters, however not all of them are useful for predicting the growth and decay. Relative humidity (RH) might be an important parameter. At almost every rain event there is an increase in the predicted relative humidity. The trend of this parameter could be a useful parameter for the estimation of the growth and decay. The relative humidity, however, also grows when there is no rain present. The same goes for the cloud cover parameters. The wind speed parameters (U10 and V10) do not seem to contain useful information about the rain events.

One of the drawback of NWP models is that they often lack accuracy in both spatial and temporal domain. For this reason the Pearson's correlation coefficient between the observed rain rate and HARMONIE parameters is calculated. The correlation coefficient between the observed rain rate and HARMONIE parameters can be found in Tables 4.2 and 4.3. The correlation is calculated for both areas  $A^{20}$  and  $A^{60}$ . The highest correlation are from the cloud cover parameters, precipitation rate, cloud base and relative humidity. The correlation also slightly increases going from area  $A^{20}$  to  $A^{60}$ , especially for the precipitation rate. This means that more information about the rain events are found in the area  $A^{60}$ . However, all correlation coefficients remain low. This is caused by HARMONIE predicting rain cells on a different location or at a different time. These bad predictions could introduce errors and noise in the data. Having too much of these errors will lower the performance or accuracy of a machine learning technique.

Table 4.2: Pearson's correlation coefficient between the average precipitation rates  $\bar{p}^{20}$ ,  $\bar{p}^{60}$  and the HARMONIE parameters averaged over area  $A^{20}$ , sorted from high positive correlation to high negative correlation.

	$\bar{p}^{60}$	$\bar{p}^{20}$
Medium Cloud Cover (MCC)	0.39	0.29
Rain precipitation rate (INR)	0.31	0.2
Total cloud cover (TCC)	0.24	0.18
High Cloud Cover (HCC)	0.21	0.14
Relative humidity (RH)	0.16	0.13
Low Cloud Cover (LCC)	0.14	0.12
Boundary layer height (BLH)	0.032	-0.007
U-component maximum gust at 10 m (UWG)	-0.053	-0.057
U-wind component at 10 m (U10)	-0.051	-0.065
Temperature of air at 2 m (2T)	-0.057	-0.066
V-wind component at 10 m (V10)	-0.074	-0.071
V-component maximum gust at 10 m (VWG)	-0.081	-0.084
Cloud base (CLB)	-0.17	-0.14

Table 4.3: Pearson's correlation coefficient between the average precipitation rates  $\bar{p}^{20}$ ,  $\bar{p}^{60}$  and the HARMONIE parameters averaged over area  $A^{60}$ , sorted from high positive correlation to high negative correlation.

	$\bar{p}^{60}$	$\bar{p}^{20}$
Medium Cloud Cover (MCC)	0.4	0.31
Rain precipitation rate (INR)	0.4	0.27
Total cloud cover (TCC)	0.26	0.19
High Cloud Cover (HCC)	0.21	0.14
Relative humidity (RH)	0.17	0.14
Low Cloud Cover (LCC)	0.15	0.12
Boundary layer height (BLH)	0.0091	-0.021
U-component maximum gust at 10 m (UWG)	-0.044	-0.059
Temperature of air at 2 m (2T)	-0.049	-0.062
U-wind component at 10 m (U10)	-0.048	-0.064
V-wind component at 10 m (V10)	-0.067	-0.064
V-component maximum gust at 10 m (VWG)	-0.068	-0.069
Cloud base (CLB)	-0.24	-0.23

#### 4.4. Validation of the backtracking algorithm

Figure 4.15 shows the histogram of the estimated flow direction upwind of area  $A^{60}$  for the observations. According to the backtracking algorithm, most of the precipitation from the radar images comes from the southwest direction and flows towards northeast. This is conform the monthly average wind direction from the KNMI. However, due to the growth and decay of rain cells, the apparent motion of the rain is not always the same as the wind direction. Therefore, some events were visually checked to ensure that the algorithm works correctly.

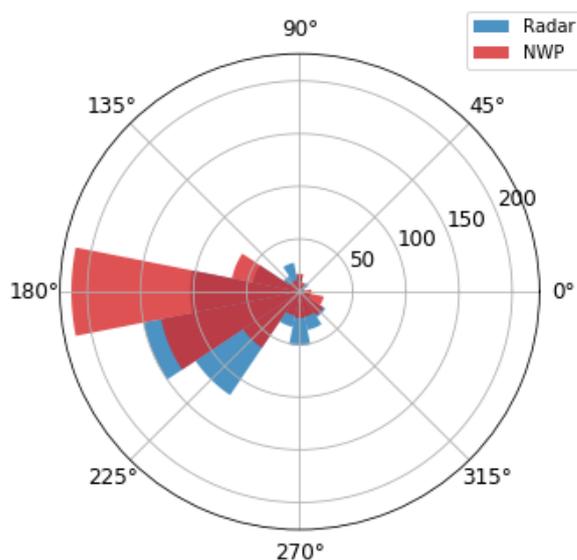


Figure 4.15: Polar histogram of the estimated flow direction. The flow directions are estimated using the optical flow method called Lucas-Kanade.

There are two options to use the backtracking algorithm for HARMONIE. The first option is to use the motion field derived from the radar images. The second is to estimate the motion field from the HARMONIE data itself. As HARMONIE is a model, it does not necessarily produce the same trajectories as the radar images. Figure 4.15 shows the histogram of the estimated flow direction upwind of the buffer area for both radar and HARMONIE. The histogram shows that HARMONIE is slightly rotated, having the most rain cells coming from the west, instead of the southwest. The radar and HARMONIE data set have two different projections and might be causing the rotation. Also the coarser resolution of HARMONIE can have an impact.

To see the difference in results, the growth and decay of the HARMONIE using both motion fields has been computed and has been compared to the radar. Table 4.4 shows the agreement of both radar and HARMONIE. The columns in the table show which motion field is used for backtracking HARMONIE. The radar growth and decay is computed for area  $A^{60}$ , because the used HARMONIE data is averaged over this area. The difference between the two motion fields is not big enough to justify the extra time needed to estimate two motion fields. Aside from the small difference, there were also 20 occasions where the motion field could not be derived from the HARMONIE data. This was mostly caused by the absence of precipitation in the HARMONIE images. Therefore in this thesis, the first option was chosen and the motion field was derived from the radar and applied to HARMONIE.

Table 4.4: Classification of the growth and decay in the area  $A^{60}$  after applying the two different motion fields.

	Radar motion field	HARMONIE motion field
Both grow	22.3%	22.4%
Both decay	14.5%	15.7%
Both no grow/decay	4.1%	3.8%
Radar grows, HARMONIE decays	16.3%	17.0%
Radar decays, HARMONIE grows	12.7%	12.3%
Radar no growth/decay, HARMONIE grows	2.1%	1.8%
Radar no growth/decay, HARMONIE decays	1.5%	2.0%
Radar grows, HARMONIE no growth/decay	16.7%	13.9%
Radar decays, HARMONIE no growth/decay	9.9%	8.1%

## 4.5. Performance regarding growth and decay

The growth and decay derived from the radar images in the Lagrangian framework will be used to evaluate the performance of the neural network models. The extrapolation method from the KNMI is based on Lagrangian persistence, meaning that there is no correction for growth and decay in the nowcasts. To set a baseline for the models the mean squared error (MSE) and mean bias error (MBE) is computed for both training and validation data set, and can be found in Table 4.5. The used error is the derived growth and decay from the radar images itself, as there is no correction applied. Both data sets have a positive MBE meaning there tends to be more growth than decay on average. The MBE computed over the buffer area for both data sets is also positive and is in the same order of magnitude as the MBE of the target area. The validation data set seems to exhibit less growth and decay as it has a smaller MSE than the training data set. The rain events in the validation data set are less intense and lasted longer hence the smaller MSE. As expected, the MSE for both data sets is also lower for the buffer area  $A^{60}$ . A larger area tends to lower the average precipitation as its averaged over a larger area. The influence of the smaller scale convective rain cells on the MSE are reduced due to the larger area. The correlation could not be computed as there is no estimated growth and decay when assuming Lagrangian persistence.

The performance of HARMONIE is also shown in Table 4.5. HARMONIE also shows the same relationship between the training and validation data set. HARMONIE performs worse than the radar nowcasts. This might be caused by predicting the rain at the wrong place and/or time. This is also why the MSE reduces when averaging over a larger area. This is however not enough and could be reduced by dynamically adjusting the position or timing of the rain cell.

Table 4.5: Performance of the KNMI radar nowcasts and HARMONIE in estimating growth and decay for 1h lead time.

	Radar nowcasts		HARMONIE	
	Training	Validation	Training	Validation
$A^{20}$				
MSE [ $dBZ^2$ ]	109.137	58.463	159.161	107.57
MBE [ $dBZ$ ]	0.886	1.090	0.707	0.562
$\rho$ [-]	-	-	0.012	0.076
$A^{60}$				
MSE [ $dBZ^2$ ]	28.173	14.164	78.463	61.313
MBE [ $dBZ$ ]	0.583	0.724	0.405	0.196
$\rho$ [-]	-	-	0.022	0.197

### 4.5.1. Lagrangian versus Eulerian framework

As mentioned in chapter 3, the rain over the area is backtracked to its original position at time  $t - 60$ . This is also called Lagrangian framework where a parcel is followed along the trajectory. There exists another framework called the Eulerian framework. In this framework the area is fixed and the change in precipitation is only computed over time. The histogram in Figure 4.16 shows the distribution of the growth and decay for the two frameworks inside the target area  $A^{20}$ . The Eulerian framework has a larger spread compared with the Lagrangian framework. This was expected because the chance of rain cells moving in or out of the area in the Eulerian framework is greater. By using the Lagrangian framework, this chance is minimized but this phenomena can still occur because of imperfections in the optical flow method. Also less growth/decay is observed, hence the frequency around zero is much higher for the Lagrangian framework. The scatterplot found in Figure 4.16 confirms these conclusions. This scatterplot also shows that most of the time both frameworks agree on whether the average precipitation is intensifying or weakening. But there are still cases where both frameworks disagree with each other, i.e. Eulerian framework shows growth whereas Lagrangian framework shows decay and vice versa. It is therefore important to backtrack the rain over the area to its original position. This will reduce the possibility of rain cells moving in or out of the area and thus also reduces extreme values in the growth and decay in the area. When considering a bigger area, for example area  $A^{60}$ , there is less chance that the rain cells will move in or out of the area. The Eulerian framework will be more consistent to the Lagrangian framework.

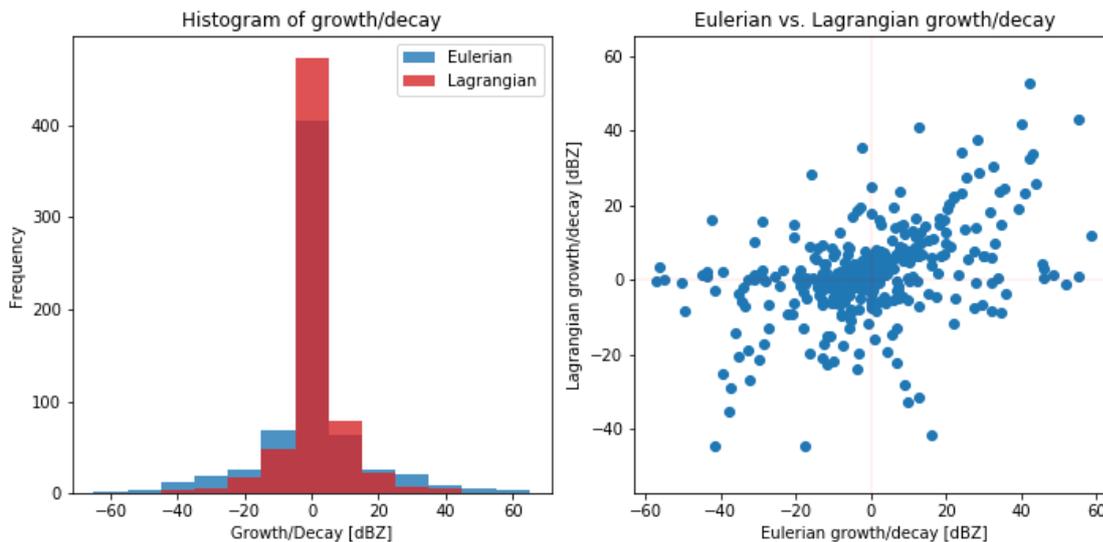


Figure 4.16: Left: Histogram of the growth and decay for the Eulerian and Lagrangian frameworks. Right: Scatterplot of the growth and decay in the Lagrangian framework against the growth and decay in the Eulerian framework. The growth and decay shown is averaged over area  $A^{20}$ .

#### 4.5.2. Comparison between radar and HARMONIE

Table 4.4 shows that even with backtracking, the agreement of HARMONIE with radar is not good. There are more cases where they disagree with each other than agree. Figure 4.17 shows the average radar growth and decay over  $A^{20}$  as a function of the average HARMONIE growth and decay over  $A^{60}$ . There seems to be no correlation between the radar and the HARMONIE growth and decay. The average HARMONIE growth and decay over  $A^{20}$  is not shown but this is even worse. This suggests that the growth and decay inferred from HARMONIE will not be a good predictor for the neural network model. Other more elaborate ways of using HARMONIE should be explored, e.g. a dynamic model that adapts for local shifts in space and/or time or a model that takes into consideration how good the previous predictions of the HARMONIE were.

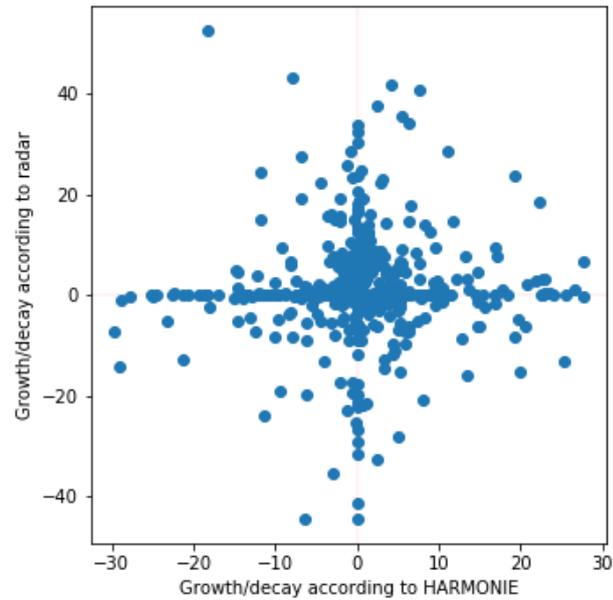


Figure 4.17: Scatterplot between the radar growth and decay and HARMONIE derived growth and decay for area  $A^{20}$ .

# 5

## Static model

### 5.1. Model formulation

The first model is based on the assumption that all of the predictions from HARMONIE are correct, i.e. it trusts all of the predictions. The first model is a simple static model that only uses the three HARMONIE features with the largest correlations. The first feature is the precipitation rate (INR). First, this feature is transformed to dB using Eq. 5.1. To avoid  $\log_{10}(0)$  a small constant ( $5 * 10^{-4}$ ) is added to the precipitation rate. This transformation is done to have the same scaling as the radar data. It is then normalized using Eq. 5.2 to scale the values to the range  $[0, 1]$ . This minimum and maximum value from the training data set are used and also used to normalize the values from the validation data set. The two other features are medium cloud cover (MCC) and relative humidity (RH). These have not been transformed and already have a range between 0 and 1. The features are averaged over the buffer area  $A^{60}$ .

$$\overline{inr} = 10 * \log_{10}(\overline{INR} * 3600 + 5 * 10^{-4}) \quad (5.1)$$

$$\overline{inr}' = \frac{\overline{inr} - \overline{inr}_{min}}{\overline{inr}_{max} - \overline{inr}_{min}} \quad (5.2)$$

As we are interested in the growth and decay the changes between  $t - 60$  and  $t$  of the features are computed. The input for the model is therefore:

1.  $\Delta inr = \overline{inr}'_t - \overline{inr}'_{t-60}$
2.  $\Delta MCC = \overline{MCC}_t - \overline{MCC}_{t-60}$
3.  $\Delta RH = \overline{RH}_t - \overline{RH}_{t-60}$

The range of the inputs are  $[-1, 1]$ . This will help the neural network regard the inputs similarly and use all of the inputs. The model has one hidden layer with 15 neurons. Different combinations of the number of neurons and hidden layers were tested but did not yield any improvement. The output of the model represents the average growth and decay over the target area  $A^{20}$ . The model is also trained to estimate the average growth and decay over the buffer area  $A^{60}$ , in order to see what the influence of using a larger area. The entire model is shown in Figure 5.1.

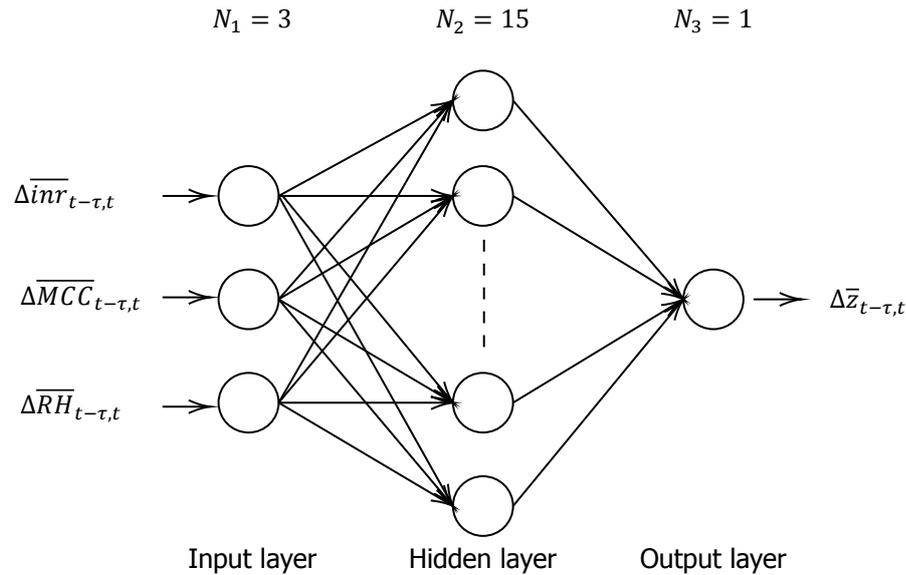


Figure 5.1: The static neural network model

## 5.2. Results for the static model

The static model has been trained ten times with each time a different weight initialization. The MSE, MBE and correlation is computed as the average of these ten runs. This is done to remove the influence of the weight initialization on the performance of the static model. The learning curve of one run is shown in Figure 5.2. Thanks to the early stopping method, the training stops at the fifth epoch after reaching a minimum MSE on the validation data set. This avoids overfitting of the model on the training data. The model, for example, is overfitting the training data after the 70th epoch for this run and training is stopped at the 75 epoch. The performance of the model is shown in Table 5.1 and indicates a small improvement. This improvement is however not big enough to compensate for the computational cost.

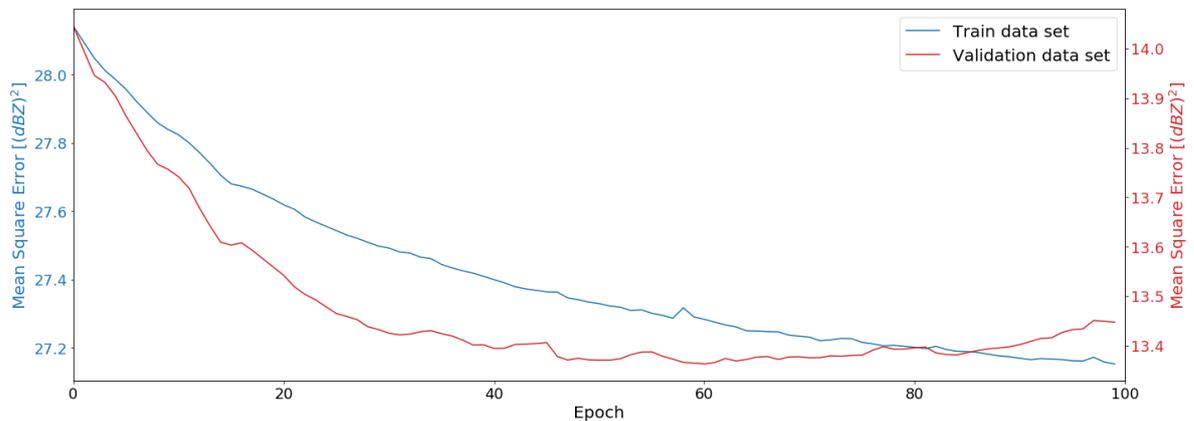
Figure 5.2: The MSE of the training and validation data set after each epoch, i.e. after each update of the weights. The model was trained to estimate the average growth and decay for area  $A^{60}$ .

Figure 5.3 shows the predicted growth and decay  $\Delta\hat{z}_{t-60,t}$  against the derived growth and decay  $\Delta z_{t-60,t}$ . The model is not capable of predicting any growth and decay and seems to only correct for the bias found in the training data set. As shown in Table 5.1, the MSE is slightly lower than of the KNMI radar nowcasts. The low correlation between the predicted and the derived growth and decay is also indicating that the model is not working. When taking a closer look at the input of the model, namely  $\Delta inr$ ,  $\Delta MCC$  and  $\Delta RH$ , it becomes clear why the model behaves this way. Figure 5.3 shows the three scatterplots of the inputs against the growth and decay derived from the buffer area  $A^{60}$ . There is no clear relationship visible between these features and the growth and decay from the radar. The model is therefore not able to learn from the input.

Table 5.1: Performance of the static model on the validation data set in estimating growth and decay for 1h lead time.

	Radar nowcast	HARMONIE	Static model
$A^{60}$			
MSE [ $dBZ^2$ ]	14.164	61.313	13.444
MBE [ $dBZ$ ]	0.724	0.196	0.162
$\rho$ [-]	–	0.197	0.130
$A^{20}$			
MSE [ $dBZ^2$ ]	58.463	107.57	57.093
MBE [ $dBZ$ ]	1.090	0.562	0.269
$\rho$ [-]	–	0.076	0.071

Adding or combining other features from the HARMONIE data set to the model did not improve its performance. The features are not sufficiently related to the growth and decay and thus the model is not able to gain information about the growth and decay. The target area  $A^{20}$  suffers more from this problem as the area is much smaller and HARMONIE being less accurate over a smaller area.

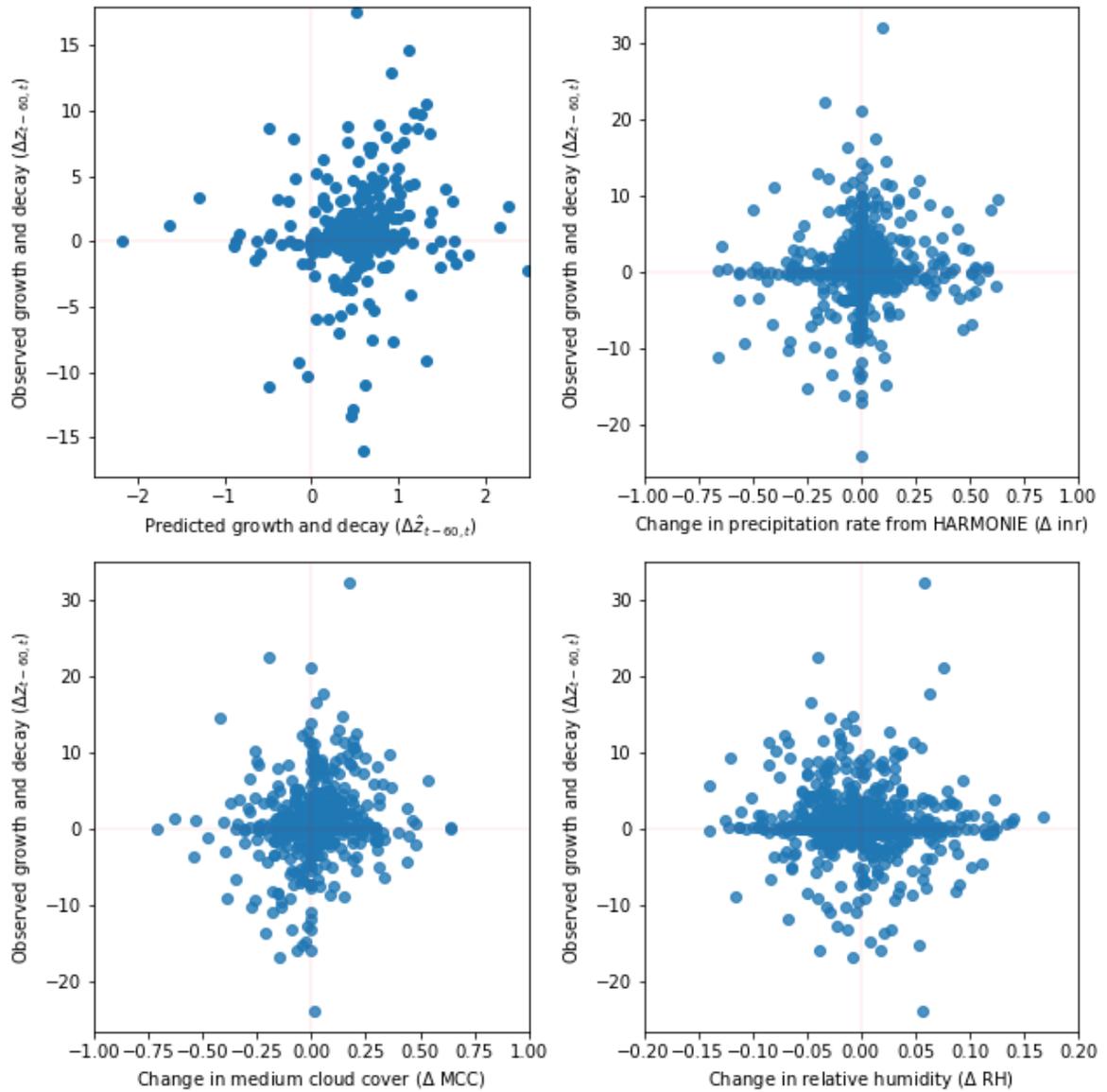


Figure 5.3: Upper left: the predicted growth and decay against the observed growth and decay. Upper right: scatterplot of growth and decay against the growth and decay of HARMONIE. Lower left: scatterplot of growth and decay against the change in medium cloud cover. Lower right: scatterplot of growth and decay against the change in relative humidity. The growth and decay values are in logarithmic reflectivity and all points represent the average over  $A^{60}$ .

The low correlation between HARMONIE and radar growth and decay is caused by giving the model the wrong information for an event. This is mainly caused by three major drawbacks. First, the rain events are often predicted too early or too late. For example, the rain event on 2019-07-26 is predicted two hours too early. The growth and decay of this event is shown Figure 5.4. The time discrepancy of this event results in giving the model the wrong information, i.e. at 18:00 HARMONIE predicts decay but radar observes growth instead. It is clearly visible that HARMONIE shows the same trend but at a different time. This event overlaps two runs of HARMONIE, at 12:00 and 18:00. At 18:00, HARMONIE seems to adjust for this delay and the growth and decay at 22:00 is predicted one hour later. The time discrepancy is thus not constant but changes locally and varies over time.

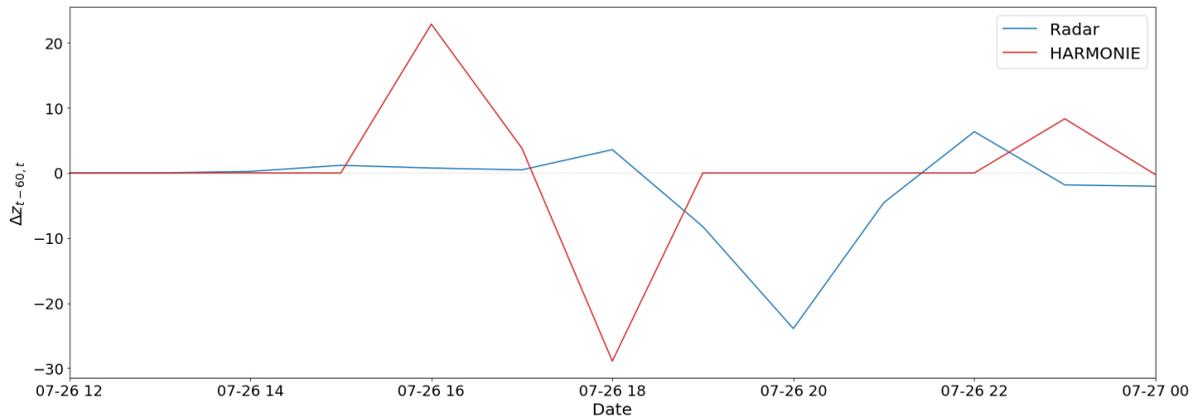


Figure 5.4: The average growth and decay over  $A^{60}$  of a rain event on 2019-07-26. The rain event is not predicted at the right time. Therefore, the growth and decay derived from HARMONIE does not match with the radar growth and decay.

HARMONIE also tends to predict rain cells at the wrong location. Figure 5.5 shows the average growth and decay over  $A^{60}$  of a rain event on 2019-08-01. HARMONIE is predicts over the area, but this should be predicted elsewhere. This leads to using the wrong information as the right information for  $A^{60}$  is found at a different place. There are other events where the rain event should be predicted over the area but is predicted elsewhere. An improvement of the static model could be made by accounting for these two drawbacks of HARMONIE. Applying local shifts in both time and space on the HARMONIE predictions may improve the agreement between radar and HARMONIE.

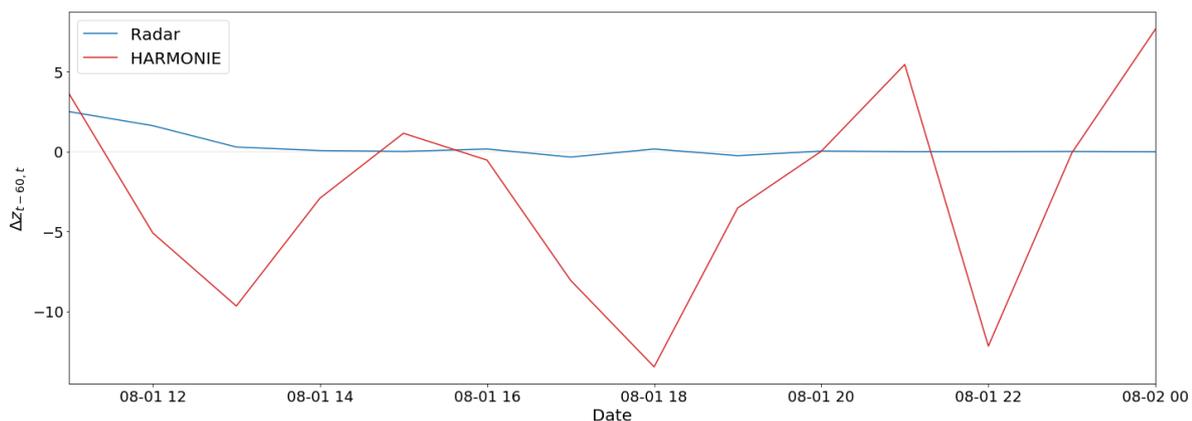


Figure 5.5: The average growth and decay over  $A^{60}$  of a rain event on 2019-08-01. The rain event is not predicted at the right place. Therefore, the growth and decay derived from HARMONIE does not match with the radar growth and decay.

The third drawback has to do with the prediction of a rain cell itself. The predictions from HARMONIE are not perfect. There are cases where rain cells are not predicted at all. Also, even if the rain cells are predicted at the right time and location it does not necessarily mean that the characteristics and shape of the rain cell is predicted correctly. HARMONIE cannot be used to predict the growth and decay for these events. We can therefore conclude that trusting every prediction from HARMONIE affects the

static model in a negative manner. Because of the contradictory data the model is not able to detect and use important features.

# 6

## Dynamic model

The dynamic model aims to improve the quality of the information given to the neural network model, also called the input. The quality of the input can be improved in two ways. The first approach aims to reduce the amount of wrong information given to the neural network. Training the model only on cases where the information from HARMONIE is in line with the radar observation could lead to an easier understanding of the input by the model. This also means that the model is not able to predict the growth and decay for every case. The second approach involves improving the quality of the input by shifting the HARMONIE image to the correct location. The information from HARMONIE could be improved in terms of accuracy by correcting the predictions from HARMONIE. HARMONIE does not always predict the rain cells at the right place and thus provides the wrong information for the neural network. Having the rain cells at the right position could lead to a performance improvement. The first approach is implemented in the dynamic model, restricting the model to only use 'good' cases in which HARMONIE can be trusted.

The dynamic model consists of two stacked neural network models. A visual representation is shown in Figure 6.1. The first neural network model is called the input classifier. The main difference between the static and the dynamic model is that the dynamic model does not predict growth and decay for all cases. Instead the dynamic model first decides whether HARMONIE can be trusted to predict growth and decay or not. This is done with the input classifier. The second neural network model then predicts the growth and decay for the good cases only where we think that HARMONIE can be used to predict growth and decay. The structure of this neural network model resembles the static model and is called the regression model. The same three features found in the static model are used for the regression model with the same amount of layers and nodes (1 hidden layer and 15 nodes), see chapter 5. This allows us to compare the performance of both the static and dynamic model. The dynamic model also acts as a proof of concept of the underlying theory. Due to limited time, no further optimisation of the input classifier and regression model has taken place.

### 6.1. Input classifier

The input classifier is a neural network model which classifies if the information from HARMONIE can be trusted to predict growth and decay. This classifier uses three features. The first two features are the correlation and the count of the number of agreements between HARMONIE and radar inside a moving window of length  $L = 5$  hours. The window length  $L = 5$  is chosen because HARMONIE performs a new data assimilation every 6 hours. The third feature counts the agreement between HARMONIE and radar only for the last two hours. This is based on the assumption that the last two predictions of HARMONIE are more important than the older ones.

The input classifier has one node in the output layer. This node is assigned either a 0 or 1. The value 0 (class 'no') means that the information from HARMONIE is not directly usable to predict growth and decay. A value of 1 (class 'yes') means that the information from HARMONIE was consistent with past trends and can therefore be trusted to predict growth and decay. The input classifier is trained using the training data set. The labels for the training data set are chosen based on the agreement between the growth and decay of HARMONIE and radar. If HARMONIE is predicting growth(decay) and

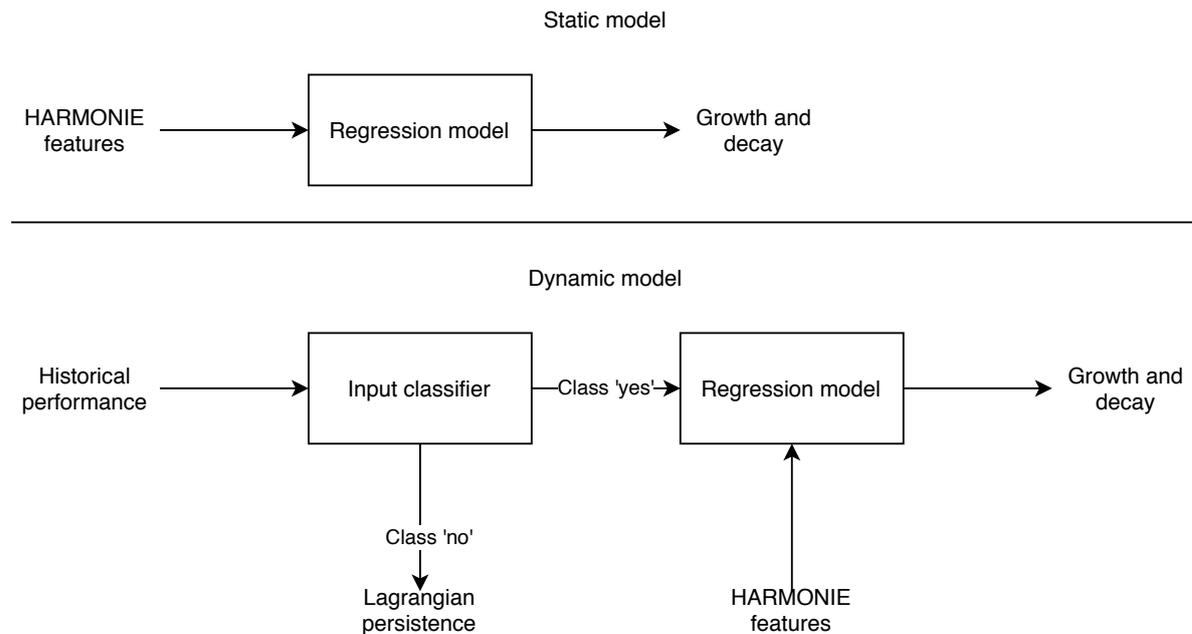


Figure 6.1: Flowchart of the static and dynamic model.

the radar observes growth(decay) then the information from HARMONIE can be used and this case will be assigned to the class 'yes'. When HARMONIE and radar do not agree then this case will be classified to class 'no'. This is visually shown in the left scatterplot of Figure 6.2. Note that the cases for which there is no growth or decay are assigned to class 'no'.

Depending on the outcome of the input classifier, the growth and decay will be predicted or not. If the input classifier outputs class 'no', the growth and decay will be assumed zero. In other words, it is better to assume Lagrangian persistence as the information from HARMONIE is not sufficiently reliable to predict growth and decay. All other cases for which the input classifier predicts 'yes' are given as inputs to the regression model to predict the expected value of the growth/decay. The regression model is therefore trained on a subset of the training data set only, consisting of all cases for which the input classifier predicted class 'yes'.

## 6.2. Results for the dynamic model

### 6.2.1. Input classifier

The input classifier classifies 60% of the cases from the validation data set to the correct class. There is however still room for improvement as the model achieves an accuracy of approximately 70% over the training data set. This applies to both target area  $A^{20}$  and buffer area  $A^{60}$ . The main purpose of the input classifier is to detect bad cases for which HARMONIE cannot be used to predict growth and decay. This minimises the risk of using bad information for predicting growth and decay. Therefore it is important that the false positive (FP) rate is as low as possible. Table 6.1 shows the confusion matrix of the input classifier for area  $A^{60}$ . The input classifier favours classifying most of the cases to class 'no', which is the most common class. Incidentally, this also minimizes the FP. Still, the confusion matrix shows that there is still room for improvement as the false negative (FN = 98) is higher than the true positive (TP = 44). This problem could be addressed by adding other features, for example the MSE over the past 6 hours or by one hot encoding the matches between HARMONIE and the radar instead of using a sliding window method.

The advantage of using an input classifier is that, by removing the cases belonging to class 'no', the correlation between the growth and decay and the features becomes higher. The scatterplot on the right in Figure 6.2 shows the distribution between the growth and decay from HARMONIE and radar of the validation data set. There is more consistency between the radar observations and the HARMONIE features. This means that the regression model will have an easier time figuring out the relationship between the features and actual growth and decay in the radar. Figure 6.2 also shows that the input

Table 6.1: Confusion matrix of the input classifier for area  $A^{60}$ , each row represents the cases in the actual class and each column represents the predicted class for these cases.

		Predicted class	
		No	Yes
True class	No	$TN = 139$	$FP = 27$
	Yes	$FN = 98$	$TP = 44$

classifier allows more growth cases than decay cases. This behaviour is more apparent for the smaller target area  $A^{20}$  and is caused by the larger amount of trusted cases.

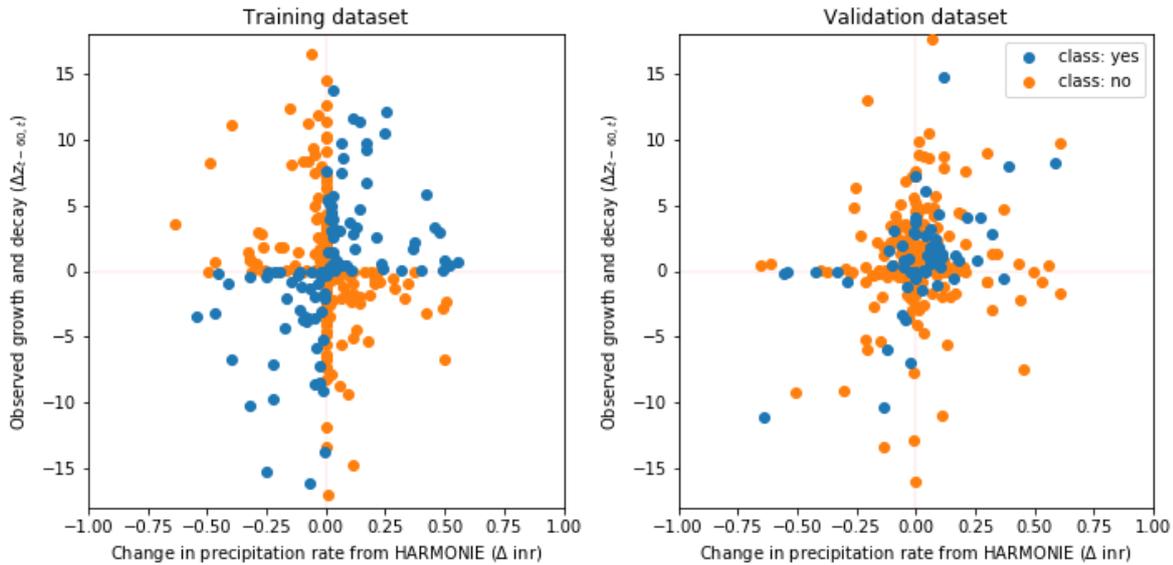


Figure 6.2: Two scatterplots of the growth and decay against the normalized growth and decay of HARMONIE for area  $A^{60}$ . The left scatterplot shows the points which were used to train the input classifier. The right scatterplot shows which points are predicted to be trusted or not by the input classifier.

The target area  $A^{20}$  is a lot smaller than area  $A^{60}$  therefore HARMONIE needs to be even more accurate. This naturally results in fewer cases where HARMONIE can be trusted. The training data set has 270 cases for class 'no' and 85 cases for class 'yes'. This suggests that HARMONIE is not accurate enough (in space and time) to predict growth and decay over a small area of  $20 \times 20$  km. The problem is not the input classifier but the low accuracy of the NWP model at such small scales. A more accurate NWP model would probably result in a better spread over the classes, enabling the regression model to use more information from HARMONIE and also predict the growth and decay for more cases.

### 6.2.2. Performance of the dynamic model

By reducing the amount of unusable information, the quality of the information given to the regression model is much higher. Table 6.2 shows the performance of the static and dynamic model. Overall, the dynamic model performs slightly better than the static model. The MSE decreased by  $0.158 \text{ dBZ}^2$  and also the correlation has increased to 0.234. However the MBE for the dynamic model is greater, meaning that the dynamic model is underestimating the growth and decay. Note that the table shows the performance for all cases, including the rejected cases for which no growth and decay is predicted. This also explains the numbers in Table 6.2. The MBE is higher because the growth and decay is not predicted for all cases resulting in underestimated growth and decay on average.

Table 6.2: Performance of the dynamic model on the validation data set in estimating growth and decay for 1h lead time.

	Radar nowcast	HARMONIE	Static model	Dynamic model
$A^{60}$				
MSE [ $dBZ^2$ ]	14.164	61.313	13.444	13.286
MBE [ $dBZ$ ]	0.724	0.196	0.162	0.594
$\rho$ [-]	–	0.197	0.130	0.234
$A^{20}$				
MSE [ $dBZ^2$ ]	58.463	107.57	57.093	56.081
MBE [ $dBZ$ ]	1.090	0.562	0.269	0.768
$\rho$ [-]	–	0.076	0.071	0.188

To provide more insight into the added value of the dynamic model, the performance metrics are also computed only for the cases classified as ‘yes’. The numbers can be found in Table 6.3. We see that the MSE has decreased by 27% and the correlation between the actual and predicted growth and decay is almost 0.5. This shows that the dynamic model is indeed working. Note that the dynamic model is not optimised and only serves as a proof of concept. The model could be improved by using other combinations of features, external information from satellites or weather stations or other, more accurate numerical weather prediction models.

Table 6.3: Performance of the dynamic model in estimating growth and decay for 1h lead time. The performance metrics are computed only over the cases where the input classifier has predicted that HARMONIE is able to predict growth and decay.

	Radar nowcast	Dynamic model
$A^{60}$		
MSE [ $dBZ^2$ ]	15.845	11.468
MBE [ $dBZ$ ]	0.993	0.364
$\rho$ [-]	–	0.488
$A^{20}$		
MSE [ $dBZ^2$ ]	100.017	77.383
MBE [ $dBZ$ ]	4.652	1.791
$\rho$ [-]	–	0.233

Figure 6.3 shows the predicted growth and decay for the cases where HARMONIE is predicted to be trusted. Taking a closer look at these cases reveals that the dynamic model is indeed effective, but it still needs to be improved. The presence of growth and decay is predicted correctly most of the time but the model lacks accuracy in predicting the actual magnitude of the growth/decay. It should be pointed out that the dynamic model has trouble with predicting little to no growth and decay. A positive feature of the dynamic model is that the extreme cases are also predicted to be extreme. The other scatter plots in Figure 6.3 shows the relationship of the features with the growth and decay. The features relating to the precipitation rate ( $\Delta$  inr) and medium cloud cover ( $\Delta$ MCC) have a good correlation with the growth and decay. The relationship between the growth and decay and these features is strong. Relative humidity ( $\Delta$ RH), on the other hand, does not appear to have a strong correlation with the growth and decay. The range of this features is also a lot smaller compared with the other two features. The usefulness of this feature on the model is therefore smaller. The correlations between the features and the growth and decay for the target area  $A^{20}$  are just a bit lower compared to area  $A^{60}$ .

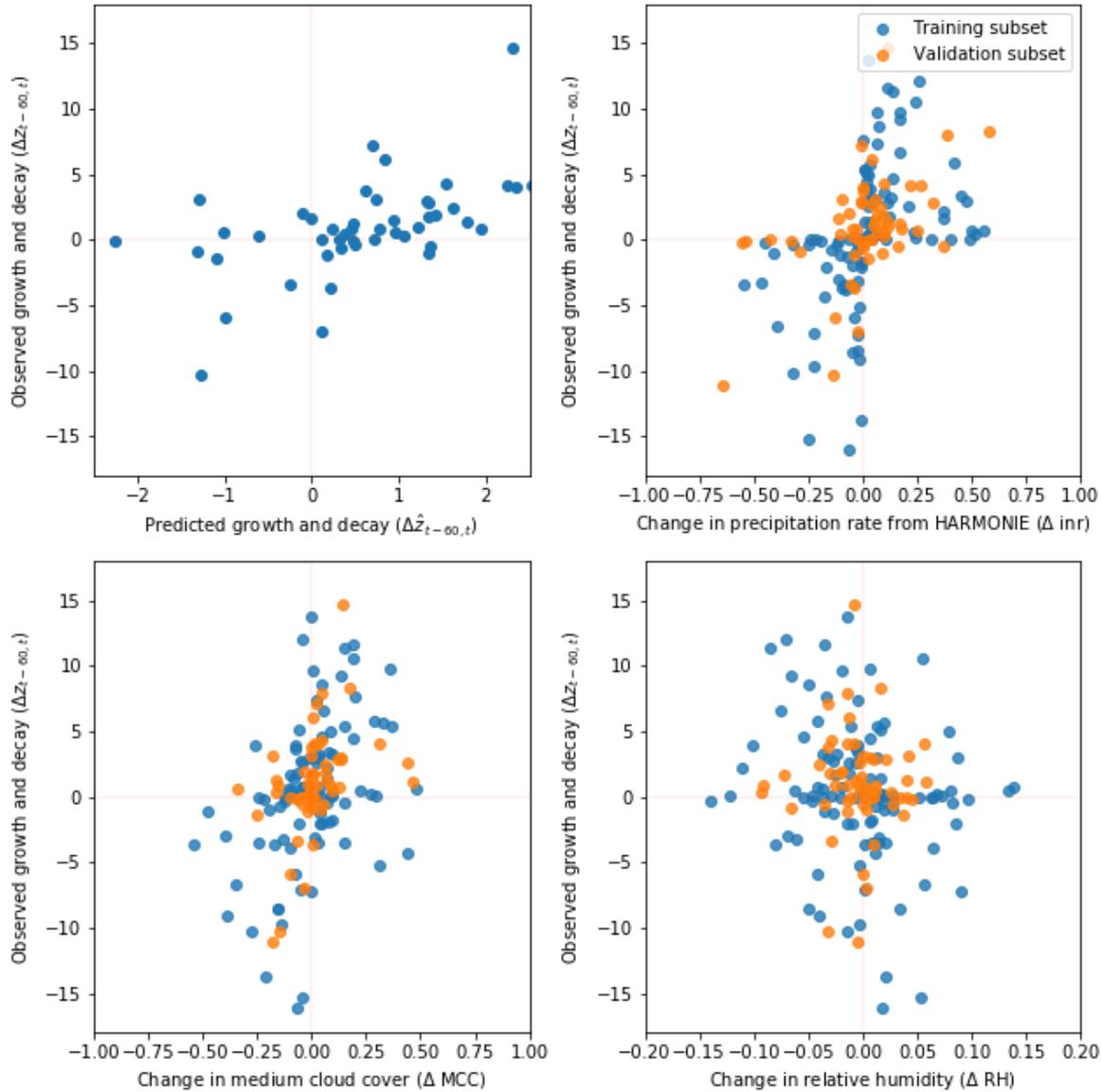


Figure 6.3: Scatterplot showing the predicted growth and decay  $\hat{\Delta z}_{t-60,t}$  against the observed growth and decay  $\Delta z_{t-60,t}$  for area  $A^{60}$ .

### 6.2.3. Performance after applying the shift correction

As mentioned at the beginning of this chapter, there are two ways to improve the prediction: by restricting the application to good cases only or by trying to improve the HARMONIE predictions themselves. Indeed, we know that HARMONIE often predicts rain cells at the wrong location. Aligning the HARMONIE images with the radar images could therefore improve the quality of the predictions from HARMONIE. Figure 6.4 shows the results of applying a global shift correction method to HARMONIE. The shift correction uses a window with length  $L$  to estimate the average shift over the domain. Unfortunately, Figure 6.4 indicates that there seems to be little to no improvement in performance after applying the shift correction. This could be caused by several factors:

- The phase correlation algorithm only accounts for translation shifts. Other transformations like rotation and scaling are not considered.
- The phase correlation algorithm maximises correlation over the whole image and does not consider individual rain cells. The algorithm could therefore propose a shift which does not align the

rain cells over the target or buffer area but still aligns other rain cells well.

- Multiple shifts with similarly large correlations could exist. The algorithm searches for the maximum correlation but another correction with slightly lower correlation might actually result in better performance in terms of growth/decay.
- The predictions from HARMONIE may not be compatible with radar at all. A rain cell could be not predicted or the prediction from HARMONIE does not represent the rain cell well enough and thus still provides the wrong information.

Maximising the correlation for the whole image is thus not recommended. Instead, we recommend to perform local adjustments on the predictions from HARMONIE. These local adjustments should not only be limited to translation shifts, but also other geometric transformation such as rotation, scaling and stretching should also be included. This statement assumes that the rain field in HARMONIE and the radar are already fairly similar. This is not always the case and therefore the input classifier is still needed to distinguish the good cases from the bad.

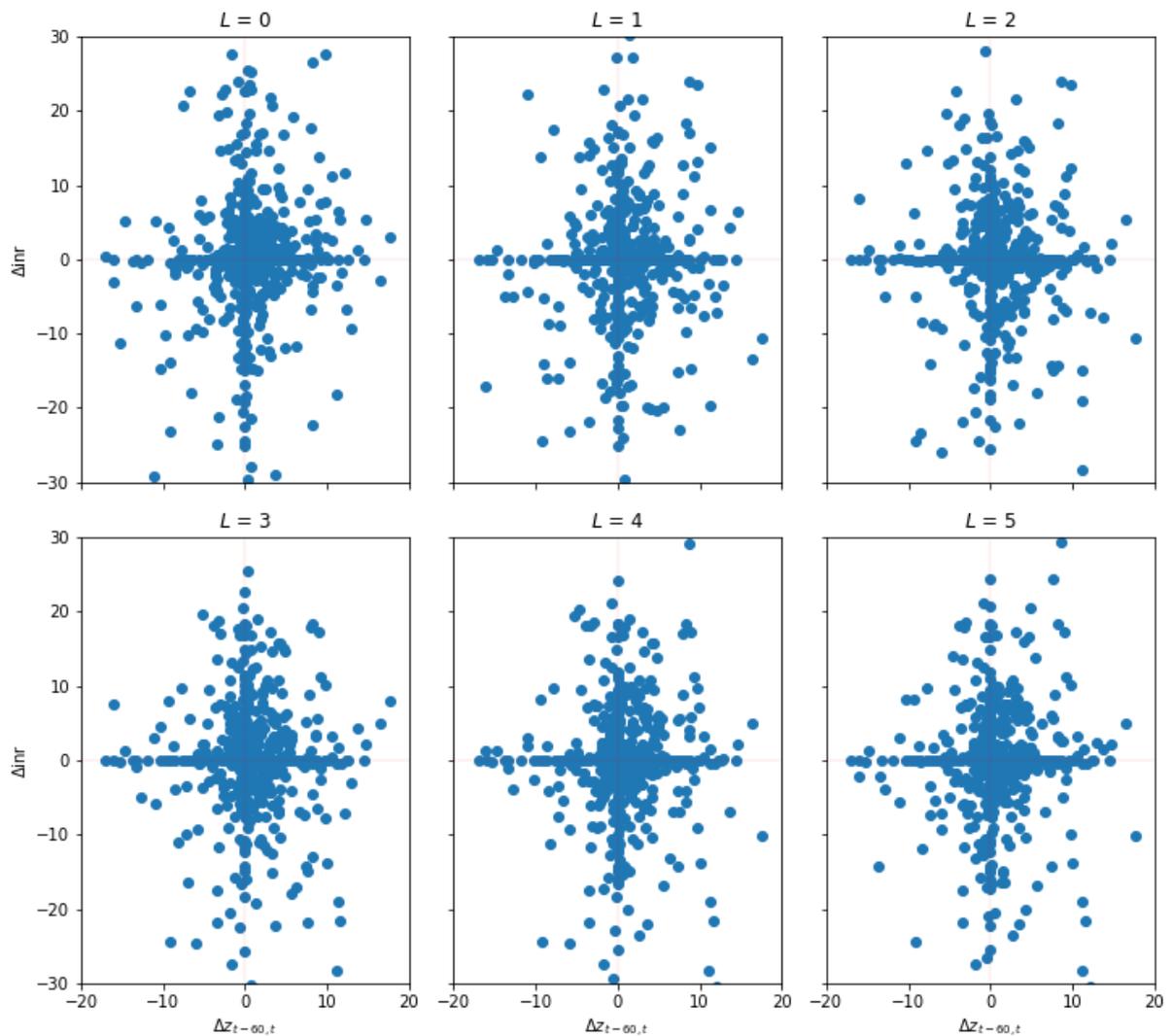


Figure 6.4: Scatterplots showing the HARMONIE growth and decay  $\Delta inr$  against the observed growth and decay  $\Delta z_{t-60,t}$ . Each scatterplot is corrected for a shift which is based on  $L$  previous images. No shift correction is applied on the first scatterplot.

# 7

## Conclusion and recommendations

This thesis has examined the possibility of using information from a numerical weather prediction model, called HARMONIE, to predict the average growth and decay in rainfall over the city of Rotterdam. Predicting the average growth and decay can help improve radar nowcasts, by minimizing the error introduced due to the assumption of Lagrangian persistence. Using a numerical weather prediction model to improve radar nowcasts has not been done before. This is difficult because HARMONIE can not always be trusted when it comes to predicting the exact location, intensity and dynamics of precipitation. To overcome this, a dynamic model is introduced which first predicts whether the prediction from HARMONIE can be trusted. The average growth and decay will be predicted if the information from HARMONIE can be trusted according to the dynamic model. If the dynamic model decides otherwise, it is better to assume Lagrangian persistence as the information from HARMONIE is not sufficiently reliable to predict growth and decay.

### 7.1. Conclusion

Based on the research done and the results the following conclusions are drawn.

- Predicting growth and decay is necessary to increase the accuracy of radar nowcasts. Without growth and decay, radar nowcasts can be both off in timing and intensity. Figure 4.11 shows that the accuracy of the nowcasts decreases as the lead time increases. The growth and decay becomes more important as we increase the lead time.
- The size of the area to predict growth and decay plays an important role. A too large area will result in very little to no growth and decay as the areal average precipitation rate will also be small. Tables 6.2 and 6.3 shows this. The MSE ( $14.164 \text{ dBZ}^2$ ) of area  $A^{60}$  is far lower than the MSE ( $58.463 \text{ dBZ}^2$ ) of area  $A^{20}$ . Also a large area will have less valuable information as users of the nowcasts are only interested in a particular area. A small area means that the predictions from the numerical weather prediction model should be more accurate. There are a total of 308 cases in the validation data set. Of these 308 cases, only 100 cases can be trusted for area  $A^{20}$ . The larger area  $A^{60}$  has 42 more cases which can be trusted.
- The correlation between the HARMONIE parameters and observed precipitation rate is quite low. The maximum correlation was 0.40 and it was between the precipitation rates of HARMONIE and radar. This confirms that the predictions from HARMONIE are not always correct. Even when the two agree in terms of rain rate, the average growth and decay from HARMONIE might still be wrong.
- The static model is based on the assumption that all predictions from HARMONIE are correct and can be trusted. This assumption is wrong, which is reflected by the poor performance of the static model. The model is not able to understand and interpret the information from HARMONIE. There is too much contradictory information and the model does not learn from the features but only from the average statistics of growth and decay. The training data set has more growth

than decay cases and the average growth and decay is 0.724 dBZ. This leads to the static model predicting growth for the majority of the cases.

- The dynamic model is different from the static one in the sense that it assumes that not all predictions from HARMONIE are correct. The dynamic model uses a classifier in order to determine which information can be used to predict growth and decay. The input classifier is able to achieve an accuracy of 60%, but this is mainly achieved by classifying the majority of the information to be not trustworthy. Of the 142 cases where HARMONIE predictions could be trusted, the classifier only found 44 cases and also classified 27 untrustworthy cases as trustworthy.
- Even though the majority of the useful cases are not used, the dynamic model still performs well. The model shows an improvement of almost 1 dBZ<sup>2</sup> over the radar nowcasts. The improvement is even bigger when we only look at the trustworthy cases classified as 'yes' by the input classifier. For those cases, the MSE decreases from 15.845 dBZ<sup>2</sup> to 11.468 dBZ<sup>2</sup>, an improvement of 27%. The input classifier plays therefore a crucial role in the dynamic model.
- The shift correction of HARMONIE did not improve the performance of the dynamic model. This is probably caused by the fact that the phase correlation algorithm tries to maximise the correlation of the entire image, instead of performing local corrections.

## 7.2. Recommendations

While this thesis shows that it is possible to predict growth and decay using numerical weather prediction models, there is still room to better understand and improve the dynamic model. A couple of recommendations for future work are provided below.

- It is necessary to adapt the Lagrangian framework in order to be able to predict growth and decay. The Lagrangian framework enables precipitation parcels to be followed along their trajectories. Allowing the change in precipitation rate and other properties to be monitored for each parcel. The Lagrangian framework also deals with the problem of rain cells coming in or leaving out of the area. The Lagrangian framework can be adapted by estimating a motion field between two consecutive images and using the motion field to compute the original location of each precipitation parcel or rain cell. The displacement terms were computed according to the constant-vector scheme. Better results could be achieved by using the semi-Lagrangian scheme as it accounts for rotation.
- Different combinations of features should be studied. From the three used features only two showed good correlation with the growth and decay. The correlation between relative humidity and growth and decays is very low. Other combinations of features for the regression model could improve the dynamic model even further. Also different features for the input classifier could improve the dynamic model. Not only other features but also other classification and regression techniques could be used to increase the accuracy of the dynamic model.
- The models have been trained with data from May, June and July and are evaluated on data from August. For such a short time span this should not be a problem, but the model could perform worse if used to predict growth and decay for other months, like December. It is recommended to predict growth and decay using a model which is trained with data from the same month or period. Also study has to be done to the performance impact of the dynamic model when using a model which is trained with summer data to predict the growth and decay for other seasons.
- It is also recommended to train and validate the model with a larger data set because a lot of cases are rejected by the input classifier. More cases will give a better representation of the performance of the dynamic model. More cases allows the model also to be more complex and use more features.
- The growth and decay is in units of logarithmic reflectivity. Averaging logarithmic values is not the same as averaging linear values and thus the performance of the dynamic model will not be the same when used with linear values. It is recommended to study the possibilities of using the dynamic model with reflectivity in units of  $mm^6/m^3$  or rainfall rates in units of  $mm/h$ .
- The shift correction method did not yield any significant improvement in the quality of the information. This could be caused by the phase correlation algorithm which maximises the correlation of the images. Maximising correlation does not necessarily mean that the images are aligned. We recommend to study the potential of performing local adjustments on the predictions from HARMONIE and how this affects the amount of trustworthy cases.





# Accessing radar data with python

Opening the KNMI radar files using the python package `h5py` is fairly straightforward and not complicated. First the package needs to be installed. This is done by typing the following command in the command line window:

```
pip install h5py
```

The function shown below can be used to open the radar files. Note that the package has to be imported beforehand.

```
import h5py

def openh5(filename):
    try:
        f = h5py.File(filename, 'r')
        f.close
        return f
    except:
        print('File {:s} cannot be opened'.format(filename))
        return []

f = openh5('RAD_NL25_PCP_FM_1205.h5')
```

Listing 1: Function for opening a radar data file with HDF5 file format.

Once the radar file is successfully opened the information can be accessed by calling the variable `f`. The HDF5 file format works with groups and attributes. The attributes hold the actual information and the groups act like directories on a computer to ensure a well-structured dataset. The radar files have multiple groups<sup>1</sup> but the most important groups is the geographic group and the radar images itself. The radar data files can be accessed through a program called 'HDFView' to quickly see all of the groups and attributes and to also view the images. The geographic group contains all of the information regarding the projection and resolution of the radar file. This group is named 'geographic' inside the HDF5 file format. How to access these attributes is shown in Listing 2.

The radar files have 25 images and each image is stored inside a group named 'image' followed by a number starting from 1 to 25. For example, 'image1' is the radar image at time  $t$  and 'image2' shows the forecast for time  $t + 5$  min and so on. The data of the images are saved as 8 bit integers, ranging from 0 to 255. The function 'convertdBZtoR' shown in Listing 3 converts these to precipitation rate. Listing 3 also shows how to access the images.

---

<sup>1</sup>More information about the radar data files can be found at <https://www.knmi.nl/kennis-en-datacentrum/publicatie/knmi-hdf5-data-format-specification-v3-5>

```

# pixel_size contains the pixel size in kilometers
# for the x and y direction respectively
pixel_size = [f['geographic'].attrs['geo_pixel_size_x'][0]
              , f['geographic'].attrs['geo_pixel_size_y'][0]]

# geo_offset is the offset of pixel (0,0) from origin of projection
geo_offset = [f['geographic'].attrs['geo_column_offset'][0]
              , f['geographic'].attrs['geo_row_offset'][0]]

# image_size contains the number of pixels of the images
image_size = [f['geographic'].attrs['geo_number_columns'][0]
              , f['geographic'].attrs['geo_number_rows'][0]]

# The proj.4 definition, the package pyproj can be used to
# convert coordinates from and to this projection
proj4_params = f['geographic']['map_projection']
               .attrs['projection_proj4_params'].decode('utf-8')

```

Listing 2: Accessing the important geographic information.

```

# Each image can be accessed with the following code
# The images are stored in a 2d array
image1 = f['image1']['image_data'][()]

# A subset of the images can also be accessed
# For example, the pixels in y direction from 200 to 400
# and in x direction from 300 to 500
image1subset = f['image1']['image_data'][()][200:400, 300:500]

# This function converts the images to the precipitation rate
def convertdBZtoR(image):
    # 8 bit to dBZ
    image = np.where(image == 255, 0, 0.5*(image-64.0))
    # dBZ to Z
    image = np.where(image == 0, 0, 10.0**(image/10.0))
    # Z to R
    image = (image/200)**0.625
    return image

# This array contains the precipitation rate for each pixel
image1R = convertdBZtoR(image1)

```

Listing 3: Accessing the images and converting to precipitation rate.

# B

## Accessing HARMONIE data with python

This appendix shows how to access the HARMONIE files with the GRIB file format. Note that this only works with the KNMI HARMONIE 36 version. The HARMONIE files can be accessed through the package `cfrib`. This package needs to be installed with the conda package system. The following command installs `cfrib` on your system.

```
conda install -c conda-forge cfrib
```

Once installed, you need to download the zip file from <https://drive.google.com/file/d/1k8YtTFFLS4FSvo3aVpIF-q65uIK2Hmik/view?usp=sharing> and extract to `PythonInstallFolder\Library\share\eccodes\definitions\grib1\localConcepts`. `PythonInstallFolder` is the directory where python is installed. This step is needed because the HARMONIE files do not adhere to the GRIB standard. In order to open a HARMONIE file the function shown in Listing 4 can be used. The function takes in two arguments. The first argument is the filename and the second is called `level`. `Level` stands for the height above ground in m. Each file has 5 levels, 0, 2, 10, 456 and 457. With each level having different parameters, see Table 2.1. Note that to get the data of all of the parameters the file has to be opened 5 times.

The HARMONIE files have been converted to netCDF files for easier access. How this is done is shown in Listing 5.

```

import xarray as xr

def readgrib(filename, level):
    # Open the file
    ds = xr.open_dataset(filename, engine='cfgrib', backend_kwargs={
        'filter_by_keys': {'typeOfLevel': 'heightAboveGround'
        , 'level': int(level)}})

    # This renames the parameters for easier reading
    if level == 0:
        try:
            ds = ds.rename({'p3111': 'swr'})
        except:
            pass
        try:
            ds = ds.rename({'p3112': 'lwr'})
        except:
            pass
        try:
            ds = ds.rename({'p3117': 'gr'})
        except:
            pass
        try:
            ds = ds.rename({'unknown': 'clb'})
        except:
            pass
        try:
            ds = ds.rename({'p3067': 'blh'})
        except:
            pass
    if level == 456:
        try:
            ds = ds.rename({'p3062': 'ins'})
        except:
            pass
        try:
            ds = ds.rename({'p3063': 'ing'})
        except:
            pass
    if level == 457:
        try:
            ds = ds.rename({'p3062': 'cs'})
        except:
            pass
        try:
            ds = ds.rename({'p3063': 'cg'})
        except:
            pass

```

Listing 4: Function for opening a HARMONIE data file with GRIB file format.

```

ds = readgrib('harm36_v1_ned_surface_2019050218_014_GB', 0)
ds.to_netcdf('harm36_v1_ned_surface_2019050218_014.nc')

```

Listing 5: Function for converting the GRIB file format to netCDF.

# Bibliography

- [1] G. Bruni, R. Reinoso, N. C. van de Giesen, F. H. L. R. Clemens, and J. A. E. ten Veldhuis, *On the sensitivity of urban hydrodynamic modelling to rainfall spatial and temporal resolution*, Hydrology and Earth System Sciences **19**, 691 (2015).
- [2] M. Reyniers, *Quantitative precipitation forecasts based on radar observations: Principles, algorithms and operational systems* (Institut Royal Météorologique de Belgique Brussel, Belgium, 2008).
- [3] K. A. Browning, C. G. Collier, P. R. Larke, P. Menmuir, G. A. Monk, and R. G. Owens, *On the forecasting of frontal rain using a weather radar network*, Monthly Weather Review **110**, 534 (1982), [https://doi.org/10.1175/1520-0493\(1982\)110<0534:OTFOFR>2.0.CO;2](https://doi.org/10.1175/1520-0493(1982)110<0534:OTFOFR>2.0.CO;2) .
- [4] U. Germann and I. Zawadzki, *Scale dependence of the predictability of precipitation from continental radar images. part ii: Probability forecasts*, Journal of Applied Meteorology **43**, 74 (2004), [https://doi.org/10.1175/1520-0450\(2004\)043<0074:SDOTPO>2.0.CO;2](https://doi.org/10.1175/1520-0450(2004)043<0074:SDOTPO>2.0.CO;2) .
- [5] G. Ayzel, M. Heistermann, A. Sorokin, O. Nikitin, and O. Lukyanova, *All convolutional neural networks for radar-based precipitation nowcasting*, Procedia Computer Science **150**, 186 (2019), proceedings of the 13th International Symposium "Intelligent Systems 2018" (INTELS'18), 22-24 October, 2018, St. Petersburg, Russia.
- [6] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Yeung, W. Wong, and W. Woo, *Deep learning for precipitation nowcasting: A benchmark and A new model*, CoRR **abs/1706.03458** (2017), arXiv:1706.03458 .
- [7] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, *Convolutional LSTM network: A machine learning approach for precipitation nowcasting*, CoRR **abs/1506.04214** (2015), arXiv:1506.04214 .
- [8] J. W. Wilson, N. A. Crook, C. K. Mueller, J. Sun, and M. Dixon, *Nowcasting thunderstorms: A status report*, Bulletin of the American Meteorological Society **79**, 2079 (1998), [https://doi.org/10.1175/1520-0477\(1998\)079<2079:NTASR>2.0.CO;2](https://doi.org/10.1175/1520-0477(1998)079<2079:NTASR>2.0.CO;2) .
- [9] L. Li, W. Schmid, and J. Joss, *Nowcasting of motion and growth of precipitation with radar over a complex orography*, Journal of applied meteorology **34**, 1286 (1995).
- [10] W.-c. Woo and W.-k. Wong, *Operational application of optical flow techniques to radar-based rainfall nowcasting*, Atmosphere **8**, 48 (2017).
- [11] B. W. Golding, *Nimrod: a system for generating automated very short range forecasts*, Meteorological Applications **5**, 1 (1998), <https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1017/S1350482798000577> .
- [12] C. Mueller, T. Saxen, R. Roberts, J. Wilson, T. Betancourt, S. Dettling, N. Oien, and J. Yee, *Ncar auto-nowcast system*, Weather and Forecasting **18**, 545 (2003), [https://doi.org/10.1175/1520-0434\(2003\)018<0545:NAS>2.0.CO;2](https://doi.org/10.1175/1520-0434(2003)018<0545:NAS>2.0.CO;2) .
- [13] S. Mecklenburg, J. Joss, and W. Schmid, *Improving the nowcasting of precipitation in an alpine region with an enhanced radar echo tracking algorithm*, Journal of Hydrology **239**, 46 (2000).
- [14] B. J. Turner, I. Zawadzki, and U. Germann, *Predictability of precipitation from continental radar images. part III: Operational nowcasting implementation (MAPLE)*, Journal of Applied Meteorology **43**, 231 (2004).

- [15] N. E. Bowler, C. E. Pierce, and A. W. Seed, *STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP*, Quarterly Journal of the Royal Meteorological Society **132**, 2127 (2006).
- [16] P. Li and E. S. Lai, *Short-range quantitative precipitation forecasting in hong kong*, Journal of Hydrology **288**, 189 (2004).
- [17] J. Johnson, P. L. MacKeen, A. Witt, E. D. W. Mitchell, G. J. Stumpf, M. D. Eilts, and K. W. Thomas, *The storm cell identification and tracking algorithm: An enhanced wsr-88d algorithm*, Weather and forecasting **13**, 263 (1998).
- [18] M. Dixon and G. Wiener, *Titan: Thunderstorm identification, tracking, analysis, and nowcasting—a radar-based methodology*, Journal of Atmospheric and Oceanic Technology **10**, 785 (1993), [https://doi.org/10.1175/1520-0426\(1993\)010<0785:TTITAA>2.0.CO;2](https://doi.org/10.1175/1520-0426(1993)010<0785:TTITAA>2.0.CO;2) .
- [19] G. Villarini and W. F. Krajewski, *Review of the different sources of uncertainty in single polarization radar-based estimates of rainfall*, Surveys in Geophysics **31**, 107 (2010).
- [20] U. Germann, I. Zawadzki, and B. Turner, *Predictability of precipitation from continental radar images. part iv: Limits to prediction*, Journal of the Atmospheric Sciences **63**, 2092 (2006), <https://doi.org/10.1175/JAS3735.1> .
- [21] J. W. Wilson, Y. Feng, M. Chen, and R. D. Roberts, *Nowcasting challenges during the beijing olympics: Successes, failures, and implications for future nowcasting systems*, Weather and Forecasting **25**, 1691 (2010), <https://doi.org/10.1175/2010WAF2222417.1> .
- [22] B. Radhakrishna, I. Zawadzki, and F. Fabry, *Predictability of precipitation from continental radar images. part v: Growth and decay*, Journal of the Atmospheric Sciences **69**, 3336 (2012), <https://doi.org/10.1175/JAS-D-12-029.1> .
- [23] A. Tsonis and G. Austin, *An evaluation of extrapolation techniques for the short-term prediction of rain amounts*, Atmosphere-Ocean **19**, 54 (1981).
- [24] N. Rehak, D. Megenhardt, and C. Mueller, *Large-scale trending of radar data*, - (2019).
- [25] J. Sun, M. Xue, J. Wilson, S. Ballard, J. Onvlee, P. Joe, M. Barker, P.-W. Li, B. Golding, M. Xu, J. Pinto, Sun, Wilson, and Xu, *Use of nwp for nowcasting convective precipitation: Recent progress and challenges*, Bulletin of the American Meteorological Society **95**, 409 (2013).
- [26] D. M. Barker, W. Huang, Y.-R. Guo, A. J. Bourgeois, and Q. N. Xiao, *A three-dimensional variational data assimilation system for mm5: Implementation and initial results*, Monthly Weather Review **132**, 897 (2004), [https://doi.org/10.1175/1520-0493\(2004\)132<0897:ATVDAS>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<0897:ATVDAS>2.0.CO;2) .
- [27] X. Tian, A. Dai, X. Feng, H. Zhang, R. Han, and L. Zhang, *A new approach for 4dvar data assimilation*, arXiv preprint arXiv:1805.09320 (2018).
- [28] M. Lindskog, K. Salonen, H. Järvinen, and D. B. Michelson, *Doppler radar wind data assimilation with hirlam 3dvar*, Monthly Weather Review **132**, 1081 (2004), [https://doi.org/10.1175/1520-0493\(2004\)132<1081:DRWDAW>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<1081:DRWDAW>2.0.CO;2) .
- [29] S. Sugimoto, N. A. Crook, J. Sun, Q. Xiao, and D. M. Barker, *An examination of wrf 3dvar radar data assimilation on its capability in retrieving unobserved variables and forecasting precipitation through observing system simulation experiments*, Monthly Weather Review **137**, 4011 (2009), <https://doi.org/10.1175/2009MWR2839.1> .
- [30] B. Macpherson, M. Lindskog, V. Ducrocq, M. Nuret, G. Gregorič, A. Rossa, G. Haase, I. Holleman, and P. P. Alberoni, *Assimilation of radar data in numerical weather prediction (nwp) models*, in *Weather Radar: Principles and Advanced Applications*, edited by P. Meischner (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004) pp. 255–279.
- [31] K.-S. Chung, I. Zawadzki, M. K. Yau, and L. Fillion, *Short-term forecasting of a midlatitude convective storm by the assimilation of single-doppler radar observations*, Monthly Weather Review **137**, 4115 (2009), <https://doi.org/10.1175/2009MWR2731.1> .

- [32] N. Gustafsson, T. Janjić, C. Schraff, D. Leuenberger, M. Weissmann, H. Reich, P. Brousseau, T. Montmerle, E. Wattrelot, A. Bučánek, M. Mile, R. Hamdi, M. Lindskog, J. Barkmeijer, M. Dahlbom, B. Macpherson, S. Ballard, G. Inverarity, J. Carley, C. Alexander, D. Dowell, S. Liu, Y. Ikuta, and T. Fujita, *Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres*, Quarterly Journal of the Royal Meteorological Society **144**, 1218 (2018).
- [33] X. Li, M. Zeng, Y. Wang, W. Wang, H. Wu, and H. Mei, *Evaluation of two momentum control variable schemes and their impact on the variational assimilation of radarwind data: Case study of a squall line*, Advances in Atmospheric Sciences **33**, 1143 (2016).
- [34] J. Sun, M. Chen, and Y. Wang, *A frequent-updating analysis system based on radar, surface, and mesoscale model data for the beijing 2008 forecast demonstration project*, Weather and Forecasting **25**, 1715 (2010), <https://doi.org/10.1175/2010WAF2222336.1> .
- [35] C. D. Jones and B. Macpherson, *A latent heat nudging scheme for the assimilation of precipitation data into an operational mesoscale model*, Meteorological Applications **4**, 269 (1997), <https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1017/S1350482797000522> .
- [36] M. L. Weisman, C. Davis, W. Wang, K. W. Manning, and J. B. Klemp, *Experiences with 0–36-h explicit convective forecasts with the wrf-arw model*, Weather and Forecasting **23**, 407 (2008), <https://doi.org/10.1175/2007WAF2007005.1> .
- [37] A. Overeem, I. Holleman, and A. Buishand, *Derivation of a 10-year radar-based climatology of rainfall*, Journal of Applied Meteorology and Climatology **48**, 1448 (2009), <https://doi.org/10.1175/2009JAMC1954.1> .
- [38] J. Marshall, W. Hirschfeld, and K. Gunn, *Advances in radar weather*, in -, Advances in Geophysics, Vol. 2, edited by H. Landsberg (Elsevier, 1955) pp. 1 – 56.
- [39] C. Tomasi and T. Kanade, *Detection and Tracking of Point Features*, Tech. Rep. (International Journal of Computer Vision, 1991).
- [40] J.-Y. Bouguet, *Pyramidal implementation of the lucas kanade feature tracker*, (1999).
- [41] C. Kuglin and D. Hines, *The phase correlation image alignment method*, IEEE International Conference on Cybernetics and Society (1975).
- [42] Y. Liang, *Phase-correlation motion estimation*, EE392J Project report (2000).
- [43] D. Svozil, V. Kvasnicka, and J. Pospichal, *Introduction to multi-layer feed-forward neural networks*, Chemometrics and Intelligent Laboratory Systems **39**, 43 (1997).
- [44] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, International Conference on Learning Representations (2014), <http://arxiv.org/abs/1412.6980v9> .
- [45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, (2015), software available from tensorflow.org.