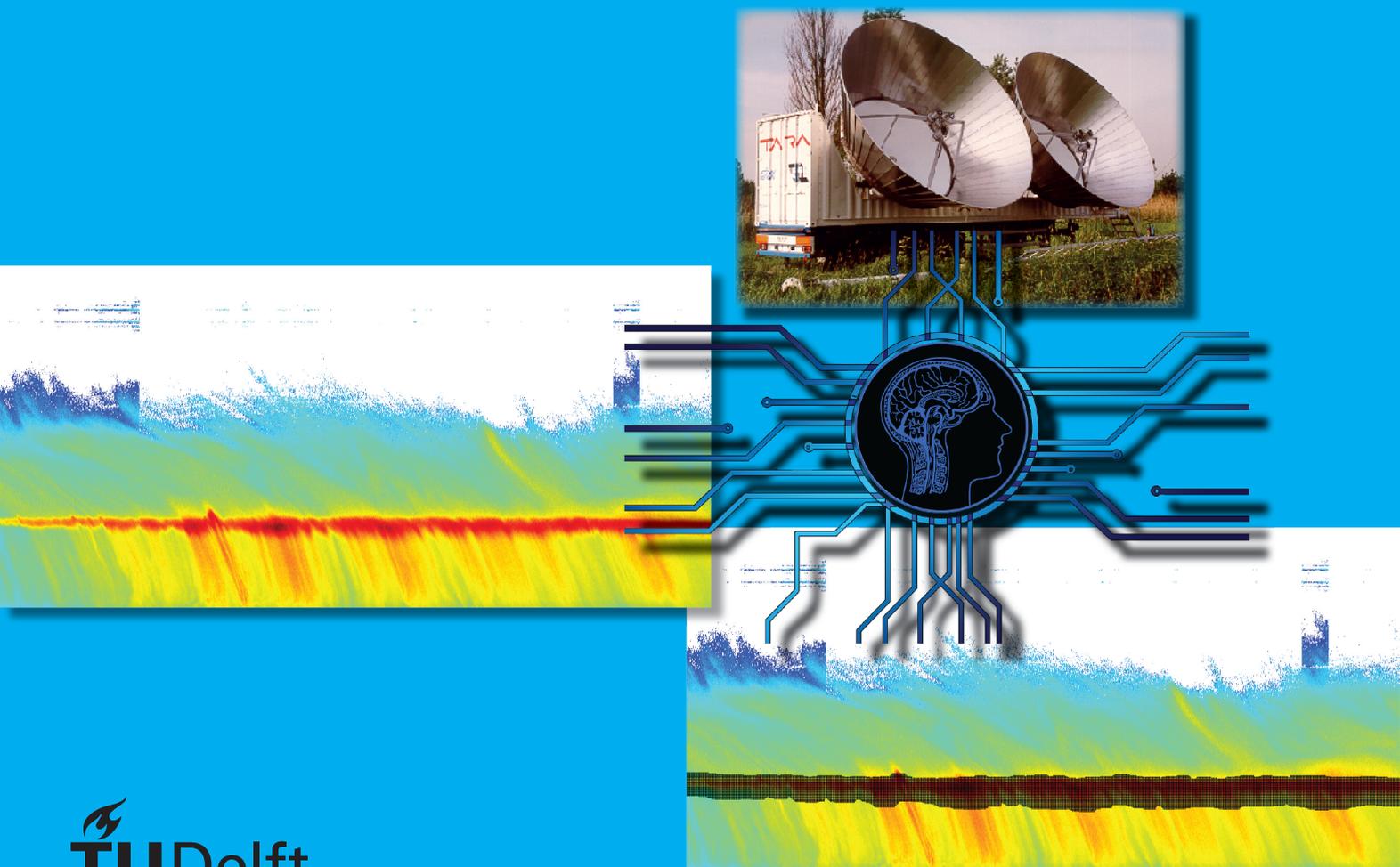


# A machine learning based approach to melting layer detection

M.T. Berndsen

Master Thesis





# A machine learning based approach to melting layer detection

by

M.T. Berndsen

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Tuesday June 18, 2019 at 14:30 PM.

Student number: 1386972  
Project duration: November 15, 2017 – June 18, 2019  
Thesis committee: Prof. dr. ir. H. W. J. Russchenberg, TU Delft, supervisor  
Ir. C. M. H. Unal, TU Delft, daily supervisor  
Dr. O.A. Krasnov, TU Delft  
Dr. M.A. Schleiss, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



# Abstract

Stratiform precipitation is one of the most important precipitation systems in the mid-latitudes. To gain understanding of the melting layer, which is part of a stratiform precipitating system, an algorithm which is able to select melting layer data from large datasets is required. The goal of this thesis work is to create and deliver a program which is able to perform this work.

For development the available data was separated into three groups: no melting layer, only melting layer and the rest. Based on the second group, typical melting layer signatures are analysed. In particular the different signatures (reflectivity, polarimetric and Doppler) occur at different heights. Based on the knowledge gained from the analysis and focussing on the reflectivity and polarimetric signatures, three different approaches were taken to detect and characterise the melting layer from the data. The first approach is based on an existing method in literature, this approach acts as the reference method. The second approach is based on image processing techniques, while the third approach is based on machine learning techniques. The second approach is later abandoned because of limitations of the techniques investigated. The third approach, machine learning, is the main contribution of this thesis work. For analysis of the performance of the different algorithms an annotated test dataset is created which represents the entire dataset. The performance is determined by the melting layer detection probability using a confusion matrix and by the determination of errors of the melting layer boundaries.

The reference method proved to show an extremely low false positive rate (0%) on the test dataset. This means that if the method detected a melting layer, it is almost certain that there is one. The overall detection probability was 80%. The method fails in detecting the melting layer when the peak reflectivity is below the threshold (30dBZ) used in the method. The detected upper melting layer boundary of the reference method is on average 365 m lower compared to the ground truth. The lower boundary is on average 149 m higher than the ground truth. This means that the reference method only selects a part of the entire thickness of the ML. The correlation between the lower boundary and the ground truth is higher than the upper boundary (0.987 vs. 0.965). The proposed machine learning method has a detection performance of almost 94%, which is higher than the reference method, but some false positives occur (3%). The upper boundary is on average 20 m above the ground truth, while the lower boundary is 66 m lower than the ground truth. The machine learned method thus overestimates the thickness of the ML. The correlation between the lower boundary and the true boundary is 0.976 and the correlation between the upper boundary and the true boundary is 0.966. The upper boundaries of both methods have a very similar correlation, while the lower boundary of the machine learning method has a lower correlation.

However, despite the lower correlation of the lower boundary and the introduction of false positives, the machine learning method is an improvement over the reference method since it has a significantly higher detection probability and it captures the entire thickness of the ML much better. It is therefore suited for ML analysis.



# Contents

<b>List of Figures</b>	<b>1</b>
<b>List of Tables</b>	<b>5</b>
<b>1 Introduction</b>	<b>7</b>
1.1 Melting layer importance . . . . .	8
1.2 Research questions . . . . .	9
1.3 ML detection and characterisation algorithm requirements . . . . .	9
1.4 Thesis outline . . . . .	10
<b>2 Literature review</b>	<b>11</b>
2.1 Radar introduction . . . . .	11
2.2 Single polarisation radar . . . . .	12
2.2.1 Reflectivity profile ( $Z_{e,hh}$ ) . . . . .	12
2.2.2 Doppler profile. . . . .	18
2.2.3 Doppler spectral width and Doppler spectral skewness . . . . .	19
2.3 Polarimetric radar. . . . .	20
2.3.1 Differential reflectivity ( $Z_{DR}$ ). . . . .	21
2.3.2 Linear Depolarisation Ratio ( $L_{DR}$ ) . . . . .	22
2.3.3 Copolar correlation coefficient ( $\rho_{hv}$ ). . . . .	24
2.3.4 Differential propagation phase ( $\Phi_{DP}$ ) and specific differential phase ( $K_{DP}$ ) . . . . .	25
2.4 Conclusion on Melting layer detection methods . . . . .	27
<b>3 Data</b>	<b>29</b>
3.1 Tara and ACCEPT measurement campaign . . . . .	29
3.2 Acquiring mean and standard deviation of ML . . . . .	31
3.2.1 Mean and standard deviation of ML signature at 45° elevation. . . . .	32
3.2.2 Mean and standard deviation of ML signature at 90° elevation. . . . .	38
3.3 Relative height of ML signatures . . . . .	41
3.4 Removal of upper atmosphere unwanted echo . . . . .	43
3.5 Comparison with radiosonde measurement . . . . .	45
3.6 Conclusion . . . . .	48
<b>4 Methodology</b>	<b>49</b>
4.1 Melting layer boundary definition . . . . .	49
4.2 Reference ML characterisation method. . . . .	50
4.3 Image processing ML characterisation methods . . . . .	52
4.3.1 Hough transform. . . . .	52
4.3.2 Radon transform. . . . .	54
4.3.3 Connected pixels (area size) . . . . .	54
4.4 Machine learning ML characterisation method . . . . .	54
4.4.1 ML detector . . . . .	55
4.4.2 ML attributer . . . . .	57
4.5 Evaluation of identification methods . . . . .	62
<b>5 Results</b>	<b>67</b>
5.1 Reference ML characterisation method . . . . .	67
5.2 Image processing ML characterisation methods . . . . .	70
5.2.1 Hough transform. . . . .	70
5.2.2 Radon transform. . . . .	71
5.2.3 Connected pixels. . . . .	71

5.3	Machine Learning ML characterisation method . . . . .	73
5.3.1	Machine learning ML detector . . . . .	73
5.3.2	Machine learning ML attributer . . . . .	79
5.3.3	Combined evaluation of machine learning ML detector and attributer . . . . .	83
<b>6</b>	<b>Conclusion and discussion</b>	<b>89</b>
6.1	Conclusions. . . . .	89
6.2	ML detection and characterisation program . . . . .	91
6.3	Recommendations for future work . . . . .	93
<b>A</b>	<b>Cells with similar features</b>	<b>95</b>
<b>B</b>	<b>Histograms SVM detectors</b>	<b>97</b>
<b>C</b>	<b>Histograms ensemble detector</b>	<b>99</b>
<b>D</b>	<b>Histograms special cases ensemble of bagged trees</b>	<b>101</b>
<b>E</b>	<b>List of files required for the GUI</b>	<b>103</b>
<b>F</b>	<b>List of files for generating thesis results</b>	<b>105</b>
<b>G</b>	<b>User manual of GUI</b>	<b>109</b>
G.1	Load and pre-process data . . . . .	109
G.2	Use machine learning part of GUI. . . . .	112
G.2.1	Modify machine learners and training data . . . . .	113
G.3	Apply default algorithms to new dataset . . . . .	116
	<b>Bibliography</b>	<b>119</b>

# List of Figures

1.1	This figure shows two different cloud types: stratiform clouds on the left and cumuliform clouds on the right. Image downloaded from <a href="https://www.britannica.com/science/atmosphere#ref952875">https://www.britannica.com/science/atmosphere#ref952875</a> . . . . .	7
1.2	Stratiform cloud structure. (From Houze (1981)). . . . .	8
2.1	Most basic radar operating principle. (Downloaded From <a href="http://radarportal.skyradar.com/radar-basics/radar-principle/">http://radarportal.skyradar.com/radar-basics/radar-principle/</a> ). . . . .	11
2.2	Schematic overview of the reflectivity of the ML. . . . .	12
2.3	Some examples of stratiform HTI plots. (images from Rico-Ramirez et al. (2007)). . . . .	13
2.4	Example of stratiform HTI with dual ML. (Image from Rico-Ramirez et al. (2007)). . . . .	13
2.5	Two examples of convective HTI plots. (images from Rico-Ramirez et al. (2007)). . . . .	14
2.6	Both vertical reflectivity and temperature profile in one plot. (Image from Stewart et al. (1984)). . . . .	14
2.7	Vertical reflectivity profile (left) and gradient of the vertical reflectivity profile (right). (image from Tilford et al. (2001) . . . . .	15
2.8	Detection of the bright band boundaries using a coordinate transformation. (Image from Rico-Ramirez and Cluckie (2007)) . . . . .	16
2.9	Analytical representation of the vertical profile of reflectivity. (Image from Pfaff et al. (2014)) . . . . .	17
2.10	Schematic representation of vertical velocity profile. (Image from Baldini and Gorgucci (2006)) . . . . .	18
2.11	Example of vertical Doppler profile (left) and vertical reflectivity profile (right). (Image from White et al. (2002)) . . . . .	19
2.12	Example of Doppler spectral width. (Image from <a href="http://apollo.lsc.vsc.edu/classes/remote/lecture_notes/radar/doppler/spectrum_width.html">http://apollo.lsc.vsc.edu/classes/remote/lecture_notes/radar/doppler/spectrum_width.html</a> ) . . . . .	19
2.13	Example of Doppler spectral skewness. (Image from <a href="https://codeburst.io/2-important-statistics-terms-you-must-know/">https://codeburst.io/2-important-statistics-terms-you-must-know/</a> ) . . . . .	19
2.14	Dual polarisation radar used on a non-spherical droplet. (Downloaded From <a href="https://commons.wikimedia.org/wiki/File:Radar-polarisation_en.svg">https://commons.wikimedia.org/wiki/File:Radar-polarisation_en.svg</a> ). . . . .	21
2.15	Principle of depolarisation of electric field due to orientation. (Image from Meischner (2013) page 138) . . . . .	23
2.16	Example of a vertical profile of the $L_{DR}$ in a stratiform event. (Image from Brandes and Ikeda (2004)) . . . . .	23
2.17	Example of a vertical profile of the correlation coefficient in a stratiform event. (Image from Brandes and Ikeda, 2004) . . . . .	24
2.18	Example of a vertical profile of the differential phase standard deviation in a stratiform event. (Image from Baldini and Gorgucci (2006)) . . . . .	26
2.19	Two images of the same stratiform rain situation. The left image shows $K_{DP}$ and the right image shows $Z_{DR}$ . In this situation a ML was present as can be seen in the right image, but in the left image $K_{DP}$ is not showing a ML characteristic. (images from Thompson et al. (2014)) . . . . .	27
3.1	TARA radar. (Image from <a href="http://www.cloud-net.org/instruments/documents/Tara.pdf">www.cloud-net.org/instruments/documents/Tara.pdf</a> ) . . . . .	29
3.2	ML profile alignment procedure on data from 24-10-2014. The maximum $Z_{e,hh}$ provides the reference height, every profile is shifted in height to match that reference. . . . .	33
3.3	Reflectivity in a clear ML signature scenario . . . . .	34
3.4	Cross polar equivalent reflectivity factor in a clear ML signature scenario . . . . .	34
3.5	$Z_{DR}$ in a clear ML signature scenario . . . . .	34
3.6	$L_{DR}$ in a clear ML signature scenario . . . . .	35
3.7	$\rho_{hv}$ in a clear ML signature scenario . . . . .	35
3.8	$\Psi_{dp}$ in a clear ML signature scenario . . . . .	35
3.9	Example of a first derivative profile of the vertical velocity (profile number 352 of the study case on the 24th of October). . . . .	36
3.10	Vertical velocity in a clear ML signature scenario . . . . .	36

3.11	Example of a vertical profile of the first derivative of the Doppler spectral width (profile number 352 of the study case on the 24th of October). . . . .	37
3.12	Doppler spectral width in a clear ML signature scenario . . . . .	37
3.13	Example of a first derivative profile of the Doppler spectral skewness (profile number 352 of the study case on the 24th of October). . . . .	38
3.14	Doppler spectral skewness in a clear ML signature scenario . . . . .	38
3.15	All available radar observables when TARA was pointing at 90°. Most observables show a significant ML characteristic. Only $Z_{DR}$ and $\Psi_{DP}$ are showing almost no ML signature. . . . .	40
3.16	A comparison of the relative altitudes with respect to their peak values. . . . .	42
3.17	The result of imposing a SNR threshold of 10 dB on the data. . . . .	43
3.18	Example of removing single data cells using image dilation of the NaN map . . . . .	44
3.19	Process of image erosion of a black and white image using a structuring element (se). 1 indicates white (data), 0 indicates black (no data). . . . .	45
3.20	Horizontal equivalent reflectivity factor of the used window for calculating the average profile. The black square represents the radiosonde measurement of the 0°C isotherm. . . . .	46
3.21	A comparison of the radiosonde 0 °C isotherm measurement and the radar observables. . . . .	47
4.1	The definition of the ML. This graph shows the average ML signatures of five different radar observables. The ML top is defined to be the transition given by the coordinate transformation method of Rico-Ramirez and Cluckie on the $Z_{e,hh}$ . The bottom of the ML is defined by the same method, but now applied to the $Z_{DR}$ . . . . .	50
4.2	The result of the Giangrande et al. method shown as overlay to the copolar correlation coefficient. . . . .	51
4.3	The result of the adjusted Giangrande et al. method shown as overlay to the copolar correlation coefficient. . . . .	51
4.4	The result of the adjusted Giangrande et al. method shown as overlay to the horizontal equivalent reflectivity factor. . . . .	52
4.5	Principle of the Hough transform. Every point in the image can be represented by a combination of angle $\theta$ and distance $r$ . . . . .	53
4.6	Principle of the Hough transform. The three lines are representing the three point of Figure 4.5. Where the lines intersect (at magenta cross) all the three points have one angle distance pair and as a result a line with that angle and distance pair can be drawn through the three points (image from <a href="https://en.wikipedia.org/wiki/Hough_transform">https://en.wikipedia.org/wiki/Hough_transform</a> ). . . . .	53
4.7	Principle of the Radon transform. The image is integrated over parallel beams. The beams are separated by 1 pixel in y direction. This process is repeated over a multitude of angles $\theta$ (image from MATLAB 2018a documentation). . . . .	54
4.8	A ML signature in two different datasets. . . . .	59
4.9	Example and solution to the problem of non-overlapping features. . . . .	60
4.10	The composition of training data for the attribution. . . . .	61
4.11	Example of a confusion matrix. The outcome of the classifier is indicated as "Output class", while the ground truth is indicated by "True class". . . . .	63
5.1	Confusion matrix of the reference method. Note that no false positives were initiated by the method. . . . .	67
5.2	Two examples of instants where the reference method did not detect the ML. The blue lines indicate the true ML boundaries according to the definition and the black dots indicate the identified ML cells. . . . .	68
5.3	Example of detection result of the reference method with the $Z_{DR}$ as the background image of the test dataset at an instance where a clear ML signature was present. The method resulted in a false negative because the peak $Z_{DR}$ is much lower than the threshold of 500 m below the cell under investigation. . . . .	69
5.4	Example of detection result of the reference method on the test dataset at an instance where a clear ML signature was present. The blue lines indicate the true ML boundaries according to the definition and the black dots indicate the identified ML cells. The reference method identifies ML cells in the lower part of the total ML, but misses the top part. . . . .	69
5.5	Upper picture is the input image, lower picture is the Hough parameter space as explained in subsection 4.3.1. . . . .	70

5.6	Result of horizontal line detection using the Hough transform. . . . .	71
5.7	Radon transform of the clear ML scenario with three detections indicated by the white squares using peak detection method. . . . .	72
5.8	Result of horizontal line detection using the Radon transform. . . . .	72
5.9	Result of connected pixel method. . . . .	73
5.10	a-d: Examples of feature sets with good class separating capabilities. e-f: Examples of feature sets without class separating capabilities. . . . .	75
5.11	Confusion matrices of the detection performance of the linear SVM detectors using the full set of features (a) and the specified subset of features (b). . . . .	76
5.12	Confusion matrices of the detection performance of the Quadratic SVM detectors using the full set of features (a) and the specified subset of features (b). . . . .	76
5.13	Confusion matrices of the detection performance of the coarse Gaussian SVM detectors using the full set of features (a) and the specified subset of features (b). . . . .	77
5.14	Confusion matrices of the detection performance of the ensemble of bagged trees detector using the full set of features (a) and the specified subset of features of the same detector (b . . . . .	77
5.15	Estimate of feature importance. . . . .	79
5.16	Examples of attributer feature sets. . . . .	80
5.17	The attribution accuracy of the best three performing KNN algorithms and the extended coarse KNN algorithm with K=150. . . . .	82
5.18	Estimate of feature importance based on the ensemble of bagged trees. . . . .	83
5.19	Raw output of the attributer of the test dataset. Figure 5.19a shows the entire dataset, while Figure 5.19b shows a specific part in more detail to show where false positives occur. . . . .	84
5.20	Detailed images of $Z_{e,hh}$ , $Z_{DR}$ and $Z_{e,hv}$ within the rectangle of Figure 5.19b. False positives shown in Figure 5.19b are the result of high values of these three radar observables. . . . .	85
5.21	Example of the raw output of the machine learning ML characterisation method. Within the rectangle a false detection is made and the attributer also selected ML cells. . . . .	85
5.22	Detailed part of the test dataset with a clear ML. The black dots indicate the raw output of the machine learning method and the blue lines indicate the ML boundaries according to the ML definition. . . . .	86
5.23	Result of post processed machine learned ML characterisation method. . . . .	86
5.24	Detailed view of the post processed output of the attributer of the same data as shown in Figure 5.22. The detected boundaries follow the true boundaries very well, but some detail is lost due to the image erosion and dilation procedures in post processing. . . . .	87
5.25	Detailed images of parts of post processed output of the machine learned ML characterisation method. . . . .	87
6.1	Example of the GUI. . . . .	92
A.1	Images show the data cells from Table A.1 and their surrounding data. Both have very similar features, but the left image is clearly not part of a ML while the right image is. . . . .	95
B.1	Histograms of total detection performance of linear SVM detector with the full feature set (a) and subset of features (b). . . . .	97
B.2	Histograms of total detection performance of Quadratic SVM detector with the full feature set (a) and subset of features (b). . . . .	98
B.3	Histograms of total detection performance of coarse Gaussian SVM detector with the full feature set (a) and subset of features (b). . . . .	98
C.1	Histograms of total detection performance of ensemble of bagged trees detector with the full feature set (a) and subset of features (b). . . . .	99
D.1	Histograms of total detection performance of ensemble of bagged trees detector when on the left all features which use $Z_{DR}$ are removed and on the right all features which use $Z_{e,hv}$ are removed. . . . .	101
G.1	Example of the GUI, within the black rectangle is the part of the tool situated which focusses on loading, processing and displaying data. . . . .	110

---

G.2	Load data window. . . . .	110
G.3	Example of the GUI, within the black rectangle is the part of the tool situated which focusses on machine learning. . . . .	112
G.4	Location of the classification learner APP. . . . .	113
G.5	Opening screen of classification learner. . . . .	114
G.6	Main screen of classification learner app. . . . .	114
G.7	Portion of code of main.m where the training of an algorithm is performed. . . . .	115
G.8	Portion of code of main.m where the training of an algorithm is performed. . . . .	116
G.9	Example of the GUI, within the black rectangle is the part of the program indicated which focusses on applying a default machine learned method to new data. . . . .	117

# List of Tables

3.1	TARA specifications during the ACCEPT campaign . . . . .	30
3.2	Reference height for alignment of vertical profiles . . . . .	32
3.3	Absolute mean peak altitude and mean relative altitude . . . . .	41
4.1	Training dataset composition for detector . . . . .	56
4.2	Training array format for detector . . . . .	56
4.3	Non-ML training dataset composition for attributer . . . . .	60
4.4	Training array format for attributer . . . . .	61
4.5	Test dataset composition . . . . .	62
4.6	Example of part of ground truth array (VP: Vertical profile) . . . . .	63
5.1	Errors w.r.t. the true ML boundaries. . . . .	70
5.2	Mean and standard deviation of total detection performance. . . . .	76
5.3	Mean and standard deviation of total detection performance. . . . .	77
5.4	Mean and standard deviation of total detection performance. . . . .	78
5.5	Sensitivity to removal of single features. . . . .	78
5.6	Sensitivity to removal of $Z_{DR}$ or $Z_{e,hv}$ . . . . .	79
5.7	Mean and standard deviation of attribution performance. . . . .	81
5.8	Sensitivity to removal of single features. . . . .	81
5.9	Errors w.r.t. the true ML boundaries. . . . .	88
A.1	Example of training data with similar features . . . . .	95
E.1	List of files required for the GUI. . . . .	103
F.1	List of M-files to generate the results in this thesis . . . . .	105
G.1	List of available data after pre-processing . . . . .	111



## Introduction

The melting layer, what makes this phenomena inside clouds so important? The melting layer and its associated stratiform clouds are responsible for most of the precipitation in the areas around the equator and about 50% in the mid latitudes. This makes the melting layer an important phenomena to be studied.

Basically, there are two types of clouds: stratiform clouds and cumuliform clouds. Both cloud types can be seen in Figure 1.1. On the left-hand side, the stratiform cloud types are displayed. Long it is understood that these two types of clouds were the only ones. Later, Houze (1981) concluded that both types indeed describe the two main cloud types, but complex combinations of both types are very common. Cumuliform clouds also have a melting process inside them, but due to the more violent nature of cumuliform clouds (turbulent mixture of all types of hydrometeors over a large vertical distance), this is not a clear layer as it is in stratiform clouds. It is outside the scope of this thesis to discuss this type of weather system.

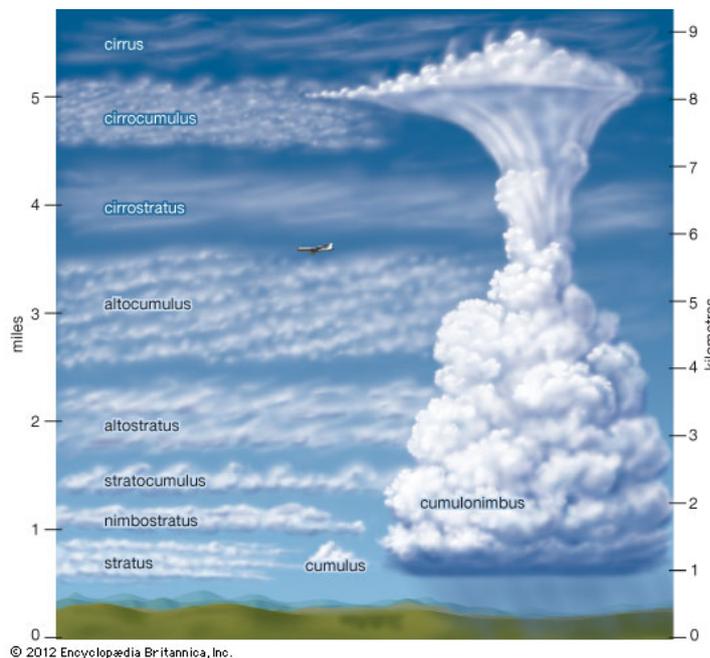


Figure 1.1: This figure shows two different cloud types: stratiform clouds on the left and cumuliform clouds on the right. Image downloaded from <https://www.britannica.com/science/atmosphere#ref952875>

Existence of the Melting Layer (from now on ML) is mostly associated with the presence of stratiform clouds according to Wolfensberger et al. (2016). Stratiform clouds occur as saturated air is (slowly) (Houze (1993)) forced upwards and remains colder than the surrounding air. The clouds are formed because the saturated air starts to condensate and forms clouds. This typically happens when large layers of air are pushed towards higher altitudes. Therefore, stratiform clouds typically cover large areas, whereas cumuliform clouds

cover smaller areas. The ML is mostly associated with stratiform clouds, because stratiform clouds are more stable than cumuliform clouds. Some characteristics of stratiform clouds are:

- *Uniform precipitation type;*
- *Little horizontal variability in rainfall;*
- *Not vertically developed.*

The general physical process of stratiform clouds is shown in Figure 1.2. It shows how precipitation is formed in a stratiform event. Ice particles travel downwards, where the temperature increases due to the increasing pressure closer to the surface of the earth (adiabatic displacement of air). When the temperature reaches  $-16^{\circ}\text{C}$ , ice aggregation starts and is enhanced when the temperature is above  $-6^{\circ}\text{C}$ . The aggregation process forms large snowflakes which begin to melt when the isothermal layer ( $0^{\circ}\text{C}$ ) is reached. Melting of the snowflakes causes liquid water to be distributed across the outer layers of the snowflake and the snowflakes start to slowly decrease in size. In the first part (from above) of the ML, snowflakes are still relatively large with a thin layer of liquid water distributed across the outer layers. Slowly, as the snowflakes descent, more water will distribute over the outer layers and snowflakes will decrease in size more rapidly. Finally, the descending snowflakes are becoming increasingly spherical, start to break up and pick up speed. When all snow and ice

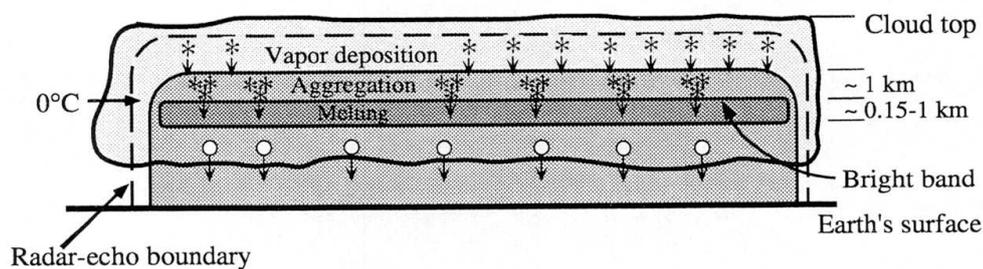


Figure 1.2: Stratiform cloud structure. (From Houze (1981).

particles have melted, the precipitation continues towards Earth in liquid form (rain). In the next section more information is given on the importance of ML characterisation.

## 1.1. Melting layer importance

Knowledge about the ML is important for the following reasons:

- Rainfall estimation;
- ML characteristics;
- Prediction of precipitation types;
- Automated removal of ML from large datasets;
- Extraction of  $0^{\circ}$  isotherm from ML radar measurements.

Knowing where a ML occurs helps in rainfall estimation. Rainfall estimation for a horizontally scanning radar at larger distances is difficult. The radar beam will penetrate clouds instead of rain as a result of the curvature of the earth and the beam elevation of the radar. Increased reflection is occurring in the melting layer, causing the rainfall estimate to be incorrect. Rainfall estimates at these locations are overestimated as reported by Fabry et al. (1992) and Smith (1986). Or rainfall estimates are underestimated when a melting layer compensation algorithm is applied, but no melting layer is present. This example has been reported by Smyth and Illingworth (1998), page 2433 and Kitchen et al. (1994), page 1245.

By detecting the melting layer, one could extract information from those detections in terms of dimensions. Information on the thickness of the ML, the height of the ML and the variability of height of the ML can be extracted. Characterisation of the ML can be carried out when the ML has been identified.

Melting layer detection is important for precipitation type prediction as stated by Kain et al. (2000). They state that the melting layer can cause a significant change in precipitation type. The melting of frozen hydrometeors costs energy, which it takes from the surrounding air, causing the air temperature to decrease. At

some point the air could become below freezing level, which in turn causes a sudden change in precipitation type because frozen hydrometeors no longer melt. When this happens, severe snowfall occurs where rain was predicted.

For researching only rain it is necessary to remove the non-linear response of the ML from a large dataset. It contains both ice and liquid particles and may influence the outcome of the research. Therefore, an automated system is desired which is able to process large datasets and is able to select the portion of the data which is "contaminated" with the ML. If such a system is able to extract the ML data from a dataset, the result can also be used to study the ML itself, because it only contains ML data. ML parameters like the thickness and time variability can be extracted from the resulting dataset. This has not been done extensively at the TU-Delft due to the fact that no automatic ML detection algorithm is in use at the moment. Also, the locally determined ML statistics could be compared to other studies around the world.

Another reason why the ML could be of importance is to provide information on the 0° isotherm. For weather forecasting, agencies would like to know as precise as possible where the 0°C isotherm layer is. The height is input for their weather prediction models to enable accurate weather predictions. Most of these models assume a linear behaviour of the ML in time, but this might not be entirely true. The time resolution of 0°C isotherm measurement is depending on radiosonde launches, where radar measurements can provide a much higher time resolution if stratiform precipitation is present.

The focus of this thesis work will be on automated selection of the ML for scientific purposes. Based on this focus, the next section will pay attention to the research questions involved.

## 1.2. Research questions

To structure the research carried out in this thesis, the following research questions are to be answered:

- What requirements need to be met to produce a ML detection and characterisation tool for scientific use?
- What is the (polarimetric) signature of the ML?
  - Is there a typical ML characteristic?
  - What are the differences between the available radar observables of a typical ML?
  - What is the influence of radar elevation on the observables in the ML?
- What steps need to be taken to prepare the available data for automatic ML detection and characterisation?
- Which methods are suitable for ML detection and characterisation?
  - What methods already exist in literature?
  - Can some basic image processing techniques work to detect and characterise the ML?
  - Is machine learning valuable in ML detection and characterisation?
- How can the developed and implemented ML detection and characterisation methods be assessed?
  - Is a suitable ML definition available for use?
  - Which performance indicators can be used to evaluate performance of the detection and characterisation methods?

The results of this thesis are twofold: answering the research questions but also deliver a "program" which automatically detects and characterises the ML with its associated boundaries. In the next section more details will be given on the requirements of the program.

## 1.3. ML detection and characterisation algorithm requirements

While ML detection and characterisation is important for a multitude of reasons, specifically the scientific use of an automated ML detection algorithm is chosen for this thesis because of its unique requirements. Requirements for the automatic ML data selection tool are:

- Ease of operation;
- Reliability of operation;
- Robust detection performance;
- Well documented;

- Calculation time within reason (no real time requirement);
- Accurately indicate the top of the melting layer (no liquid water present);
- Accurately indicate the bottom of the melting layer (no solid water present);
- Preferred option: algorithm independent of radar type;
- Ability to verify detection and characterisation results.

#### **1.4. Thesis outline**

To accomplish the development of such an algorithm, first a knowledge base about (polarimetric) radar and its use for ML detection is established in Chapter 2. Then Chapter 3 is spent on the data which is going to be used for this thesis. In this Chapter steps are taken to select and prepare the data, to gain knowledge about the signature of the ML and to compare the ML signature with a true 0°C isotherm. Chapter 4 continues by first setting up a definition of the ML which is appropriate for the goal of this thesis. Then the proposed ML detection and characterisation methods are explained. The methods consist of a reference method from the literature study, basic image processing methods and a machine learning method is proposed and developed. After the explanation of the ML characterisation methods, the performance metrics are explained. The results of the ML detection and characterisation methods will then be shown in Chapter 5. In this Chapter the methods will be tested against a large fixed dataset, so the methods can be compared in terms of ML detection and characterisation performance. Focus is given to the reference and the new proposed machine learning method. Finally, in Chapter 6, conclusions are drawn and recommendations for future work are suggested.

# 2

## Literature review

**This chapter will describe the literature study on using radar measurements for ML detection. For each radar observable the principle is explained and the different methods found in literature which use the observable are discussed. At the end of the chapter a conclusion on the different ML detection and characterisation methods will be drawn and a decision will be made on which methods will be used in the thesis.**

### 2.1. Radar introduction

Radar measurements are generally used for ML detection and analysis. Therefore, in this section a short introduction into radar basics is given. Only the necessary parts will be explained to be able to understand the thesis. Radar is an acronym of Radio Detection And Ranging. It means that objects can be detected and the range can be determined by means of using radio (electromagnetic) waves. The radar is transmitting an electromagnetic wave, the wave is partially reflected (backscattered) by the object and is received by the radar. The receiver part of the radar analyses the reflected signal. The time the electromagnetic wave took to travel forth and back is used to determine the range of the object. A simple overview can be seen in Figure 2.1. Radar systems can be divided into different types of radar systems. It is not the goal of this thesis to extensively

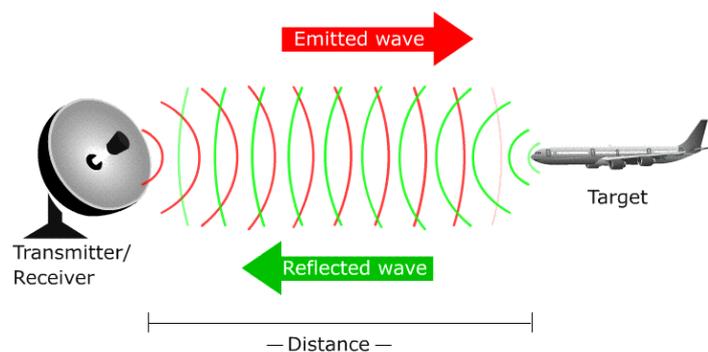


Figure 2.1: Most basic radar operating principle. (Downloaded From <http://radarportal.skyradar.com/radar-basics/radar-principle/>).

describe all types of radars, but only the necessary basis is laid down for understanding the thesis. Basically, radar systems can be divided into two types: non-polarimetric (or single polarisation) and polarimetric radar. The first transmits and receives electromagnetic waves in one polarisation direction and the latter, depending on the radar, can transmit and/or receive electromagnetic waves in different polarisation directions.

There are numerous ways of detecting the melting layer. Some methods are based on reflectivity only, but more modern methods are based on the products of polarimetric radars. Most of the early "detection" methods were only build for correcting the rainfall estimates (e.g. Klaassen (1988), Hardaker et al. (1995)). There is a distinct difference between the "detection" methods described in those papers and the required detection method in this thesis; their purpose is different. Most methods described in the earlier papers only work

when a stratiform rain event has already been registered. Then they are able to correct the surface rainfall rate using the developed methods, so if these methods would be applied to radar data without a stratiform rain event, the rainfall estimates will be severely underestimated. Later, better detection methods were developed, which recognized if a stratiform event was present or not. In the following section, the methods which only use single polarisation radar are being discussed. In the subsequent section also the polarimetric based detection methods are being discussed. The literature research dates back to some old methods since the author of this thesis had little a priori knowledge about ML detection methods.

## 2.2. Single polarisation radar

Single polarisation radars can only transmit and receive the electromagnetic wave in one polarisation direction. A horizontally polarised radar transmits and receives the electric field in the horizontal polarised direction. The receiver measures the amount of reflected power as a function of time. When the received electromagnetic wave is plotted in a figure, the result is a reflectivity profile in the direction of line of sight of the radar.

### 2.2.1. Reflectivity profile ( $Z_{e,hh}$ )

As already mentioned, the ML appears as a bright band in the reflectivity profile. Stewart et al. (1984) already published a paper about the characteristics of the ML in stratiform clouds. The conclusion is that the bright band is there because "aggregation, ice crystal multiplication, and raindrop breakup are operating within or near the ML, and that the characteristics of the radar bright band are critically dependent upon such processes". Essentially, ice starts to melt and a layer of water is formed around the ice particles and thus changing the dielectric constant of the particle, increasing its backscattering component Battan (1973). The dielectric constant of water is a factor 5 larger than the dielectric constant of ice according to Yau and Rogers (1996). At some point the backscattering component reaches a maximum (still a large ice particle and a reasonable amount of liquid water distributed over the outer layers of the snowflake). Below that level, the snowflakes start to break up, increase in velocity and melts further. A steep decrease in backscattering can be seen in a reflectivity profile. A schematic of the vertical reflectivity profile of the ML in stratiform clouds can be seen in Figure 2.2. In this figure, the characteristic parameters of the ML can be read: the bright band thickness ( $B$ ), the relative position of the peak ( $r_{peak}$ ), the height of the top ( $h_{top}$ ) and bottom ( $h_{bot}$ ) and the peak equivalent reflectivity ( $Z_{e(peak)}$ ).

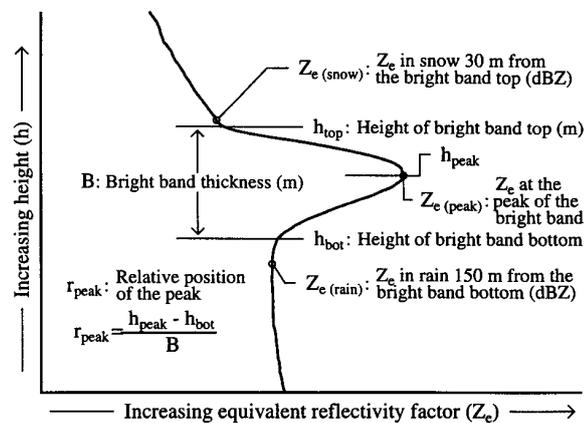


Figure 2.2: Schematic overview of the reflectivity of the ML.

An automatic detection algorithm should be able to handle the most common situations. Therefore, a deeper understanding of different vertical reflectivity profiles is in place. Some vertical profiles are shown in Figure 2.3. The figures are showing the height on the y-axis and time on the x-axis, which are called HTI (Height Time Indicator) plots. In these figures most of the time the ML is visible as the bright horizontal line. In the top-left figure, a horizontal ML is visible at approximately 1.7 km. In the top-right figure, the ML is not constant in time, the height of the ML changes due to a passing warm front. The lower left figure shows evaporation occurring in lower heights and the lower right figure shows a stratiform event which is disturbed by strong convection around the ML. A rarer event with two MLs can be seen in Figure 2.4 and some examples

of convective rain are shown in Figure 2.5. There is no clear visible ML to be detected when there is convective rain.

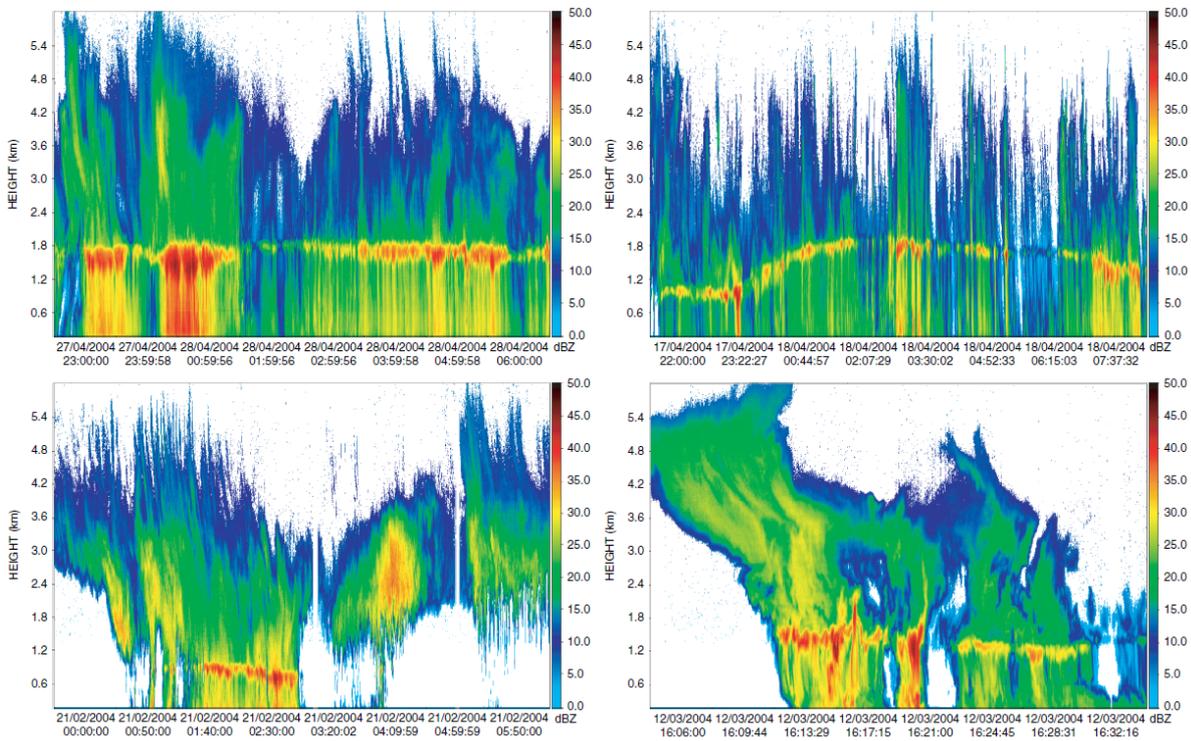


Figure 2.3: Some examples of stratiform HTI plots. (images from Rico-Ramirez et al. (2007)).

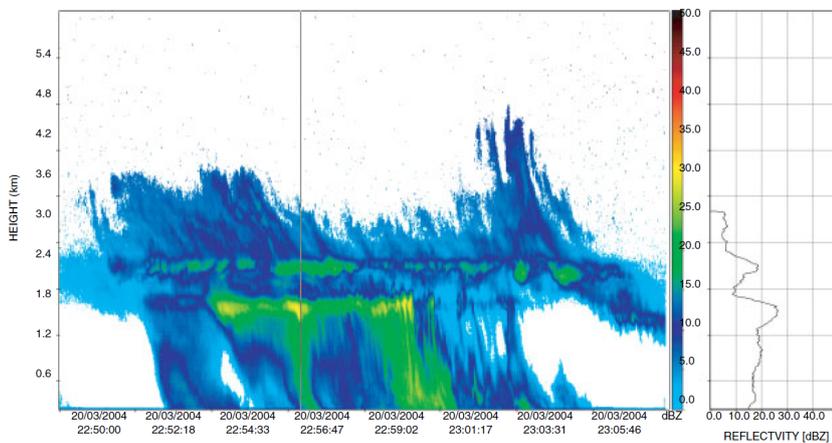


Figure 2.4: Example of stratiform HTI with dual ML. (Image from Rico-Ramirez et al. (2007)).

Now some examples from practice have been shown, how does the stratiform system relate to a temperature profile? Stewart et al. (1984) has measured the reflectivity and the temperature profile. The graph from that paper is shown in Figure 2.6. From the graph it is noticed that the peak of the bright band does not correspond to the actual layer where the temperature is 0°C, but it does correspond to a temperature between 0° C and 5° C. This corresponds to the figure from Houze (1981), shown in Figure 1.2. Now that a better understanding is created about the different situations in stratiform precipitation, let's start off with the ML detection methods found in literature which use the reflectivity as their main source.

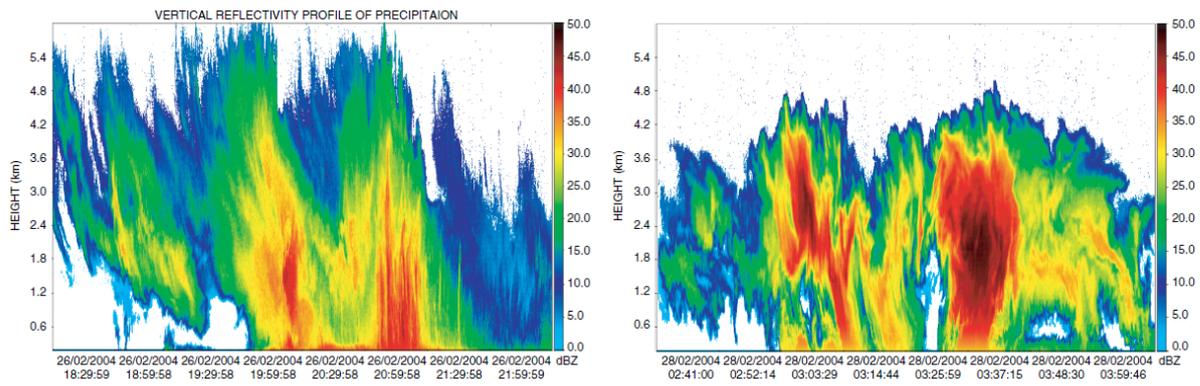


Figure 2.5: Two examples of convective HTI plots. (images from Rico-Ramirez et al. 2007).

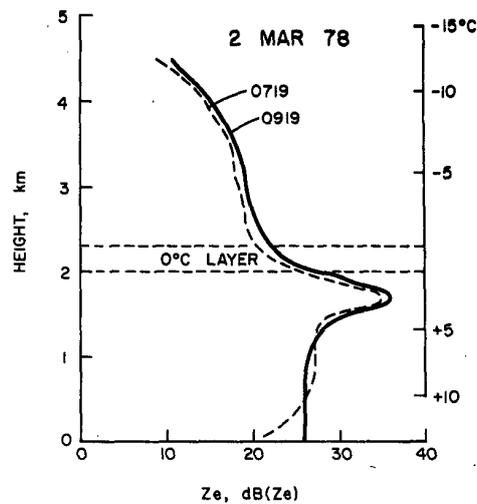


Figure 2.6: Both vertical reflectivity and temperature profile in one plot. (Image from Stewart et al. (1984)).

### 2.2.1.1 ML detection methods based on reflectivity profiles

As shown in the previous section, the reflectivity in the melting layer has been reported as significantly higher than above or below the melting layer. This could be used for detection. All ML detection methods in this section are based on retrieval of the vertical profile of reflectivity (from hereafter VPR). A downside of using the VPR is that one has to assume that the acquired VPR is representative for the entire radar coverage. This has been reported by Gourley and Calvert (2003) and others.

The first step is to acquire the VPR. Generally, this can be done in three ways: use a dedicated vertically pointing radar to actually measure the VPR, use a radar which scans in elevation (create RHI plots) or use a horizontally scanning radar (PPI) and derive the apparent VPR by making use of the elevation of the radar beam or use multiple beams in elevation. The differences between the methods is mostly on the spatial validity of the (derived) VPR. A dedicated vertically oriented radar produces a VPR which is measured directly above the radar and thus there is no certainty about the (horizontal) spatial validity of the acquired VPR. A VPR created with a radar scanning in elevation is able to measure in one horizontal direction a piece of the horizontal variability of the VPR. At low elevation angles, the distances are much greater than at high elevation. Attenuation, beam broadening, difference in aspect angles are aspects which need to be taken into account. A VPR derived from a volume scanning radar should be able to measure a (coarse) spatial variability. Again, the same aspects of a radar scanning in elevation have to be taken into account. Therefore, the elevation scanning radar and volume scanning radar will be discussed together.

**Vertically oriented radar** Several papers have been published about using the VPR acquired by a vertically oriented radar to detect the melting layer. The earliest paper found is from Fabry and Zawadzki (1995). They

used a vertically oriented X-band radar and needed an algorithm to detect the melting layer to observe the long-term statistics. The method they used is based on the curvature of the reflectivity profile. Using the curvature, they were able to determine the top and bottom of the bright band. In between the top and bottom determined by the curvature, the peak reflectivity is determined as the bright band peak.

Another attempt in detecting the bright band had been developed by Tilford et al. (2001). They used the vertical gradient to determine the bright band characteristics. This can be seen in Figure 2.7. From the vertical gradient, the local maxima and minima are deduced. They represent the upper and lower limb of reflectivity. The crossing of the y axis between the upper and lower limbs is used as the peak reflectivity point. The only thing which is not mentioned in the paper is what happens when there is no bright band. The maxima and the minima of the gradient will still be there if there is no bright band, but it will not represent the bright band.

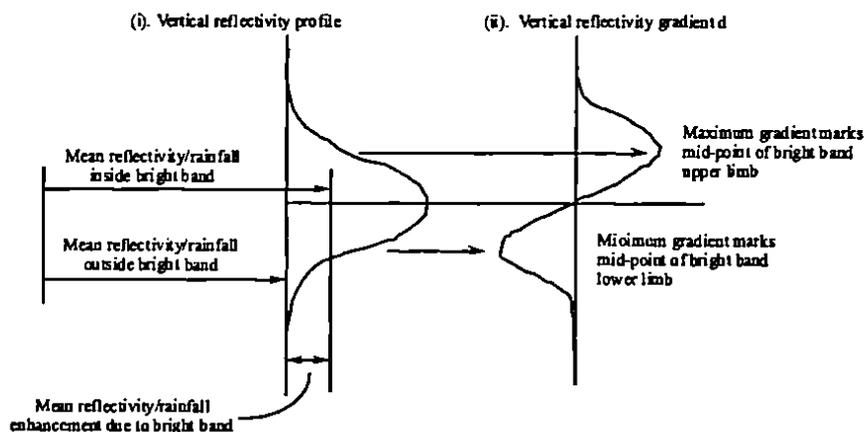


Figure 2.7: Vertical reflectivity profile (left) and gradient of the vertical reflectivity profile (right). (image from Tilford et al. (2001))

Rico-Ramirez and Cluckie (2007) have looked at a slightly different method of detection again using a vertically oriented X-band radar. They searched the VPR for the highest reflectivity, which becomes the datum for two new coordinate systems. This maximum reflectivity is obtained by looking at the height derivative and searching for a value equal or close to zero. After the datum translation towards the new datum, a rotation is carried out around the new datum. The rotation angles are different for the upper and lower bright band boundary determination. The rotation angles are determined in such a way that the rotated axis must intercept the reflectivity profile at 800 above and below the peak reflectivity. When this step is done, the bright band boundaries are simply the maxima of the graphs in the new coordinate systems. This can be seen in Figure 2.8. The authors also compared their method to the methods proposed by Tilford et al. (2001) and Fabry and Zawadzki (1995) as described above. Their conclusion was that their method "performs more reliably than the algorithms presented by Tilford et al. (2001) and Fabry and Zawadzki (1995)".

The authors of "Detection of the bright band with a vertically pointing K band Radar" (2014) have used three different methods for detecting the radar bright band. They tried to fit the VPR with an analytical representation of a VPR, use the gradient of the reflectivity and they used hydrometeor falling velocity. The latter will be discussed in section 2.2.2. The fitting of the analytical model of the bright band can be seen in Figure 2.9. The model is rather simplistic, but still requires quite some parameters to describe the model. The second method was based on the gradient of the reflectivity profile, almost just like Tilford et al. (2001) had carried out their detection. A rather strange difference can be noticed, because Pfaff et al. (2014) used only the maximum and minimum levels of the gradient as the top and bottom boundary of the melting layer, while these do not coincide with the ML boundaries.

**RHI and PPI radar** Smith (1986) had thought of a method for bright band detection on the basis of a PPI radar using two radar beams at different elevation angles. She used the peaks in reflectivity to determine the bright band peak. This detection method is used in all sectors covered by the radar. Then she uses the fact

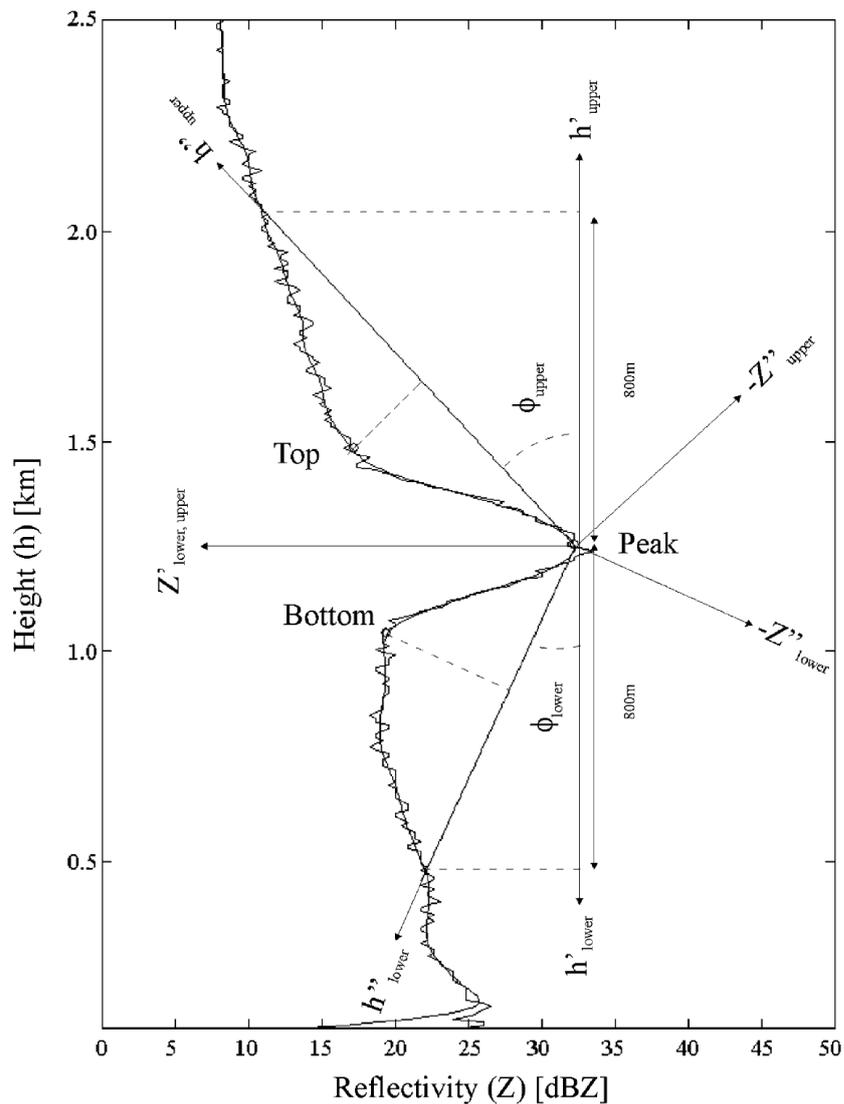


Figure 2.8: Detection of the bright band boundaries using a coordinate transformation. (Image from Rico-Ramirez and Cluckie (2007))

that the bright band is an annular feature in the PPI plot. She therefore seeks averages of the peaks in the different sectors. The average is the mean bright band which can be used for correction procedures.

A different approach is been taken by Cheng and Collier (1993). They looked at the estimated rainfall rates. Their detection method is based on the observed rainfall rate versus the estimated area average rainfall rate. The bright band is causing a severe overestimation of the rainfall rate, which can be compared to the area average rainfall rate. Where this ratio is large, possibly a bright band is present. The detection algorithm is comparing the ratio against a threshold to determine if a bright band is present. The rainfall estimation algorithm was based on reflectivity and thus this algorithm is also based on the reflectivity.

In 1995 Andrieu and Creutin attempted to create an apparent VPR from a horizontally scanning two beam PPI radar. The ratio between the reflectivity of the two radar beams is only dependent on the VPR, but there is no direct analytical relation between the ratio and the VPR. They therefore try to deduce a VPR from the ratio profile. Borga et al. (1997) has evaluated the method based on radar simulations. The VPR retrieved from the method proposed by Andrieu and Creutin (1995) was able to reduce the rainfall rate estimation error by 40%

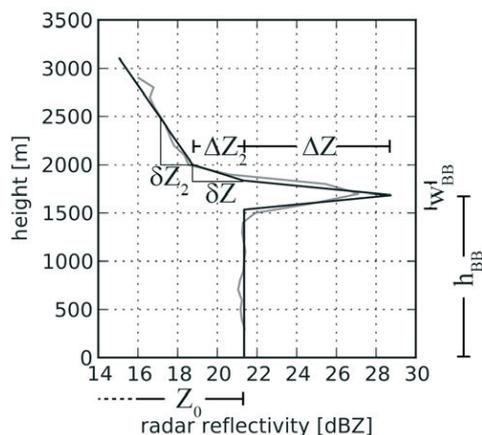


Figure 2.9: Analytical representation of the vertical profile of reflectivity. (Image from Pfaff et al. (2014))

or more in each considered case. The reported downside of the method is that it does not account for spatial variability, since the method generates a mean VPR.

The first article about the use of a volume scanning radar is written by Vignal et al. (1999). They tried to resolve the biggest problem of all the previous methods. The previous papers all assumed that the VPR was valid for the entire radar coverage. Since this is generally not true, correction of rainfall rate is also biased. The authors expanded the theory from Andrieu and Creutin (1995) from a two-elevation beam radar to multi beam radar. This enabled them to deduce local VPRs from the radar data. The radar rainfall estimates improved due to the ability to cope with spatial variations in VPR, but a disadvantage of the proposed method was the limited time resolution. The VPRs were created hourly, so when strong time variability is present, the method fails.

Vignal et al. (2000) has compared three different mechanisms for acquiring the VPR. This paper is not so much about (automatic) detection of the bright band rather than a comparison between previously described methods and their validity in correcting rainfall estimates.

A true automatic detection algorithm based on radar data from a PPI radar has been proposed by Gourley and Calvert (2003). The first step in their detection process is to determine the maximum reflectivity from each elevation radar beam. The value must exceed a certain threshold. The algorithm then searches in the adjacent elevation beams for a 20% reduction in reflectivity. The detection algorithm reports a bright band if there is a layer of high reflectivity from 10 to 30 km from the radar. Two criteria are used to determine this. The first examines the number of grid columns that have at least one bin at any elevation angle with exceeds a certain reflectivity threshold. This count is divided by the maximum amount of grid columns and if this ratio is larger than 6% the first criterion is met. The second criterion is the spatial variability of the bright band height. This criteria is met when the standard deviation of the averaged bright band top height is 500 m or less. When bright band is detected, the bright band is temporally averaged over 30 minutes. Fast changes in VPR will be reduced because of the temporal averaging.

Based on the same radar, another paper has been published by Zhang et al. (2008) on identification of the bright band using the WSR-88D radar. They introduce a three-step procedure to detect the bright band. First, they separate convective rain from stratiform rain events. This step is carried out by searching the radar bin column for a value greater than 50 dBZ or a value greater than 30 dBZ at  $-10^{\circ}\text{C}$  height or above. If either of the cases is true, the rain is classified as convective. All other are classified as stratiform. The second step is to calculate the mean VPR for both the stratiform and the convective groups. After this step the real detection of the bright band is carried out. The detection algorithm searches for the maximum reflectivity below the  $0^{\circ}$  isotherm + 500 m. The  $0^{\circ}$  isotherm is retrieved from a model which is updated hourly. From this maximum the next criterion is that the reflectivity should decrease monotonically by 10% in both directions. The last criterion is to check whether the thickness and symmetry is checked against some fixed values. A disadvantage is that information on the  $0^{\circ}$  isotherm is necessary for this algorithm to work.

More modern single polarisation radar cannot only measure the amount of received power, but are also able to determine the change in frequency induced by a radially moving object. In the next section ML detection based on induced frequency change is explained.

### 2.2.2. Doppler profile

A radar is able to determine the change in frequency induced by a radially moving object if the radar is a coherent radar. These radars are not only able to measure power, but also phase of the transmitted and received signals. The Doppler shift is measured by the phase change between pulses. Doppler shift is thus proportional to the radial (line of sight of the radar) velocity of the object. By extracting this Doppler frequency shift, the radial velocity of an object can be determined. The vertical Doppler profile relates to the fall velocity of particles, but the radial velocity of objects needs to be translated to a vertical velocity to acquire the fall velocity of hydrometeors. For a vertically oriented radar, this is not a problem, because the fall velocity corresponds with the line of sight of the radar. Even then, the fall velocity of hydrometeors might be influenced by vertical winds. In case of a radar which is not elevated to  $90^\circ$ , to be able to determine the fall velocity of hydrometeors knowledge about the horizontal wind field as a function of altitude is required. In a typical vertical velocity profile (without vertical wind) one can imagine that small light ice particles above the ML are falling with a low velocity and when they start to melt, the velocity increases because the ice particles decrease in size and thus reducing drag. When the ice particles are completely melted, the fall velocity will increase further until a limit is reached and the drag of the droplets is equal to gravity. Baldini and Gorgucci (2006) has measured vertical velocity profiles of precipitation. A schematic of the vertical velocity of a stratiform precipitation event is depicted in Figure 2.10. The solid line represents the vertical velocity and one can see that in the ML, the vertical velocity increases until the ML is passed. Then a very slow increase in vertical velocity is still present. The next section will explain the ML detection methods found in literature which use

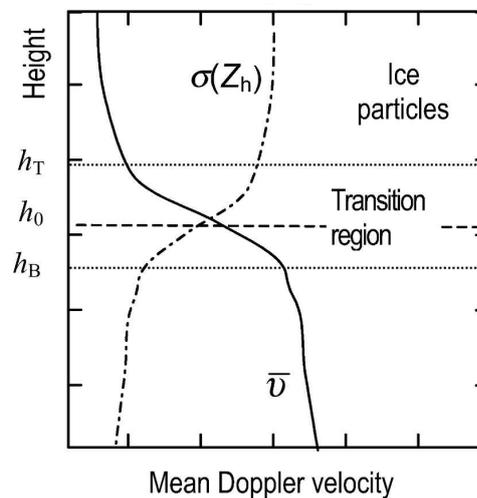


Figure 2.10: Schematic representation of vertical velocity profile. (Image from Baldini and Gorgucci (2006))

the Doppler information to detect the ML.

#### 2.2.2.1 ML detection methods based on Doppler profile

Not many researchers have used vertical velocity as a method of detecting the bright band. The first found paper to make use of velocity information for detecting the bright band is written by White et al. (2002). Their detection is based on the correlation between reflectivity and vertical velocity. In the upper part of the bright band, the reflectivity increases as well as the velocity. This correlates positively. In the lower part of the bright band, the reflectivity decreases and the velocity still increases, which is thus negatively correlated. This can be seen in Figure 2.11. The first step in their algorithm is to detect whether it is raining or not. This step is carried out by using a pair of thresholds. One threshold is for vertical velocity and one threshold is for reflectivity. When 3 or more range gates below 3 km satisfy both criteria, rain is considered present. The bright band detection algorithm is then invoked and analyses (from bottom upwards) the Doppler vertical profile and the vertical profile of reflectivity. A bright band is detected when a velocity decrease is present of  $-1.5$  m/s over 210 m and a reflectivity increase of 2.5 dB over 210 m. The peak reflectivity is then retrieved by searching for the maximum reflectivity in the 525 m above the reflectivity jump.

Another paper which used the vertical velocity has been previously described at the end of section 2.2.1.1. Pfaff et al. (2014) have used vertical velocity to help the detection process. Essentially, they used the same

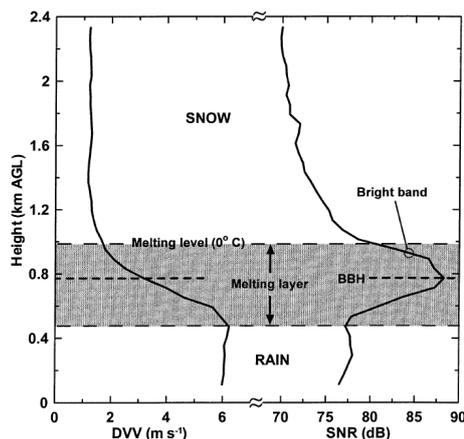


Figure 2.11: Example of vertical Doppler profile (left) and vertical reflectivity profile (right). (Image from White et al. (2002))

procedure as White et al. (2002). They evaluated this procedure against a model least squares fitting and a method which used gradients of the VPR. The method which used the Doppler information provided the best results.

### 2.2.3. Doppler spectral width and Doppler spectral skewness

The Doppler spectral width and spectral skewness have been grouped in this section, since they use the same parameter (Doppler spectrum) and no publications have been found which used these two observables in ML detection and characterisation. Doppler spectral width is the square root of the variance of the Doppler power spectrum, which is the standard deviation  $\sigma$ . In case of homogeneous hydrometeors in steady state a single fall velocity is expected. Since this is never the case as hydrometeors have different shapes and sizes resulting in different fall velocities, widening the Doppler power spectrum. Often this translates to a symmetric distribution of the Doppler power spectrum. The larger the variance, the wider the distribution and the higher is the spectral width. This is graphically shown in Figure 2.12.  $V$  is the falling velocity of hydrometeors and  $S$  is the backscattered signal power.

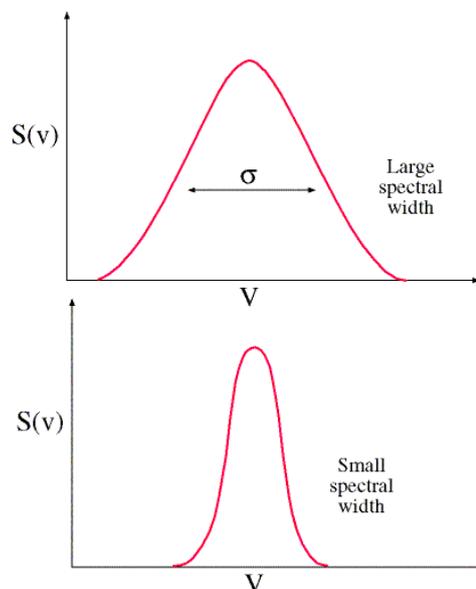


Figure 2.12: Example of Doppler spectral width. (Image from [http://apollo.lsc.vsc.edu/classes/remote/lecture\\_notes/radar/doppler/spectrum\\_width.html](http://apollo.lsc.vsc.edu/classes/remote/lecture_notes/radar/doppler/spectrum_width.html))

The previous explanation on Doppler spectral width assumed a symmetrical distribution of the Doppler power spectrum, but in reality, this is often not the case. The deviation of the measured Doppler power spectrum from a symmetrical power spectrum is called the Doppler spectral skewness. This is graphically shown in Figure 2.13. In this figure the reference symmetrical distribution is shown in the middle while the positive skewed distribution is shown on left side and the negative skewed distribution is shown on the right. The spectral skewness is a unit-less value and is 0 if the distribution is symmetric. Distributions with skewness values below -1 and above 1 are considered highly skewed. The general equation for spectral skewness is the third standardised moment  $s$  and is given in Equation 2.1.

$$s = \frac{E(x - \mu)^3}{\sigma^3} \quad (2.1)$$

Where  $\mu$  is the mean of  $x$ ,  $E$  is the expectation and  $\sigma$  is the standard deviation. In a set of samples the skewness can be calculated using the formula given in Equation 2.2.

$$s = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^3}{\left( \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \right)^3} \quad (2.2)$$

Where  $N$  is the sample size and  $\bar{x}$  is the sample mean. Since Doppler spectral width and skewness are avail-

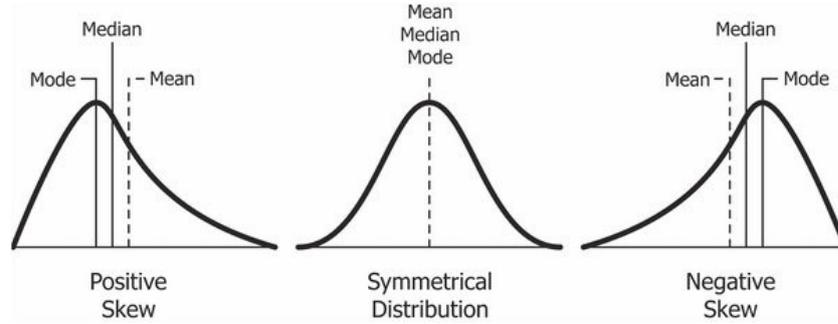


Figure 2.13: Example of Doppler spectral skewness. (Image from <https://codeburst.io/2-important-statistics-terms-you-need-to-know-in-data-science-skewness-and-kurtosis-388fef94eaa>).

able, they will be shown and considered for ML detection and characterisation in the remainder of the thesis. Next the polarimetric observables which are used for ML detection and characterisation are introduced.

### 2.3. Polarimetric radar

A radar which is able to transmit and receive in both horizontal and vertical polarisation directions is called a "polarimetric radar". This gives a wealth of extra features which can be analysed. With a single polarisation radar energy is transmitted and received in one polarisation direction. Interaction of electromagnetic waves with hydrometeors cause a portion of the energy shifts to a different polarisation direction. The backscattered energy is not one complex number (as it is with the coherent single polarisation radar), but can be made up out of four complex numbers according to the backscattering properties of an object. The backscattering components can be written in a matrix:

$$S = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{vh} & S_{vv} \end{bmatrix} \quad (2.3)$$

This matrix is known as the backscattering matrix and is valid for one object. It describes the response of an object to the incident electric field. The first subscript represents the receiving polarisation direction and the second subscript represents the transmit polarisation direction. The backscattering matrix depends on the radar wavelength, incidence angle, axis ratio of the object, volume of the object and relative permittivity. To be able to use the backscattering matrix, a relation with the reflectivity must be established. The relation between the top left backscattering component and the reflectivity is given by:

$$z_{hh} \propto |S_{hh}|^2 \quad (2.4)$$

This applies to the other components of the backscattering matrix as well. This is still only valid for one object, while the radar resolution volume is likely to contain more than one object. Averaging is necessary to filter out the randomness of the response. In addition, we will use the equivalent reflectivity  $z_{e,hh}$  provided by the weather radar where "equivalent" means that the relative permittivity of water is used in the weather radar equation. The relation becomes:

$$z_{e,hh} \propto \langle |S_{hh}|^2 \rangle \quad (2.5)$$

where  $\langle \rangle$  represents averaging. The resulting backscattering power matrix is:

$$P = C \begin{bmatrix} \langle |S_{hh}|^2 \rangle & \langle |S_{hv}|^2 \rangle \\ \langle |S_{vh}|^2 \rangle & \langle |S_{vv}|^2 \rangle \end{bmatrix} \quad (2.6)$$

Using the relation between the equivalent reflectivity and the backscattering power matrix, new polarimetric observables can be explored as is done in the following sections.

### 2.3.1. Differential reflectivity ( $Z_{DR}$ )

One of the new parameters which can be extracted from a polarimetric radar is the differential reflectivity. When a pulse is transmitted in horizontal and received in horizontal direction, one can determine the horizontal backscattered energy. The next pulse is then transmitted in vertical direction and received in vertical direction. From this the backscattered energy is determined from the vertically polarised energy. By dividing the backscattered energy of the horizontally and vertically polarised pulses, the differential reflectivity ( $Z_{DR}$ ) is determined, as can be seen in Equation 2.7.

$$Z_{DR} = 10 \log_{10} \left( \frac{\langle |S_{hh}|^2 \rangle}{\langle |S_{vv}|^2 \rangle} \right) = 10 \log_{10} \left( \frac{z_{e,hh}}{z_{e,vv}} \right) \quad (2.7)$$

Capital Z meaning the  $10 \log_{10}$  of z. The differential reflectivity is related to the shape of the object. This can be seen in Figure 2.14. In the case of Figure 2.14, the horizontally polarised wave will be reflected more than the vertically polarised wave, because the object is larger in the horizontal direction. The differential reflectivity tends to be 0dB when an object is spherical.  $Z_{DR}$  gives information about the axis ratio of the hydrometeors, which are modelled as spheroidal particles. This can be used in rainfall estimation, because

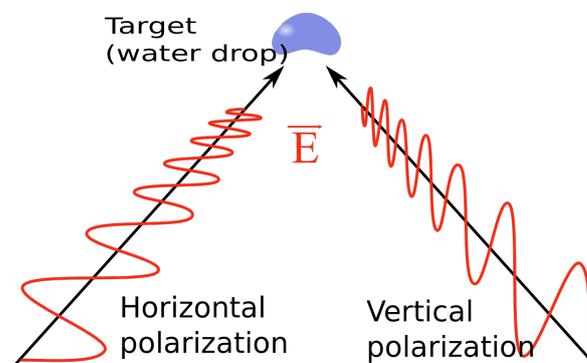


Figure 2.14: Dual polarisation radar used on a non-spherical droplet. (Downloaded From [https://commons.wikimedia.org/wiki/File:Radar-polarisation\\_en.svg](https://commons.wikimedia.org/wiki/File:Radar-polarisation_en.svg)).

the axis ratio of the droplets relates to the size of the droplets. If the differential reflectivity is used in vertical incidence to make a vertical profile of the stratiform events,  $Z_{DR}$  will be close to 0 dB. Looking at precipitation from below, hydrometeors appear to be circular and as a result the  $Z_{DR}$  will be close to 0 dB.  $Z_{DR}$  is therefore not suited for ML detection viewing at  $90^\circ$  elevation.  $Z_{DR}$  can be used for ML detection in other elevation angles, because a significant increase in  $Z_{DR}$  is associated with the ML. In the lower part of the ML, large liquid hydrometeors (and irregular shaped mixed phase hydrometeors) form from the melting particles. When their velocity increases and melting is completed, these large droplets break up and form smaller hydrometeors. The increase in  $Z_{DR}$  is therefore associated with the lower part of the ML. In the next section, ML detection methods which use the ( $Z_{DR}$ ) are being discussed.

### 2.3.1.1 ML detection methods based on Differential Reflectivity ( $Z_{DR}$ )

The first paper found to use  $Z_{DR}$  as a detection method for the bright band is written by Baldini and Gorgucci (2005). They used a vertically oriented C-band radar for their data gathering. Instead of using the  $Z_{DR}$  directly they looked at the second order statistics (standard deviation), because that is related to the  $\rho_{hv}$ . As they explained in their paper the detection performance of using the second order statistics of  $Z_{DR}$  is much better than using a detection based on the reflectivity profile. A possible advantage could be that if a weather radar does not provide the  $\rho_{hv}$  as an output, but does provide  $Z_{DR}$ , one can still use a detection based on the  $\rho_{hv}$  characteristics.

Giangrande et al. (2008) also used  $Z_{DR}$  as a detection method for detecting the bright band, but they only used it complementary to  $\rho_{hv}$  and  $Z_{e,hh}$ . Their radar was a volume scanning radar in the S-band. Pre-processing of the data consisted of averaging the  $Z_{e,hh}$  with a window of 3 resolution cells,  $Z_{DR}$  and  $\rho_{hv}$  with 5 cells. After selecting cells with a threshold on  $\rho_{hv}$  between 0.90 and 0.97, the algorithm searches for a maximum of  $Z_{e,hh}$  and  $Z_{DR}$  within a 500 m window above the marked cells. If the maximum value of  $Z_{e,hh}$  and  $Z_{DR}$  falls between 30 and 47 dBZ and respectively between 0.8 and 2.5 dB, then the cell is considered to be a cell within the melting layer. In this paper the authors used three types of thresholds to detect the melting layer cells. A drawback of using this method is that if one of the parameters does not comply to the threshold, the melting layer might be missed. For instance, the reflectivity  $Z_{e,hh}$  is not always that significant. And in convective systems, the behaviour of the detection algorithm is not (yet) known. It is also questionable if the used limits can be used for different radar types and frequencies, since Boodoo et al. (2010) already used slightly different thresholds.

### 2.3.2. Linear Depolarisation Ratio ( $L_{DR}$ )

The  $L_{DR}$  is similar as the  $Z_{DR}$ , but now the cross polar backscattered energy is compared to either the vertical or horizontally polarised backscattered energy. The  $L_{DR}$  is given by dividing the received cross polar power by the received power at horizontal polarisation, as can be seen in Equation 2.8. This value gives a sense about how much energy has been shifted from horizontal polarisation to vertical polarisation as a result of interaction with hydrometeors.

$$L_{DR} = 10 \log_{10} \left( \frac{\langle |S_{vh}|^2 \rangle}{\langle |S_{hh}|^2 \rangle} \right) \quad (2.8)$$

The  $L_{DR}$  is related to the orientation of particles with respect to the incident electric field. In Figure 2.15 the principle is explained. The top row of the images is when the incident electric field is in parallel with the object and as a result, the radiated electric field has the same polarisation direction. The bottom row represents an object which is tilted with respect to the incident electric field. As a result, the radiated electric field is now not only horizontally polarised, but part of the electric field is radiated with a vertical polarisation. In uniform atmospheres (above and below ML) the  $L_{DR}$  is generally low, but in the ML the  $L_{DR}$  is increased due to the non-uniformity (tumbling) of the melting snowflakes, because they rock and roll as they fall in contrast to rain, which has on average a fixed orientation.  $L_{DR}$  relies on reception of cross polar backscattered energy which is generally very low. This might influence the usability of  $L_{DR}$  for ML detection. Due to the low received cross polar energy, the received energy might be close to the receiver noise level and as a result unreliable results may occur. An example of a  $L_{DR}$  in stratiform rain event is given in Figure 2.16. What can be seen in Figure 2.16 is that the  $L_{DR}$  has a maximum in the ML and could therefore be used to detect the ML. In the next section, ML detection methods found in literature which use the  $L_{DR}$  will be discussed.

#### 2.3.2.1 ML detection methods based on $L_{DR}$

The  $L_{DR}$  measurement has been considered in methods to detect the bright band by several authors. The first who used the  $L_{DR}$  is Bandera et al. (1998). They used data from a RHI radar of unknown frequency for their research. In this paper, the  $L_{DR}$  is only used for detecting the lower boundary of the melting layer, where the reflectivity is used for detecting the upper boundary of the melting layer. The detection of the lower boundary (using the  $L_{DR}$ ) is done making use of the gradient of the  $L_{DR}$ . Once this initial detection had been made, a refinement is carried out using the ratio between two averaged gradients, one at the initial detected lower boundary and one 1 km below the boundary. If this was not possible, a lower value was chosen. This method could run into instability if the lowered value was getting close to the melting layer bottom itself (in case of a very low melting layer).

Rico-Ramirez et al. (2005) also used the  $L_{DR}$ , but only to distinguish between melting snow and rain or snow. They used data from a RHI radar in the S-band. A fuzzy logic-based classifier had been developed

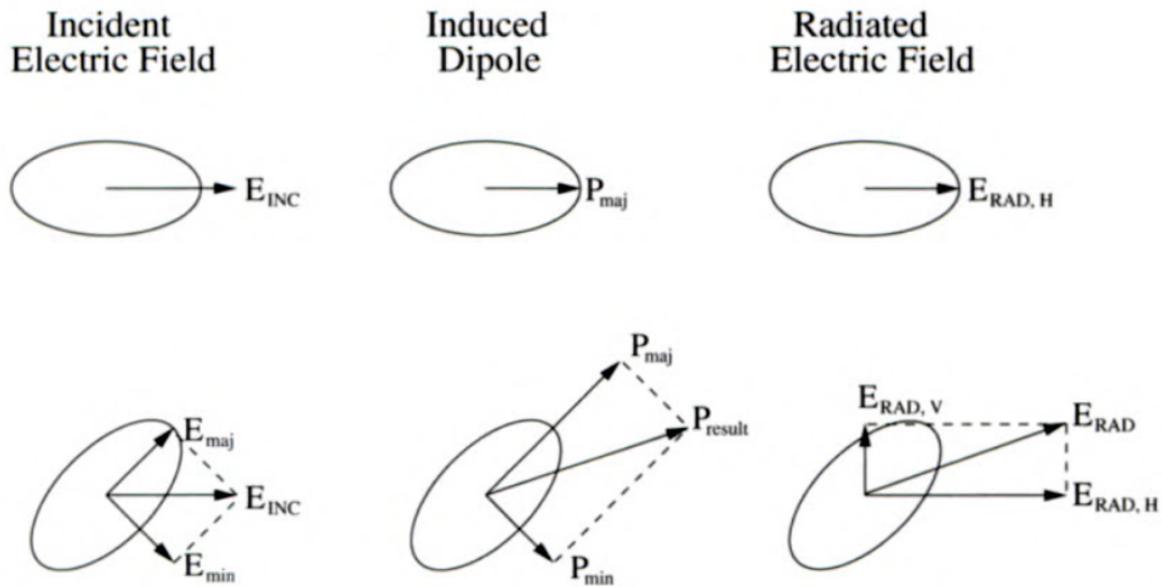


Figure 2.15: Principle of depolarisation of electric field due to orientation. (Image from Meischner (2013) page 138)

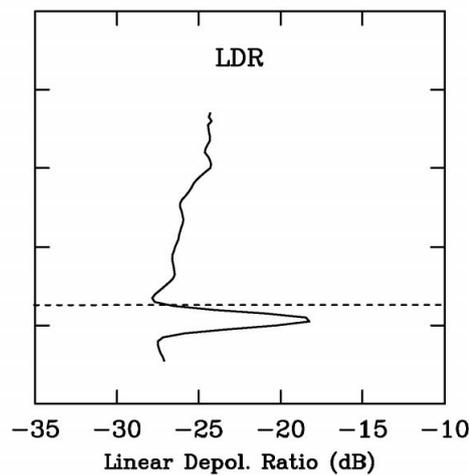


Figure 2.16: Example of a vertical profile of the  $L_{DR}$  in a stratiform event. (Image from Brandes and Ikeda (2004))

which divided the classes based on magnitude of  $L_{DR}$  and the measured height. Since snow and rain had overlapping regions in  $Z_{e, hh}$ ,  $Z_{DR}$  and  $L_{DR}$ ,  $L_{DR}$  is only used to classify melting snow. It marks the transition between snow and rain. Thus, one can assume that above the melting layer snow is present and this is used to change the membership functions of rain and snow.

A later paper is written by Illingworth and Thompson (2011). They also used a S-band RHI radar for data collection. The first step they use is to determine the maximum  $Z_{e, hh}$  within 750 m of the  $0^\circ$  isotherm. If the maximum is within 500 m of the  $0^\circ$  isotherm and has a value above 25 dBZ, this is marked as the bright band peak value. The value of  $Z_{e, hh}$  is then searched 750 m below this height to obtain the  $\Delta Z_{e, hh}$ , the bright band enhancement. If no maximum is found, no bright band is detected in this way. An alternative method is then used: the bright band enhancement is calculated from two values of  $Z_{e, hh}$ ; 250 m and 1 km below the  $0^\circ$  isotherm. The noted values of  $Z_{e, hh}$  are then plotted against the maximum values of  $L_{DR}$  within 750 m of the  $0^\circ$  isotherm. They also found a relation between the bright band enhancement and the maximum value of  $L_{DR}$ , which indicates that a detection algorithm may be based on this conclusion. A disadvantage of this detection method is that knowledge about the altitude of the  $0^\circ$  isotherm is required for this detection algorithm to work.

A quite recent paper from almost the same authors is describing the use of  $L_{DR}$  for distinguishing between stratiform and convective rainfall, Sandford et al. (2017). This paper is only available to the repository staff until the 8th of May 2018.

### 2.3.3. Copolar correlation coefficient ( $\rho_{hv}$ )

The copolar correlation coefficient is a unit-less value between 0 and 1 and indicates the similarity of the polarimetric signature in returned echo from pulse to pulse. Again, a pulse is transmitted in the horizontal direction and next in the vertical direction (or using a slant  $45^\circ$ ). The backscattered energy is observed, the phase angle and magnitude are determined of both the horizontally and vertically polarised waves. If the resulting variance of axis ratio and density of hydrometeors is low, the copolar correlation coefficient will be close to unity. The copolar correlation coefficient therefore relates to the uniformity of particles within the radar resolution cell. The copolar correlation coefficient is calculated with the following equation.

$$|\rho_{hv}| = \left| \frac{\langle S_{vv} S_{hh}^* \rangle}{\sqrt{\langle |S_{hh}|^2 \rangle \langle |S_{vv}|^2 \rangle}} \right| \quad (2.9)$$

The copolar correlation coefficient is related to the variability of the polarimetric signature of scatterers. The copolar correlation of rain and ice will be close to unity, but in the ML it will not. Rain and ice particles are quite uniform and stay uniform in time. Therefore, there is not much difference in magnitude and phase from pulse to pulse. In the ML, there will be a difference in magnitude and/or phase between horizontal and vertical polarised waves due to the non-uniformity from pulse to pulse of particles in the ML. The parameter itself is *unaffected* by attenuation. Although the parameter itself is unaffected by attenuation, a significant amount of backscattered signal must be present. So, when the received signal is reaching levels close to the receiver noise level, the  $\rho_{hv}$  is not reliable any more. Another challenging situation occurs when dendrites grow, the  $\rho_{hv}$  could reach the same values as is being used to detect the melting layer. Dendrites form generally well above the melting layer in temperatures between  $-12^\circ\text{C}$  and  $-18^\circ\text{C}$ . An example of a vertical profile of the  $\rho_{hv}$  in case of a stratiform event can be seen in Figure 2.17. In the figure, it is visible that above and below the ML,

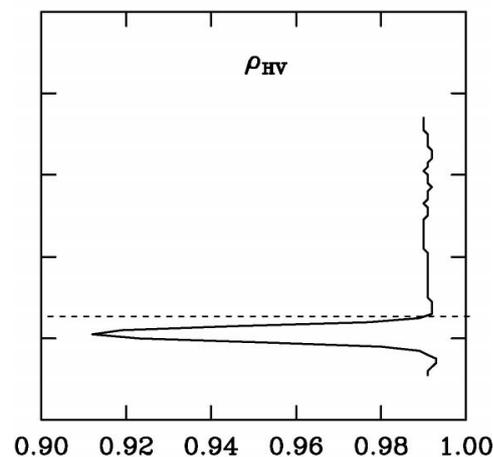


Figure 2.17: Example of a vertical profile of the correlation coefficient in a stratiform event. (Image from Brandes and Ikeda, 2004)

the  $\rho_{hv}$  is close to unity, while in the ML it is not. It shows a dip in the ML as a result of the mixture of differently shaped and sized hydrometeors. The copolar correlation coefficient is positioned below the reflectivity peak due to the fact that it is not reacting on the melting particles, but on non-uniformity. Hydrometeors first need to melt (partially) to have an effect on the copolar correlation coefficient. Therefore, the altitude of the bright band and the altitude of the copolar correlation coefficient "dark band" do not entirely coincide. The band is called the "dark band" due to the decrease of copolar correlation coefficient, instead of the increase of other observables. Since the copolar correlation coefficient is showing a detectable signature in the ML it is very usable for ML detection. Detection methods based on the  $\rho_{hv}$  will be discussed next.

### 2.3.3.1 ML detection methods based on copolar correlation coefficient ( $\rho_{hv}$ )

The use of  $\rho_{hv}$  for detection of the melting layer was already recognized by Zrnić et al. (1994). They concluded that the copolar correlation coefficient is a very usable observable to detect the melting layer.

It was until 2004 that the copolar correlation coefficient was used by Brandes and Ikeda to first estimate the freezing level height. They fitted the measured vertical profiles with a model profile, from which the maximum signature was taken. This fitting to a model is a method which could be used to detect the melting layer, although it is not used for this purpose in this paper. They also found a weak relation between the height of the  $\rho_{hv}$  minimum and the  $Z_{e,hh}$  maximum value. The higher the maximum  $Z_{e,hh}$ , the bigger the separation between the maximum  $Z_{e,hh}$  and the minimum  $\rho_{hv}$ .

Matrosov et al. (2007) used the  $\rho_{hv}$  to identify melting layer. Their method is used in conjunction with a horizontally scanning radar with a 3° elevation beam. They first filtered out the ground clutter and then the  $\rho_{hv}$  should descent stably below a threshold of 0.95. When  $\rho_{hv}$  is then ascending stably above 0.9, the limits of the melting layer are identified.

A different approach of using  $\rho_{hv}$  is taken by Tabary et al. (2006). They only used the  $\rho_{hv}$  for characterising the melting layer. Their method to detect the melting layer is based on a theoretical model of the  $\rho_{hv}$  profile. The model is only dependant on the bright band thickness and the freezing level height. Using these two parameters, they simulate a profile with 200 m increments on the height and bright band thickness. Each pair generates a slightly different profile, which is then correlated with the measured profile. The profile with the highest correlation is selected and thus the two parameters are known, which characterise the melting layer.

As already described in section 2.3.1.1, Giangrande et al. (2008) used the  $\rho_{hv}$  to pre-select cells which could be ML cells. The criterium was a threshold between 0.9 and 0.97. These cells were selected for further investigation based on  $Z_{DR}$  and  $Z_{e,hh}$ . A similar approach has been taken by Boodoo et al. (2010). They used the same principles as Giangrande et al. (2008), but had some different thresholds. The threshold on the  $\rho_{hv}$  was adopted. Another paper to classify winter precipitation is written by Thompson et al. (2014). The melting layer part is classified using the same method as proposed by Boodoo et al. (2010).

### 2.3.4. Differential propagation phase ( $\Phi_{DP}$ ) and specific differential phase ( $K_{DP}$ )

Differential propagation phase and specific differential phase can be derived from comparing the difference in phase angle from the horizontally and vertically polarised received waves. It originates from the fact that electromagnetic waves penetrating some object are delayed. When looking again at Figure 2.14, it can be seen that the horizontally polarised wave has to penetrate more liquid than the vertically polarised wave. This causes the horizontally polarised wave to be delayed more than the vertically polarised wave. The delay is proportional to the amount of liquid to be penetrated. The delay causes a phase shift w.r.t. to the vertically polarised wave. This phase shift is called the differential propagation phase ( $\Phi_{DP}$ ).  $K_{DP}$  is the range derivative of the  $\Phi_{DP}$ . It indicates the slope of differential phase and this value is related to the concentration of particles in the medium. This method has a downside because a good estimation of  $K_{DP}$  requires a significant number of particles and hydrometeors need to be oblate. Therefore, it only works if a significant quantity of rain is falling or a large distance through the medium is covered. The main application of the  $K_{DP}$  parameter is rainfall rate estimation.

Generally said, the higher elevation the less apparent the polarimetric parameters become. Polarimetric radars use the difference in properties when the particles are interacting with either horizontally or vertically polarised waves. As already mentioned in the previous sections, hydrometeors appear circular from below and therefore the difference in reflection between the vertically and horizontally polarised waves is decreasing as the elevation angle is increased. Since the differential propagation phase and specific differential phase are based on the concentration of non-spherical hydrometeors, the differential propagation phase will not be very high at high elevation angles. Both the horizontally and vertically polarised waves have to penetrate almost the same amount of medium. A way of using a vertically oriented polarimetric radar to still be able to use it for ML detection is by looking at the standard deviation of the measured differential phase. This method is proposed by Baldini and Gorgucci (2006). On average the differential propagation phase will not be much different in rain or snow, but the standard deviation is different in rain or snow. This could be used in detecting the ML. An example of the vertical profile of differential phase standard deviation is shown in Figure 2.18. Therefore,  $\Phi_{DP}$  and  $K_{DP}$  are probably not the best parameters to detect the ML at any elevation angle. However,  $\Phi_{DP}$  and  $K_{DP}$  are very suitable in heavy rainfall situations, where there is a large phase difference between horizontally and vertically oriented waves at lower elevation angles.

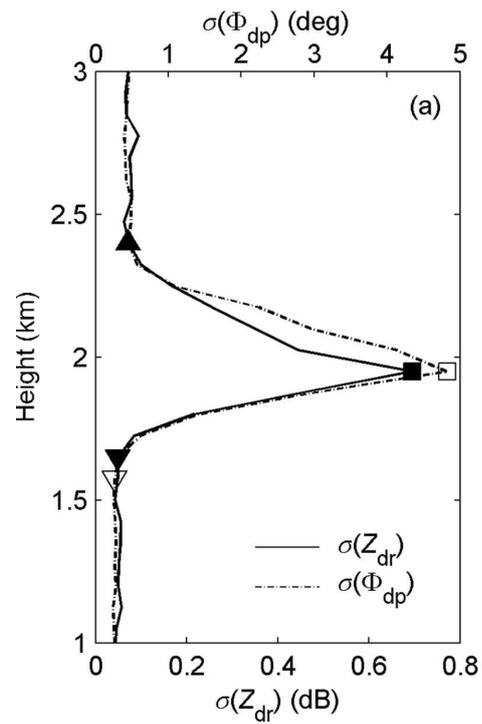


Figure 2.18: Example of a vertical profile of the differential phase standard deviation in a stratiform event. (Image from Baldini and Gorgucci (2006))

#### 2.3.4.1 ML detection methods based on Differential propagation phase ( $\Phi_{DP}$ ) and specific differential phase ( $K_{DP}$ )

No papers have been found which describe the use of  $\Phi_{DP}$  and  $K_{DP}$  as a measure to detect the melting layer. An example of a stratiform rain event is depicted in Figure 2.19a. In this figure it is visible that the  $K_{DP}$  does not have a significant characteristic, while in this situation a ML was present as can be seen in the right image where  $Z_{DR}$  of the same situation is displayed.

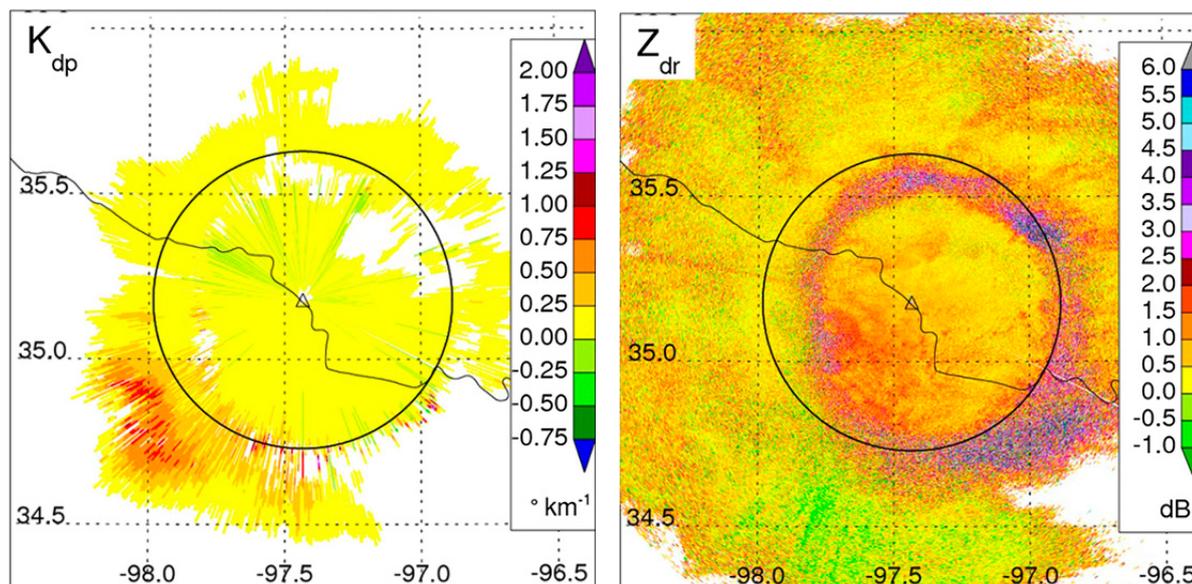
(a)  $K_{DP}$  in a stratiform rain event.(b)  $Z_{DR}$  in the same stratiform rain event. A ML is clearly visible in the PPI plot of the  $Z_{DR}$ .

Figure 2.19: Two images of the same stratiform rain situation. The left image shows  $K_{DP}$  and the right image shows  $Z_{DR}$ . In this situation a ML was present as can be seen in the right image, but in the left image  $K_{DP}$  is not showing a ML characteristic. (images from Thompson et al. (2014))

## 2.4. Conclusion on Melting layer detection methods

When looking at the papers which have been written on the subject, one can see a shift from using the reflectivity towards polarimetric products for detection of the melting layer. Reflectivity only is not suitable in all conditions, sometimes there is no clear reflectivity peak, while there is a melting layer. Also, the reflectivity of heavy rainfall can be just as pronounced as the reflectivity of the bright band in the melting layer. The use of Doppler information is useful in trying to detect the melting layer, but Doppler information is often contaminated with (vertical) wind effects. To improve detection of the melting layer one should turn towards polarimetric products of radar. Different authors used the  $L_{DR}$  in detecting the melting layer, but the  $L_{DR}$  requires one measurement of a strong signal (same polarisation direction for transmission and reception) and the measurement of a weak signal (cross polarisation measurement). This causes issues when the signal is reaching the sensitivity level of the radar, which occurs when distances are long.  $Z_{DR}$  is another parameter which is being used for melting layer detection.  $Z_{DR}$  is depending on orientation and shape of the hydrometeors, but also depends on radar orientation. The hydrometeors are modelled as spheroidal particles, mainly oblate or spherical.  $Z_{DR}$  increases with the oblateness of the hydrometeors viewed from the side by the radar. However,  $Z_{DR}$  tends to go to 0 dB when viewed from below. Another disadvantage is that  $Z_{DR}$  in certain circumstances is also significant above the melting layer. The copolar correlation coefficient  $\rho_{hv}$  is one of the most used polarimetric parameters for melting layer detection. Multiple authors use this parameter to detect the melting layer. The parameter itself is significantly pronounced within the melting layer and is unaffected by attenuation, which is not the case for  $Z_{DR}$  and  $L_{DR}$ . Concluding, the method of Giangrande et al. (2008) will be used as the reference method to which the developed ML detection and characterisation algorithms in this thesis are compared. This method is chosen because it is a recognized method, it is not too difficult to implement and it uses multiple radar observables which are suited for ML detection. The method by Rico-Ramirez and Cluckie (2007) is going to be used to form a ML boundary definition.



# 3

## Data

Attention is given to extract the melting layer signature using Doppler and polarimetric radar data obtained in the case of slantwise and vertical profiling. At the end a conclusion is drawn on the usability of radar observables for automatic processing.

### 3.1. Tara and ACCEPT measurement campaign



Figure 3.1: TARA radar. (Image from [www.cloud-net.org/instruments/documents/Tara.pdf](http://www.cloud-net.org/instruments/documents/Tara.pdf))

Data from TARA (Transportable Atmospheric RADar, Figure 3.1) during the ACCEPT campaign will be used for the development of the algorithms. The TARA specifications are listed in Table 3.1. During the ACCEPT campaign, which ran from 08-10-2014 until 14-11-2014, the radar was fixed at an elevation angle of 45°. Before the campaign, one day was selected for the calibration measurements where the radar pointed at 90° elevation.

During the campaign daily logs and overviews were made. Also, on specific moments radiosondes were launched to measure temperature profiles. During the campaign multiple events were recorded including stratiform rain events with a distinct bright band, convective events and stratiform events with slight embedded convection. Since the algorithm has to deal with any type of weather, all the data will be used. The data will be divided into four different categories to be able to select different circumstances:

1. Scenarios with constant significant melting layer;
2. Scenarios with interrupted melting layer;
3. No stratiform scenarios, thus without melting layer (rest of the data);

These categories of data are not composed on a fixed set of rules but on the judgement of the author. The first category contains data where a significantly pronounced ML signature was present over a longer period of time (in the order of hours) and the ML was not changing altitude. The intensity of the ML and rain are allowed to change, as long as the ML remains significant. The second ML scenario was composed of data where a significant ML was present, but did not last long (much less than one hour). This ensured that the ML could not display a flat horizontal line. The last scenario contained the rest of the data, without stratiform

Table 3.1: TARA specifications during the ACCEPT campaign

Characteristic	Value	Remarks
<b>Type</b>	<b>FM-CW</b>	
Central frequency	3.298 GHz	
Max. transmitted power	100 W Out of amplifier	Automatic decrease by steps of 10 dB in case of receiver saturation
<b>Signal generation</b>		
Sweep time $T_s$	0.5 ms	
Range resolution	30 m	Can be changed
# range bins	512	
<b>Polarimetry</b>		
Polarisation	VV HV HH	Main beam only
Measurement cycle	VV HV HH OB1 OB2	
<b>Doppler</b>		
# of Doppler bins	512	
Resolution	3.6 cm/s	
Max. velocity	$\pm 45.5$ m/s	Main beam after spectral polarimetric dealiasing
Max. velocity	$\pm 45.5$ m/s	offset beams after spectral dealiasing
Max unambig. velocity	$\pm 9.1$ m/s	
<b>Antenna</b>		
Beam width	2.1°	
Gain	38.8 dB	
Near field	$\leq 200$ m	
<b>Beams</b>	elevation	azimuth related to North
Main beam	45°	246.5°
Offset beam 1	60°	246.5°
Offset beam 2	43.1°	267.3°
<b>Clutter suppression</b>		
Hardware	Antennas	low side lobes
Processing	Doppler spectrum	Spectral polarimetry (main beam) and spectral (offset beam)

precipitation. The first category of scenarios will be used to identify the specific signatures of the melting layer using the different (polarimetric) radar observables. The second and last scenarios will be used to test the developed algorithms.

During the campaign TARA measured the atmosphere at a fixed 45° elevation angle. Almost everyday measurements were carried out recording the following observables:

- $Z_{e,hh}$ ;
- $Z_{e,hv}$ ;
- $Z_{DR}$ ;
- $L_{DR}$ ;
- $\rho_{hv}$ ;
- $\Psi_{dp}$ ;
- Vertical velocity;
- Doppler spectral width;
- Doppler spectral skewness.

As mentioned, TARA was pointing at 90° elevation during the calibration day. The recording made during that day also contained stratiform rain with a significant melting layer present. Unfortunately, the Doppler related observables were not readily available for that day. At 90° elevation the Doppler measurement should deliver a direct measurement of the vertical velocity of hydrometeors. Doppler measurements only depend on the

(radial) velocity of the hydrometeors and not on the shape. The Doppler measurements at 45° elevation could be influenced by horizontal winds, while the 90° elevation measurements are not. On the basis of this reasoning I can conclude that the Doppler based observables are just as usable at 90° elevation as they were at 45° elevation. The available observables at 90° are:

- $Z_{e,hh}$ ;
- $Z_{e,hv}$ ;
- $Z_{DR}$ ;
- $L_{DR}$ ;
- $\rho_{hv}$ ;
- $\Psi_{dp}$ ;

The 90° measurement can give a valuable insight in how different radar observables behave under a different orientation. As explained in section 2.3.1 we expect the  $Z_{DR}$  to become 0 dB since a hydrometeor observed from below appears spherical. The same can be said about the differential propagation phase, the differential propagation phase will remain about 0°.

### 3.2. Acquiring mean and standard deviation of ML

The purpose of determining the mean and standard deviation of ML profiles is to find melting layer characteristics with different radar observables and determine the variety of these characteristics. On the basis of that knowledge a profound understanding of the different melting layer characteristics of radar observables is build. Making use of that knowledge base a profound choice can be made on which radar observables can be best used for melting layer detection. To be able to acquire the mean and standard deviation, a four-step procedure is created:

- Smoothing of data;
- Selection of data;
- Alignment of profiles;
- Determine mean and standard deviation of ML signature (in linear domain).

**Smoothing of data** The data retrieved from TARA is already processed data, however they are not averaged and therefore exhibit statistical variations. To prepare the data for further processing, averaging using a moving window over 5 profiles is carried out (15 seconds). The choice of 5 profiles is a trade-off between reduction of the radar observable standard deviation and time resolution. Averaging in the vertical direction is performed and is dependent on the actual variability of the radar observable itself. Averaging over the vertical is therefore mentioned in the subsections of section 3.2.1 accordingly.

**Selection of data** Once averaged data is available, a part of the data is selected for further processing. Only data is selected where the melting layer is clearly present (use the scenarios with a clear ML signature). The selected data ranges from 0 m altitude to 3500 m altitude, since no ML occurred above 3500 m in the clear ML signature scenarios. Rain and the direct surroundings of the ML is included in the data selection, because it does not influence the next step and it gives valuable insight in the difference between the ML and its immediate surroundings.

**Alignment of profiles** To be able to calculate the mean and standard deviation of the ML profiles they need to be aligned to some reference height. Since the ML height is variable in time the ML signatures need to be aligned. The alignment is performed for different observables according to Table 3.2. For  $\Psi_{dp}$  it was not possible to find a good measure to create a reference height.  $\Psi_{dp}$  therefore uses the reference height of  $\rho_{hv}$  to align the  $\Psi_{dp}$  profiles. This is not a big problem, because it still shows the mean and standard deviation of the correct  $\Psi_{dp}$  profile, but the true reference height of the  $\Psi_{dp}$  profiles is not obtained. When the relative height is compared between the observables in section 3.3, the height of the  $\Psi_{dp}$  mean profile will be the same as the height of the  $\rho_{hv}$  mean profile due to this choice. The alignment procedure of the profiles can be seen in Figure 3.2. The profiles are aligned using the height of the maximum  $Z_{e,hh}$ . The same approach is applied on the other observables using table Table 3.2 to select the reference height.

Table 3.2: Reference height for alignment of vertical profiles

Observable	Ref. height
$Z_{e,hh}$	Maximum
$Z_{e,hv}$	Maximum
$Z_{DR}$	Maximum
$L_{DR}$	Maximum
$\rho_{hv}$	Minimum
$\Psi_{dp}$	Minimum $\rho_{hv}$
Vertical velocity	Maximum first derivative
Doppler spectral width	Minimum first derivative
Doppler spectral skewness	Minimum first derivative

**Determination of ML mean and standard deviation** With the profiles aligned, the mean and standard deviation of the radar observable under investigation in each range cell can be determined accordingly. For profiles which have their values in dB the mean and standard deviation calculation is carried out in linear scale and then converted back to dB scale. The purpose of calculation of the mean and standard deviation is to understand the variability of the radar observable in and around the melting layer. In the next section, the above described procedures are applied to the data. First the mean and standard deviation of the ML signature of the radar measuring at 45° elevation is discussed and afterwards attention is given on the 90° elevation data.

### 3.2.1. Mean and standard deviation of ML signature at 45° elevation

The study case scenario consists of 1984 vertical profiles. The data from TARA was already transformed to vertical profiles by TARA processing. For every observable, the mean, the mean + standard deviation and mean - standard deviation curves will be given to examine the ML signature.

#### Horizontal equivalent reflectivity factor ( $Z_{e,hh}$ )

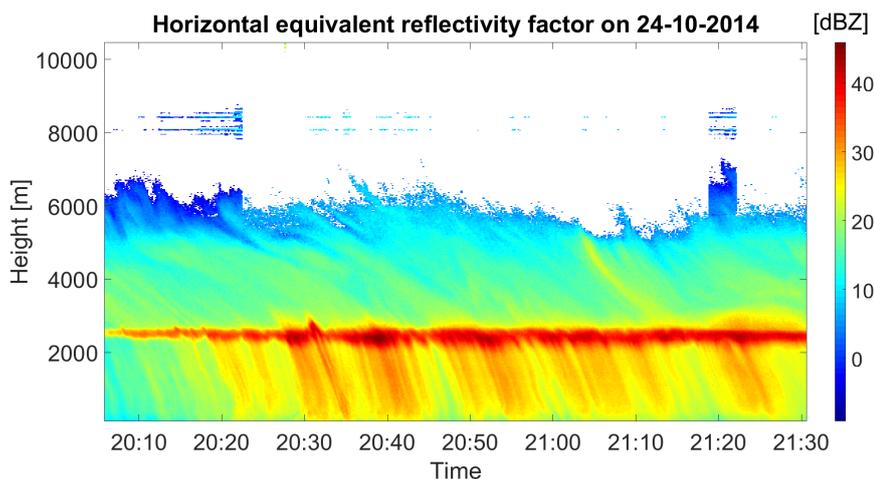
The first radar observable to be analysed in the melting layer is the horizontal equivalent reflectivity factor ( $Z_{e,hh}$ ). The equivalent reflectivity factor used for analysis can be seen in Figure 3.3a. No vertical averaging has been applied for  $Z_{e,hh}$ . A clear bright band signature of the melting layer is visible around 2500 m altitude. The mean and standard deviation are shown in Figure 3.3b. The height in that figure is no longer absolute height, but relative height with respect to the maximum reflectivity. When we take a look at the melting layer peak and how much it differs from its surroundings, we can see that the mean peak value is 39 dBZ, the mean value below the melting layer is 26 dBZ and above the melting layer it drops at least until 18 dBZ. In terms of change with respect to the peak value this is a drop of 13 dBZ below the melting layer and 21 dBZ above the melting layer. The increase of  $Z_{e,hh}$  due to the ML is significant and usable for ML detection.

#### Cross polar equivalent reflectivity factor ( $Z_{e,hv}$ )

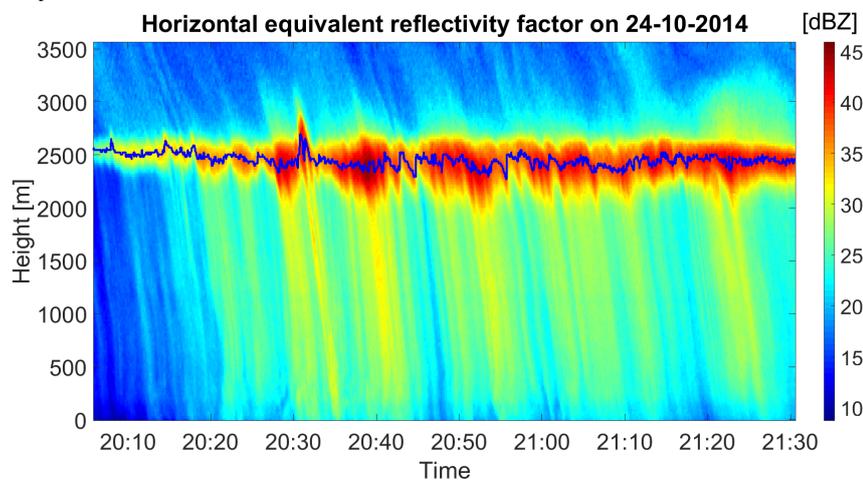
The same analysis has been carried out for the cross polar equivalent reflectivity factor. No vertical averaging has been applied for  $Z_{e,hv}$ . The peak of the cross polar reflectivity is about 20.5 dBZ while the values above and below the melting layer are respectively 0.5 dBZ and 2 dBZ. The difference between the mean peak value and above the melting layer is thus 20 dBZ and below the melting layer 18.5 dBZ. Note that the mean backscattered power above and below the ML is almost equal. The increase of  $Z_{e,hv}$  due to the ML is significant and therefore usable for ML detection. Also note that the backscattered cross polar power cannot be measured far above the ML, since the SNR becomes too low.

#### $Z_{DR}$

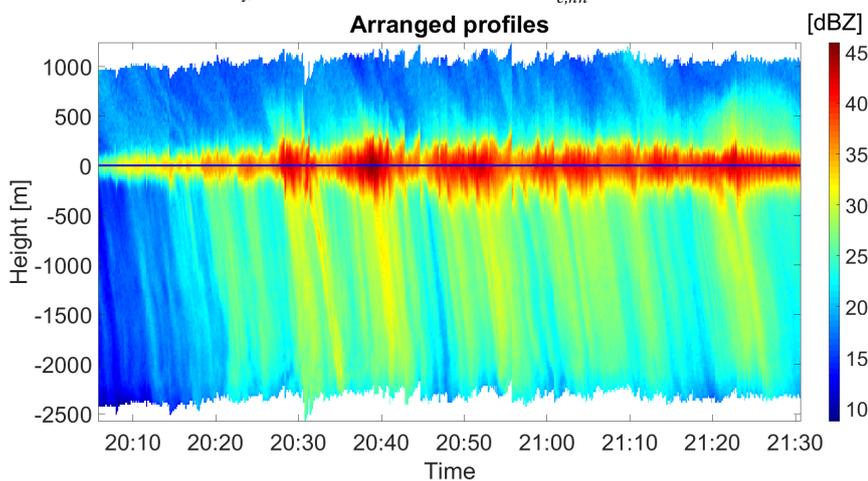
Again, the alignment procedure is carried out on the  $Z_{DR}$  for the same scenario. No vertical averaging has been applied on the profiles of  $Z_{DR}$ . Again, the maximum value for the  $Z_{DR}$  was used to align the peak of the melting layer to be able to determine the signature of  $Z_{DR}$  in the ML. The result can be seen in Figure 3.5a and Figure 3.5b. When we take a look at the ML peak and how much it differs from its surroundings, we can see that the mean peak value is 1.4 dB, the mean value below the ML is 0.48 dB and above the ML 0.3 dB. The



(a) Complete dataset



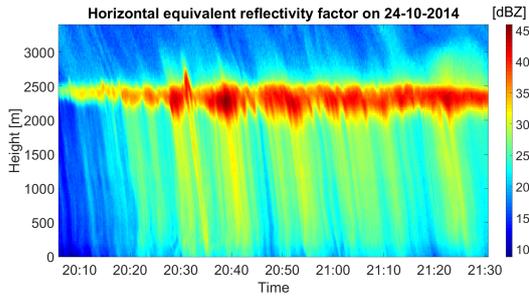
(b) Hand selection of ML data only, the blue line indicates the maximum  $Z_{e, hh}$



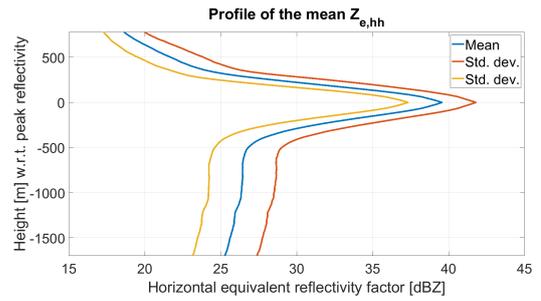
(c) Result of alignment procedure, the reference height corresponds to the maximum  $Z_{e, hh}$

Figure 3.2: ML profile alignment procedure on data from 24-10-2014. The maximum  $Z_{e, hh}$  provides the reference height, every profile is shifted in height to match that reference.

$Z_{DR}$  is slightly higher below the ML than above the ML. Based on this data, the conclusion can be drawn that  $Z_{DR}$  shows a significant increase in the ML and is thus suitable for ML detection.

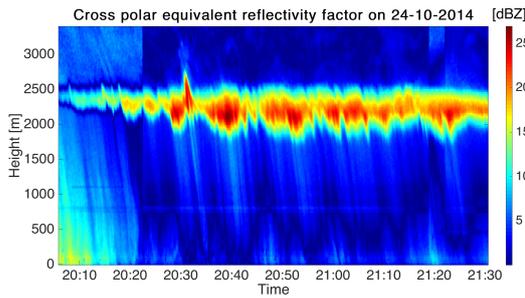


(a) Image of the horizontal equivalent reflectivity factor on the 24th of October 2014 during the ACCEPT campaign. A clear bright band signature is visible between 2200 m and 2700 m altitude.

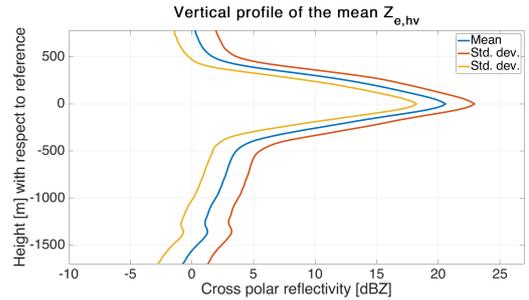


(b) Mean and standard deviation of the melting layer equivalent reflectivity factor profiles.

Figure 3.3: Reflectivity in a clear ML signature scenario

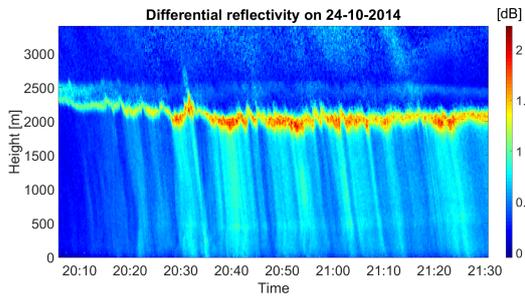


(a) Image of the cross polar equivalent reflectivity factor on the 24th of October 2014 during the ACCEPT campaign. A clear bright band signature is visible between 2200 m and 2700 m altitude.

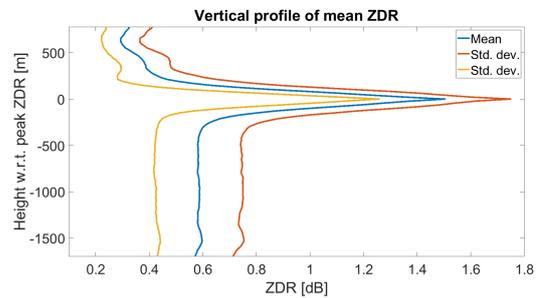


(b) Mean and standard deviation of the melting layer equivalent cross polar reflectivity factor profiles.

Figure 3.4: Cross polar equivalent reflectivity factor in a clear ML signature scenario



(a) Image of the  $Z_{DR}$  on the 24th of October 2014 during the ACCEPT campaign. A clear signature is visible between 2100 m and 2400 m altitude.



(b) Mean and standard deviation of the ML  $Z_{DR}$  profiles.

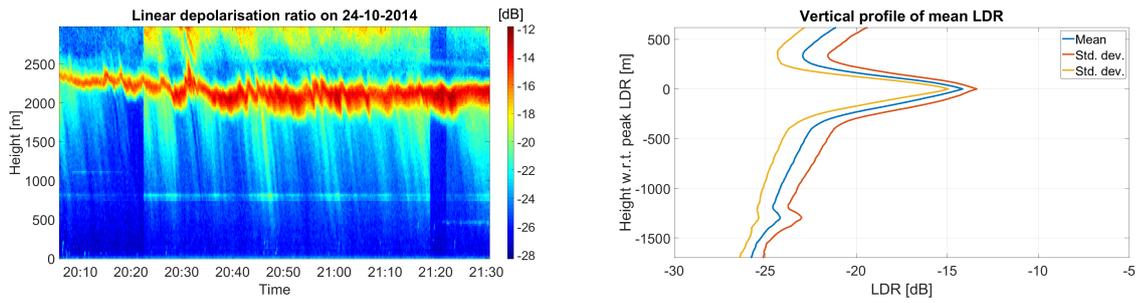
Figure 3.5:  $Z_{DR}$  in a clear ML signature scenario

### $L_{DR}$

The next radar observable is the  $L_{DR}$ . No vertical averaging has been applied for  $L_{DR}$ . The result is shown in Figure 3.6a and Figure 3.6b. When we take a look at the ML peak and how much it differs from its surroundings, we can see that the mean peak value is -14 dB, the mean value below the ML is -23 dB and above the ML -23 dB. Note that we see the noise-to-signal ratio instead of  $L_{DR}$  in the precipitating cloud part of Figure 3.6a.  $L_{DR}$  is therefore usable for ML detection, only care must be taken to prevent the inclusion of the upper part of the clouds.

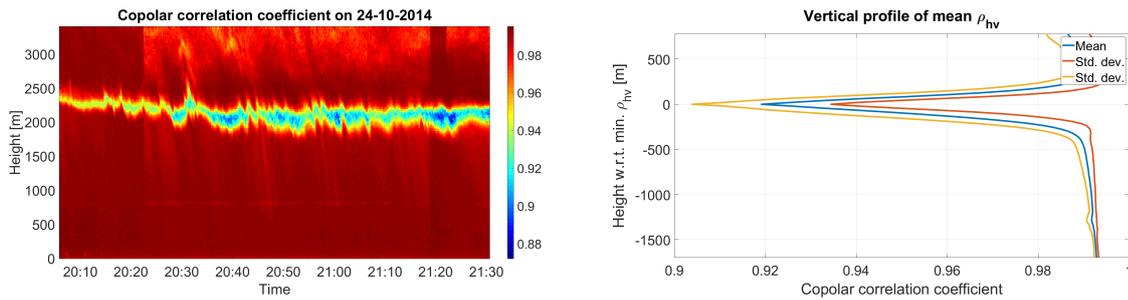
### $\rho_{hv}$

When the procedure is applied to  $\rho_{hv}$ , again no vertical averaging is applied. The difference with respect to the previous procedures is the fact that to align the peaks of the ML signatures, the minimum is used instead of the maximum. The result is shown in Figure 3.7a and Figure 3.7b. When we take a look at the ML dip and



(a) Image of the  $L_{DR}$  on the 24th of October 2014 during the ACCEPT campaign. A clear signature is visible between 2000 m and 2500 m altitude. (b) Mean and standard deviation of the ML  $L_{DR}$  profiles.

Figure 3.6:  $L_{DR}$  in a clear ML signature scenario



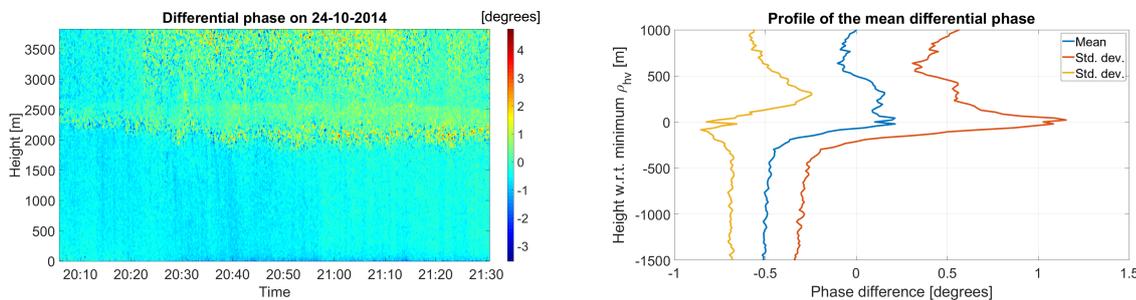
(a) Image of the  $\rho_{hv}$  on the 24th of October 2014 during the ACCEPT campaign. A clear signature is visible between 2000 m and 2500 m altitude. (b) Mean and standard deviation of the ML  $\rho_{hv}$  profiles.

Figure 3.7:  $\rho_{hv}$  in a clear ML signature scenario

how much it differs from its surroundings, we can see that the mean dip value is 0.92, the mean value below the ML is 0.99 and above the ML 0.99. Note that the decrease in  $\rho_{hv}$  in the top part of the precipitating cloud is due to the decrease in signal to noise ratio. The  $\rho_{hv}$  is usable for ML detection, but in our case, care must be taken to prevent the upper part of the data from being included in the ML characterisation algorithms.

$\Psi_{dp}$

The procedure is also applied to the  $\Psi_{dp}$ . It was quickly seen that this was more difficult to perform, because the signature in the ML was not very clear. To be able to align the profiles of the ML, the alignment of the copolar correlation coefficient was used. By doing so, relative heights between copolar correlation coefficient and  $\Psi_{dp}$  signatures were assumed zero and no averaging was necessary in vertical direction because the copolar correlation was used. To make the ML better visible in Figure 3.8a the colorbar had to be truncated from -3 to 5 degrees. The result is shown in Figure 3.8a and Figure 3.8b. The  $\Psi_{dp}$  is not showing a clear



(a) Image of the  $\Psi_{dp}$  on the 24th of October 2014 during the ACCEPT campaign. A signature is visible between 2000 m and 2500 m altitude. (b) Mean and standard deviation of the ML  $\Psi_{dp}$  profiles.

Figure 3.8:  $\Psi_{dp}$  in a clear ML signature scenario

ML signature. The mean is showing a very slight increase in differential phase, but primarily the standard

deviation is showing an increase in the ML. The differential phase is increasing toward 0.5 degrees and after the ML it drops to 0.3 degrees. Which points out the presence of the differential backscattering phase in the ML because the propagation phase difference between horizontal and vertical polarisations cannot decrease. Since the retrieval of a ML signature of  $\Psi_{dp}$  was already difficult to achieve, the  $\Psi_{dp}$  is deemed not useful in detecting the ML.

### Vertical velocity

The same procedure can be carried out on the vertical velocity. Averaging with a moving window over 20 range cells was necessary to create a dataset from which the maximum first derivative could be obtained accurately and reliably. An example of a first derivative of the vertical velocity is given in Figure 3.9. This shows that the maximum of the first derivative is suitable as a reference height of the vertical velocity. The velocity is derived

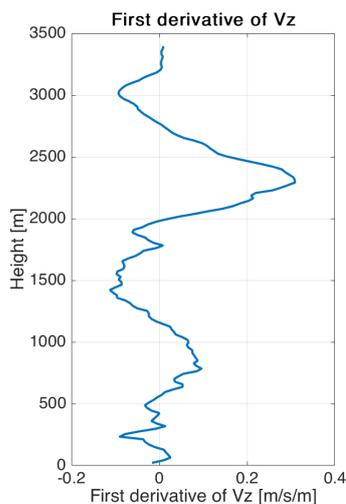
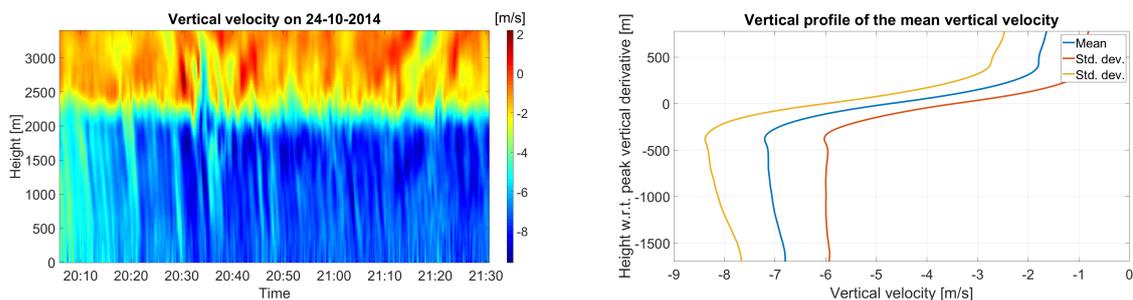


Figure 3.9: Example of a first derivative profile of the vertical velocity (profile number 352 of the study case on the 24th of October).

from the Doppler shift of falling hydrometeors. It is shown in Figure 3.10a that a transition is visible within the ML between slow moving hydrometeors above the ML and faster moving hydrometeors below the ML. The result is shown in Figure 3.10b. The transition phase is quite clear from this graph. When we take a look



(a) Image of the vertical velocity on the 24th of October 2014 during the AC-CEPT campaign. A transition is clearly visible in the ML between 2100 m and 2600 m altitude.

(b) Mean and standard deviation of the ML vertical velocity profiles.

Figure 3.10: Vertical velocity in a clear ML signature scenario

at the ML signature and how much vertical velocity changes in the ML, we can see that above the ML, the mean falling velocity is -1.7 m/s, the mean value below the ML is -7.1 m/s. This means an increase in velocity in the ML of 5.4 m/s. This could be used for ML detection, but vertical velocity is very dependent on vertical winds as well.

### Doppler spectral width

The procedure has been repeated on the Doppler spectral width. Averaging with a moving window over 20 range cells was necessary to even out statistical variations such that the minimum of the first derivative could be obtained accurately. An example of a first derivative of the Doppler spectral width is given in Figure 3.11. This shows that the minimum of the first derivative is suitable as a reference height of the Doppler spectral width. Also, the Doppler spectral width is showing a transition phase in the ML. This can be seen in Fig-

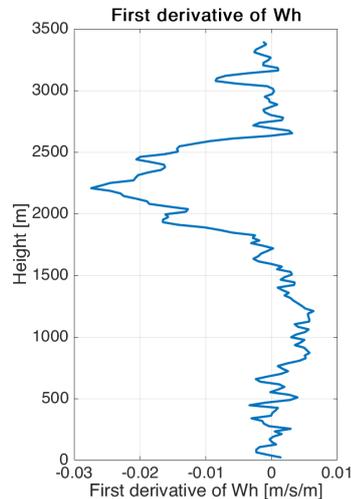
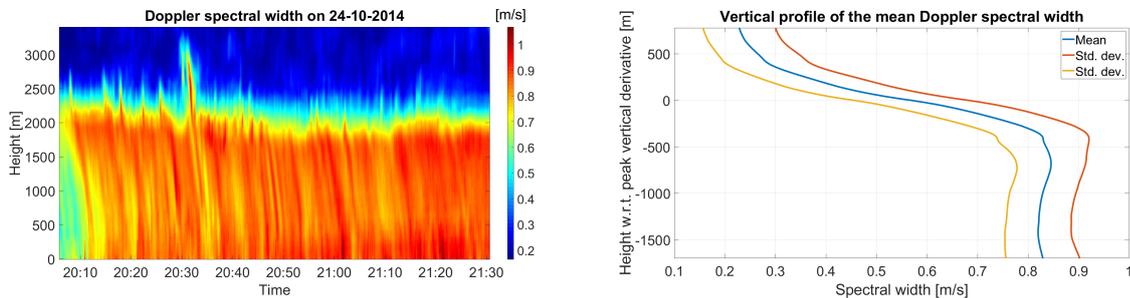


Figure 3.11: Example of a vertical profile of the first derivative of the Doppler spectral width (profile number 352 of the study case on the 24th of October).

ure 3.12a. The result can be seen in Figure 3.12b. When we take a look at the ML signature and how much the



(a) Image of the Doppler spectral width on the 24th of October 2014 during the ACCEPT campaign. A transition is clearly visible in the ML between 2000 m and 2600 m altitude.

(b) Mean and standard deviation of the ML Doppler spectral width profiles.

Figure 3.12: Doppler spectral width in a clear ML signature scenario

Doppler spectral width changes in the ML, we can see that above the ML, the mean Doppler spectral width is 0.23 m/s, the mean value below the ML is 0.85 m/s. Note that the vertical velocity and the Doppler spectral width are negatively correlated. The increase of Doppler spectral width in the ML could be used for ML detection.

### Doppler spectral skewness

The last radar observable on which the procedure is applied is the Doppler spectral skewness. In this case the data was averaged in vertical direction with a moving window of 10 range resolution cells. An example of a first derivative of the Doppler spectral skewness is given in Figure 3.13. This shows that the minimum of the first derivative is suitable as a reference height of the Doppler spectral skewness. The Doppler spectral skewness is a measure which tells how much the Doppler spectrum deviates from a symmetric distribution. As can be seen in Figure 3.14a again a transition layer is visible in the ML. When we take a look at the ML

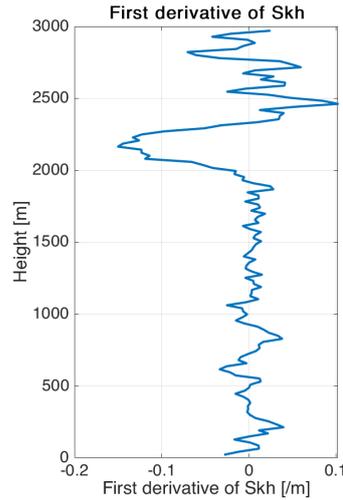
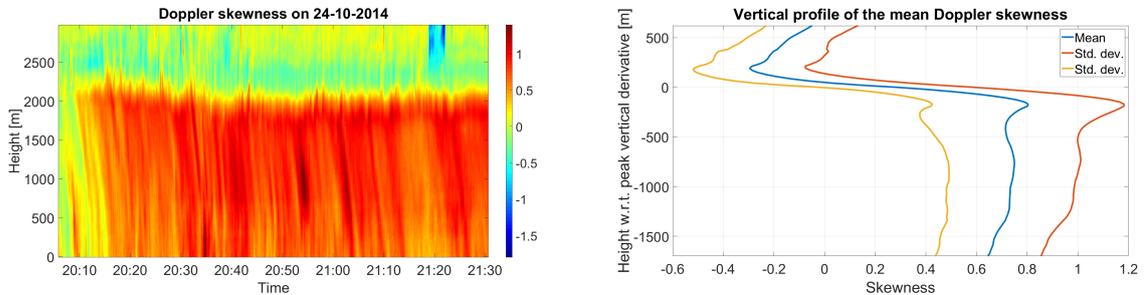


Figure 3.13: Example of a first derivative profile of the Doppler spectral skewness (profile number 352 of the study case on the 24th of October).



(a) Image of the Doppler spectral skewness on the 24th of October 2014 during the ACCEPT campaign. A transition is clearly visible in the ML between 2000 m and 2600 m altitude.

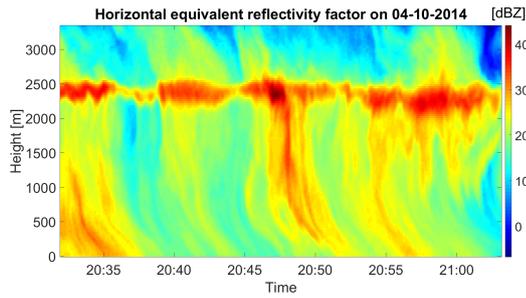
(b) Mean and standard deviation of the ML Doppler spectral skewness profiles.

Figure 3.14: Doppler spectral skewness in a clear ML signature scenario

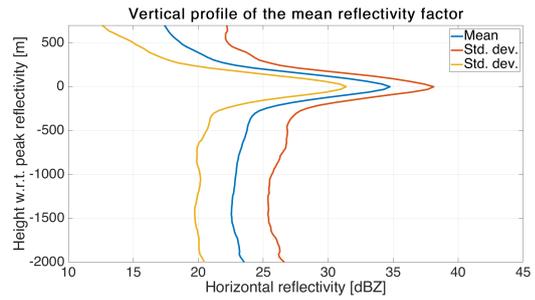
signature and how much the Doppler spectral skewness changes in the ML, we can see that above the ML, the mean Doppler spectral skewness is between -0.2 and 0, the mean value below the ML is 0.75. So, there is significant change occurring within the ML to the Doppler spectral skewness. Another thing which can be noticed is that the Doppler spectral skewness is showing a sign change because of the ML, this also could be of interest for ML detection.

### 3.2.2. Mean and standard deviation of ML signature at 90° elevation

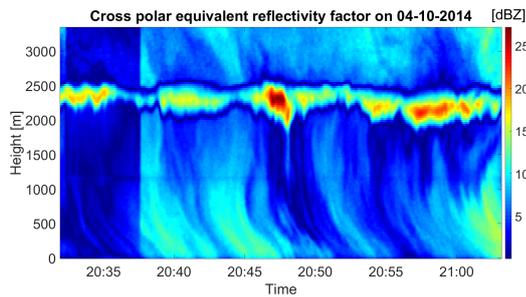
As mentioned in section 3.2 the same analysis is carried out on the calibration day measurements, when TARA was elevated to 90°(vertical). The horizontal equivalent reflectivity factor shown in Figure 3.15a and Figure 3.15b still shows a significant ML signature which can be used for detection of the ML. The cross polar equivalent reflectivity factor as shown in Figure 3.15c and Figure 3.15d shows a very usable ML signature at 90° elevation. The  $Z_{DR}$  shows, as expected, almost no ML signature at 90° elevation. This is shown in Figure 3.15e and Figure 3.15f. It shows a very slight increase, which is hardly detectable.  $L_{DR}$  still shows a solid signature in the ML as shown in Figure 3.15g and Figure 3.15h and is thus usable for ML detection at 90° elevation. The  $\rho_{hv}$  also shows a significant ML signature at 90° elevation. This is shown in Figure 3.15i and Figure 3.15j. Finally, the  $\Psi_{dp}$  shows hardly a ML signature as can be seen in Figure 3.15k and Figure 3.15l. The mean remains around zero, but the standard deviation increases in the ML. It is noticeable, but not significant enough for automatic ML detection.



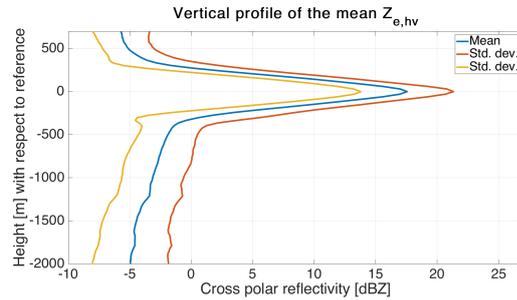
(a) Image of the horizontal equivalent reflectivity factor on the 4th of October 2014 during the ACCEPT campaign. A clear bright band signature is visible between 2300 m and 2700 m altitude.



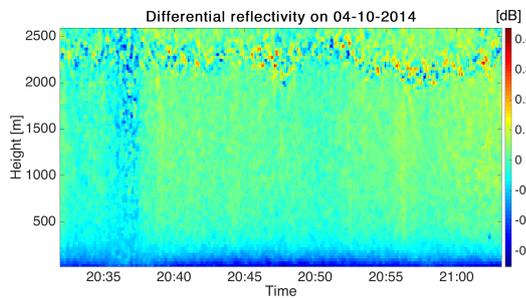
(b) Mean and standard deviation of the ML equivalent reflectivity factor profiles.



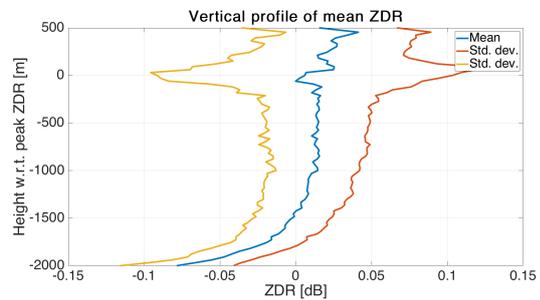
(c) Image of the cross polar equivalent reflectivity factor on the 4th of October 2014 during the ACCEPT campaign. A clear bright band signature is visible between 2300 m and 2700 m altitude.



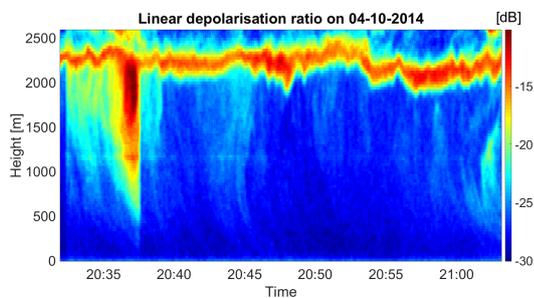
(d) Mean and standard deviation of the ML equivalent cross polar reflectivity factor profiles.



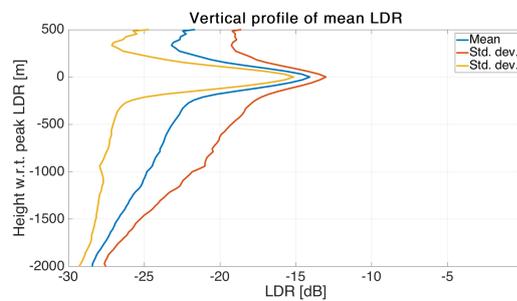
(e) Image of the  $Z_{DR}$  on the 4th of October 2014 during the ACCEPT campaign. No clear ML signature is visible.



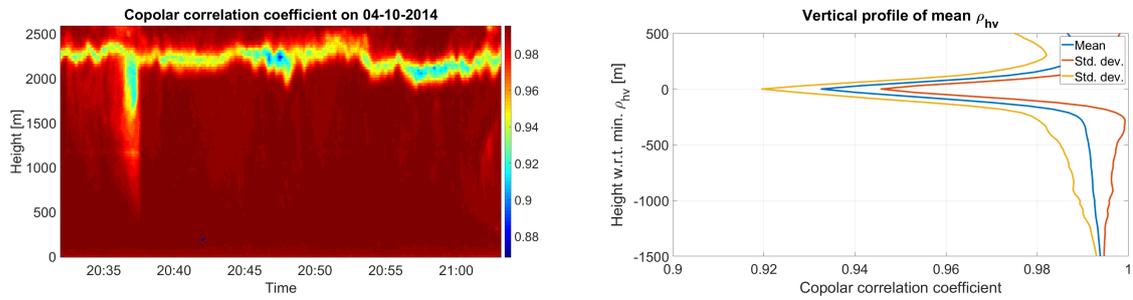
(f) Mean and standard deviation of the ML  $Z_{DR}$  profiles.



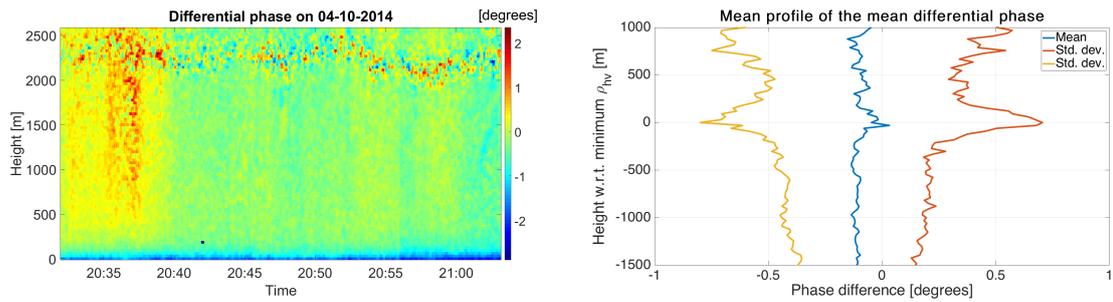
(g) Image of the  $L_{DR}$  on the 4th of October 2014 during the ACCEPT campaign. A clear signature is visible between 2000 m and 2500 m altitude.



(h) Mean and standard deviation of the ML  $L_{DR}$  profiles.



(i) Image of the  $\rho_{hv}$  on the 4th of October 2014 during the ACCEPT campaign. A clear signature is visible between 2000 m and 2500 m altitude. (j) Mean and standard deviation of the ML  $\rho_{hv}$  profiles.



(k) Image of the  $\Psi_{dp}$  on the 4th of October 2014 during the ACCEPT campaign. A signature is visible between 2000 m and 2500 m altitude. (l) Mean and standard deviation of the ML  $\Psi_{dp}$  profiles.

Figure 3.15: All available radar observables when TARA was pointing at 90°. Most observables show a significant ML characteristic. Only  $Z_{DR}$  and  $\Psi_{DP}$  are showing almost no ML signature.

### 3.3. Relative height of ML signatures

Since we have the profile alignment procedure using a defined reference height, it is possible to gain more understanding about the relative height of the different radar observables. From theory it is known that different radar observables react to different physical aspects within the ML. This implies (and is also recognized in literature) that the radar ML signatures lie on different altitudes. Using the knowledge gained from section 3.2, we know the altitude of the maxima of all the profiles. We also know the mean of all the profiles in the ML. The mean profiles can be plotted in one figure, where the peak of the signature corresponds to the mean height of the peak. This can be applied to all radar observables using the reference height based on Table 3.2. This results in Figure 3.16. In this figure, all observables are plotted against their reference peak value. Now it is easily seen that the different radar observables have their ML signatures on different altitudes. The list of observables and their respective absolute and relative altitudes are shown in Table 3.3. From this it can be noticed that there seem to be roughly four categories. Doppler spectral skewness first derivative and  $Z_{e,hh}$  occur on the upper part of the ML. Then around 100 m below the  $Z_{e,hv}$ , vertical velocity and spectral width occur. Again 100 m lower  $L_{DR}$ ,  $\rho_{hv}$  and  $\Psi_{dp}$  occur. Lastly  $Z_{DR}$  shows the lowest altitude. Note that  $\Psi_{dp}$  lies on the exact same height as the  $\rho_{hv}$ , this is due to the fact that the  $\rho_{hv}$  minima height is used to align the profiles of  $\Psi_{dp}$  as explained in section 3.2. As can be seen in the image as well as the table is that the ML spans more

Table 3.3: Absolute mean peak altitude and mean relative altitude

Observable	Peak [m]	$Z_{e,hh}$	$Z_{e,hv}$	$Z_{DR}$	$L_{DR}$	$\rho_{hv}$	$\Psi_{dp}$	Vz	Wh	Skh
$Z_{e,hh}$	2343	0	106	247	184	194	194	116	91	-21
$Z_{e,hv}$	2237	-106	0	141	78	88	88	10	-15	-127
$Z_{DR}$	2096	-247	-141	0	-63	-59	-59	-131	-156	-268
$L_{DR}$	2159	-184	-78	63	0	10	10	-68	-93	-205
$\rho_{hv}$	2149	-194	-88	59	-10	0	0	-78	-103	-215
$\Psi_{dp}$	2149	-194	-88	59	-10	0	0	-78	-103	-215
Vertical velocity	2227	-116	-10	131	68	78	78	0	-25	-137
Doppler spectral width	2252	-91	15	156	93	103	103	25	0	-112
Doppler spectral skewness	2364	21	127	268	205	215	215	137	112	0

than the thickness of a ML signature of a single radar observable. This should be taken into consideration for a ML definition.

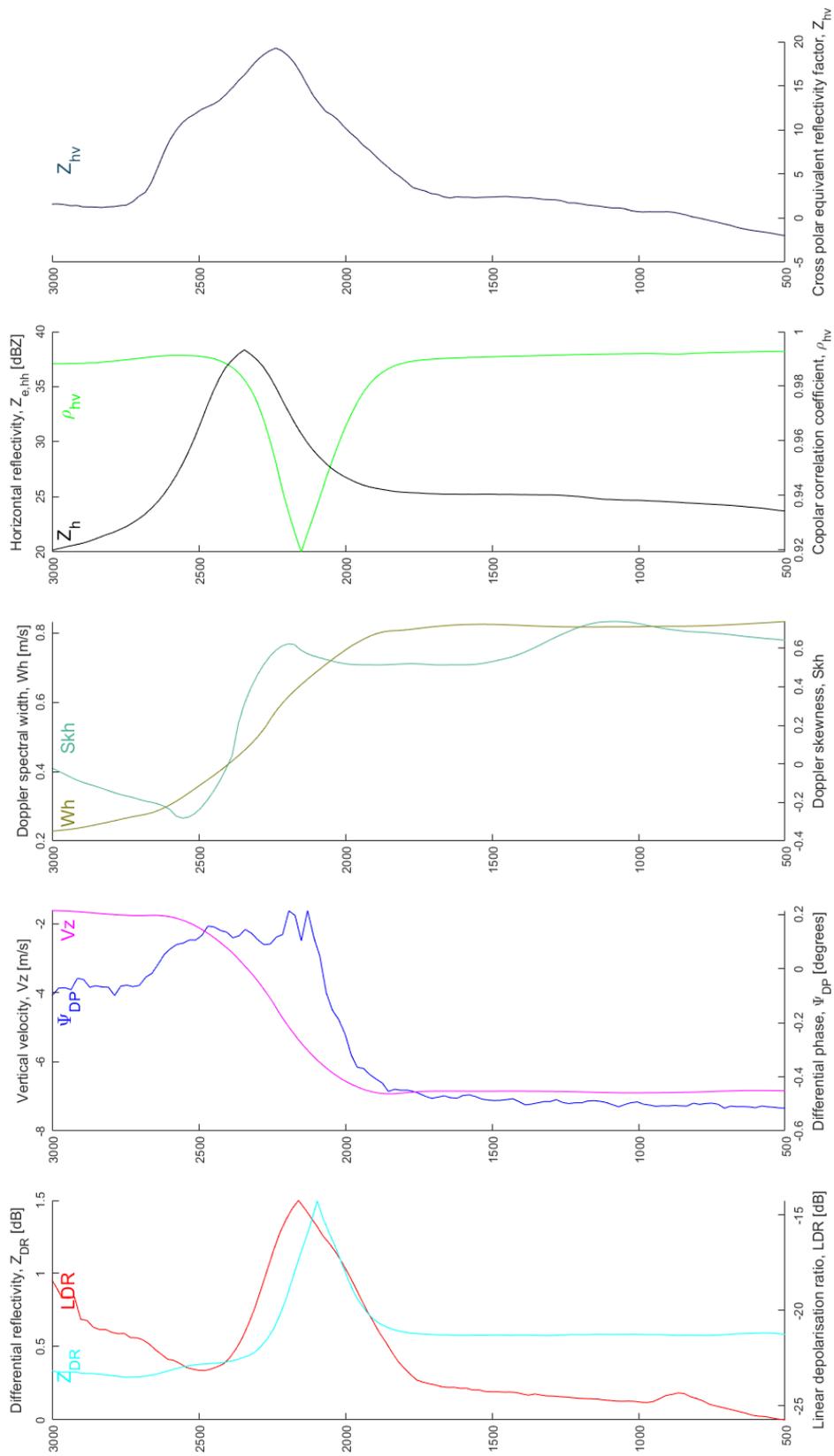
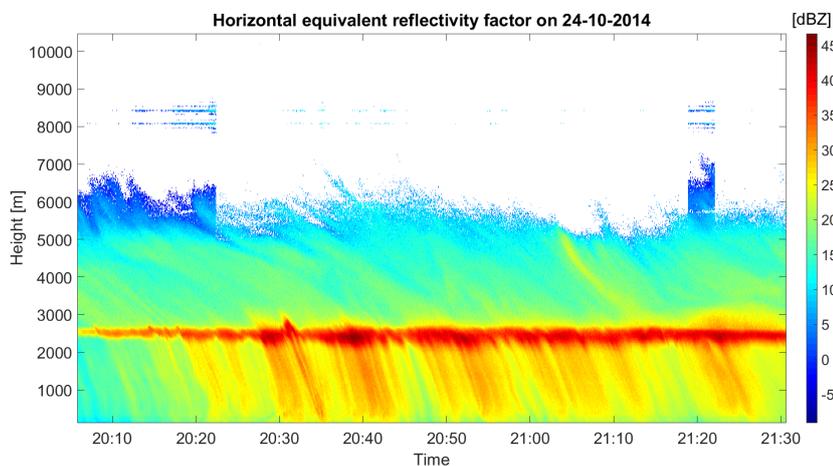


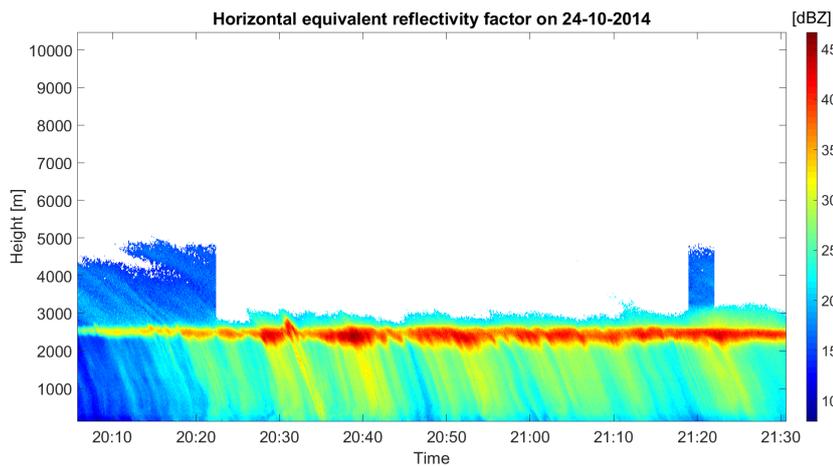
Figure 3.16: A comparison of the relative altitudes with respect to their peak values.

### 3.4. Removal of upper atmosphere unwanted echo

As can be seen in Figure 3.2a in the upper part of the atmosphere (above 7000 m) some echo is showing. This was not a problem for the previous steps, because only data from 0 m to 3500 m was considered. For an automated algorithm to work, it must be able to process all the data. The echo in the upper part of the atmosphere is unwanted, since it interferes with the data analysis later on. Especially polarimetric radar observables are noisy and can be biased at the higher altitudes due to a low signal to noise ratio. A way to overcome this is to set a SNR threshold on the entire dataset. The ML will probably not be affected by the imposed SNR threshold, since the ML has a high SNR. At first instance a SNR threshold of 10 dB (on the  $Z_{e,hh}$ ) is introduced. The SNR threshold level will be evaluated during the execution of the SNR threshold to the dataset later in the thesis. The 10 dB SNR significantly improves the amount of usable data for ML detection and characterisation. It removes most of the data above the ML as can be seen in Figure 3.17b. The SNR threshold can be applied because TARA supplies a noise estimate per range resolution cell (in dBZ). The noise matrix coming from TARA is subtracted from the  $Z_{e,hh}$ . If the difference is lower than the threshold, data in that specific resolution cell is set to NaN. The same NaN map is used on all other radar observables. The result can be seen in Figure 3.17. Figure 3.17a shows the original dataset, without SNR threshold and Figure 3.17b shows the result after applying the SNR threshold. The instantaneous increases and decreases of signal which are visible in Figure 3.17b around 20:20 and 21:20 is due to a change in transmit power of TARA. TARA can only increase and decrease transmit power in increments of 10 dB.



(a) Complete dataset, no SNR threshold applied.



(b) The same dataset, but now a SNR threshold of minimal 10dB is applied.

Figure 3.17: The result of imposing a SNR threshold of 10 dB on the data.

After application of the SNR threshold, still some outliers remain in the data. The second step in improving the data is therefore to remove all single pixel data cells. As can be seen in Figure 3.18a, some single pixels

of data remain with very high values even if they are not surrounded by other data. The process to remove these pixels is simple and based on an image processing method. First a logical map of the data array will be created. A logical one is where a data is present in the data array, a logical 0 is where a NaN is present. The logical map is in essential a black and white image and thus the image can be eroded or dilated. By eroding the logical map with a square structuring element, the data parts in the logical map are decreased in size and as a result eroded.

The process of image erosion is graphically displayed in Figure 3.19. When the structuring element in the middle is applied to every pixel of the left black and white image, the result is shown on the right black and white image. Essentially the white part (cells containing a 1) of the image is decreased in size. In this case this means that the data area is decreased in size by one pixel in every direction, eliminating all the single pixels surrounded by NaN values (as is shown in the lower right corner of the example). The original size of the image can be regained using image dilation, the inverse of image erosion. The result of the procedure is seen in Figure 3.18b. The single data cells with high values are all removed.

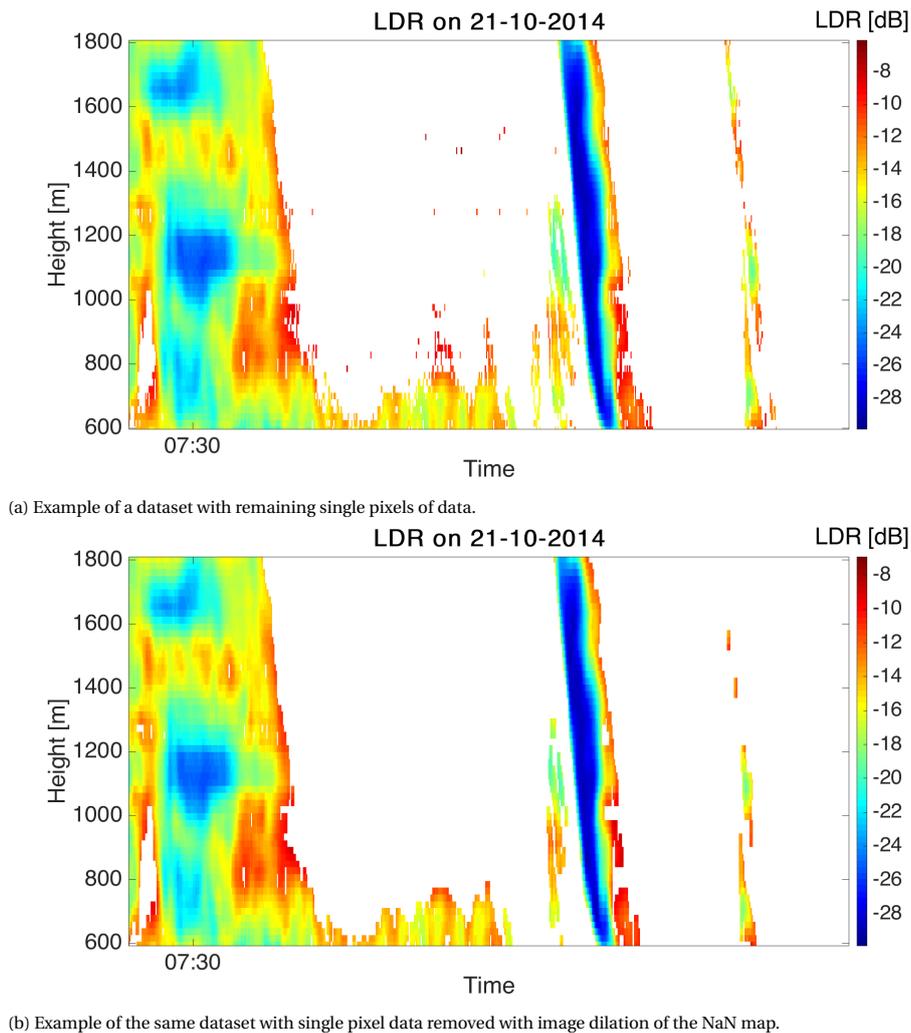


Figure 3.18: Example of removing single data cells using image dilation of the NaN map

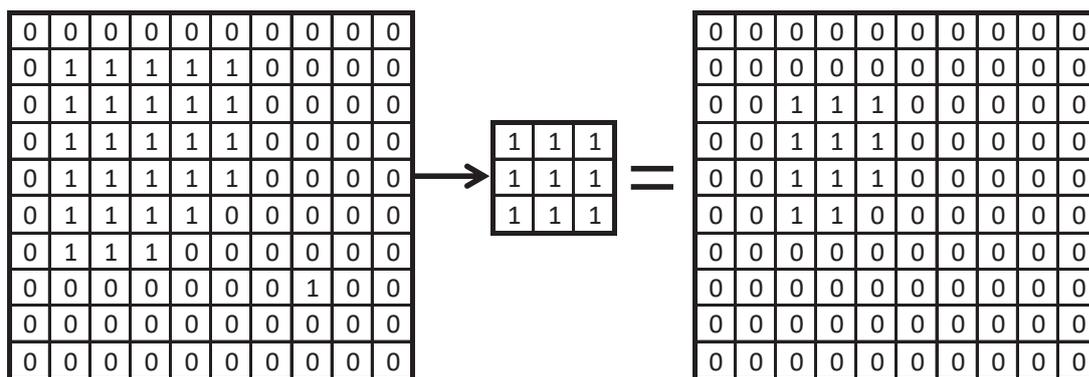


Figure 3.19: Process of image erosion of a black and white image using a structuring element (se). 1 indicates white (data), 0 indicates black (no data).

### 3.5. Comparison with radiosonde measurement

The purpose is to find (if there is one) a relation between the 0° isotherm altitude and the radar observables. It provides useful information for the definition of the ML. During the ACCEPT campaign some radiosondes were launched to record temperature profiles. The first step is to select the time instants when they were launched. Secondly it should be checked if during those launches any stratiform precipitation was present. If there was stratiform precipitation and there was a significant ML signature, then that data should be compared to the 0° altitude. To perform this operation, a window of 20 minutes surrounding the time instant when the 0° was measured should be taken from the dataset of radar observables. That dataset is then averaged with a moving average of 5 profiles and 5 range resolution cells to average out most of the variances. The mean of the profiles of the 20-minute window can then be used to be compared to the radiosonde measured 0° isotherm.

The radiosondes record different data types:

- Time (in seconds since launch)
- Height
- Temperature
- Dew Point
- Liquid saturation
- Ice saturation

Even more parameters are measured, but the above mentioned are most important. The time, height and temperature are very relevant to this thesis, since they give the measured 0°C isotherm at a certain height and time. Time can be used to correlate the radar observed data and the radiosonde data. To exactly correlate the radar observed data and radiosonde data, first the radiosonde data was checked when the 0°C isotherm was observed. That elapsed time corresponds to a height and to a time since launch. The time elapsed added to the launch time can then be correlated to the radar observed data. Using this method imposes a risk, since the radiosonde is traversing during ascent as well. That means that radiosonde measurements are probably not at the same position where the radar measures atmospheric data. The assumption is made that the 0°C isotherm does not change significantly in the 20-minute window surrounding the radiosonde measurement. To make sure that is the case, a plot of the horizontal equivalent reflectivity factor is produced and shown in Figure 3.20. In that figure the measurement of the 0°C isotherm by the radiosonde is represented as the black square in the middle. Finally, a plot of the data from both radar and radiosonde measurements is shown in Figure 3.21. Only six instances were found where a radiosonde launch had taken place when also radar data of some stratiform precipitation was available. This makes a thorough comparison of radar observations with radiosonde data difficult and restrictive. From Figure 3.21 we can conclude that the 0°C isotherm lies just in

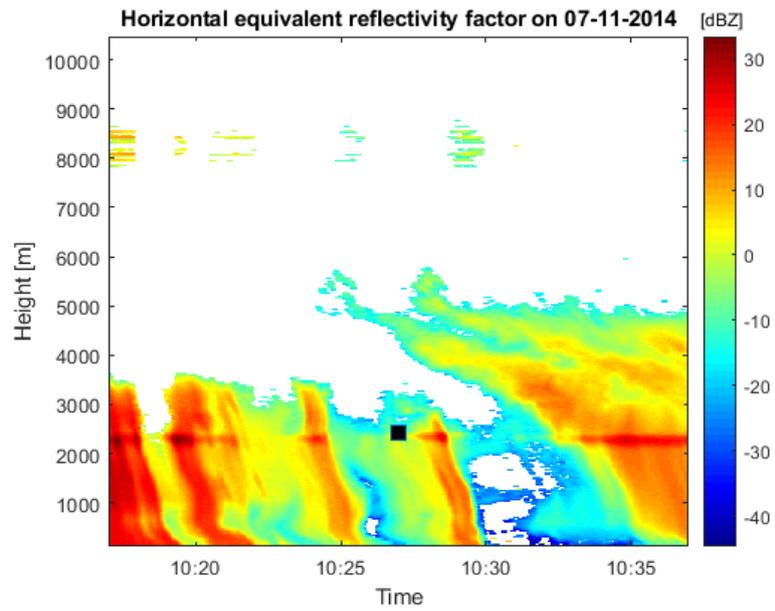


Figure 3.20: Horizontal equivalent reflectivity factor of the used window for calculating the average profile. The black square represents the radiosonde measurement of the 0°C isotherm.

the transition between the linear and non-linear response of most radar observables. This corresponds to White et al. (2002) and Stewart et al. (1984).

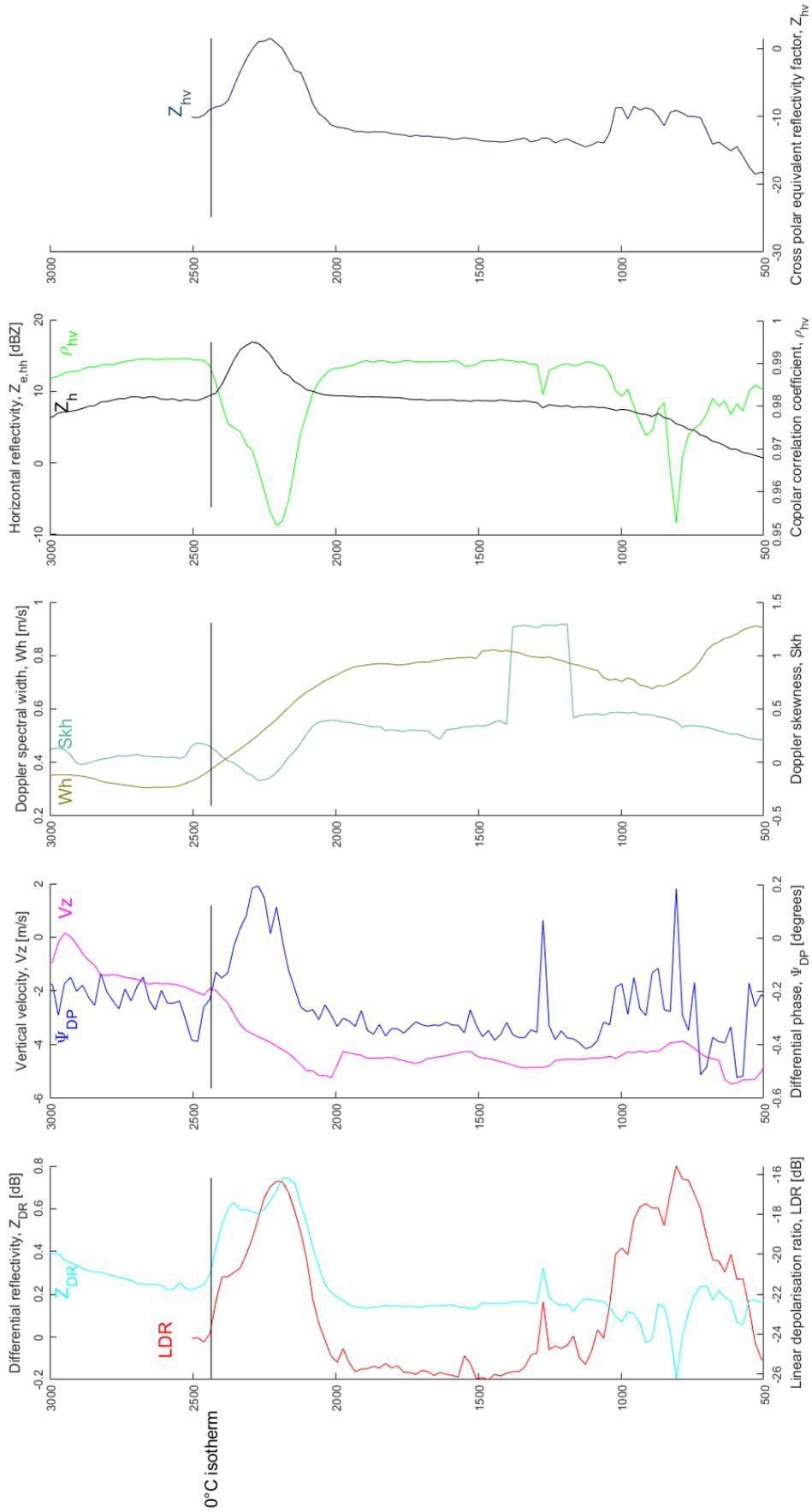


Figure 3.21: A comparison of the radiosonde 0 °C isotherm measurement and the radar observables.

### 3.6. Conclusion

The conclusion can be drawn that from the available radar observables only the  $\Psi_{dp}$  and  $K_{DP}$  are not suited for ML detection or analysis. It was very difficult to even see a signature in the ML. Other observables have a quite well distinguishable ML characteristic which can be used for detection. Doppler spectral skewness, Doppler spectral width and vertical velocity are radar observables which are subject to more variability, considering the extra necessary averaging. These observables are therefore more difficult to use for ML detection. In the case of 90° elevation, it is shown that the  $Z_{DR}$  is not usable for ML detection as well as  $\Psi_{DP}$ . Based on argumentation in section section 3.2, we could conclude that the Doppler related radar observables were just as usable as when a radar is pointing at 45° elevation. The list of radar observables being used for ML detection within this thesis will therefore be:

- $Z_{e,hh}$ ;
- $Z_{e,hv}$ ;
- $Z_{DR}$ ;
- $L_{DR}$ ;
- $\rho_{hv}$ .

The difference between their maxima and their values well above and below the ML is significant which might be used for detection of the ML. Another characteristic of the ML is the height difference between the radar observed maxima or minima. The other (Doppler) observables might serve as a backup, that remains to be determined during algorithm development.

From the full profiles we can conclude that the upper part of the measured atmosphere might pose a problem for automatic ML detection, because the observed values in the upper part of the atmosphere exhibit large variances. This is due to a low SNR where the measurements are not reliable any more for especially the polarimetric observables. A big improvement is made by imposing a threshold on the SNR of the equivalent horizontal reflectivity factor. Most of the data of the upper part of the atmosphere is removed by using a threshold of 10 dB on the data.

Finally the comparison of radiosonde data with radar observations gave some insight in the relative height of the radar observables with respect to the 0°C isotherm. The 0°C isotherm looks to be just in the transition of the profile of the radar observables, but the amount of usable data was very limited. More data is required to support this conclusion.

# 4

## Methodology

**This Chapter will describe the proposed methods which are going to be developed and investigated during the Master project. The goal is to create a program which is able to automatically detect and characterise the melting layer within a dataset. First a definition of the ML is proposed, then a reference method from literature is proposed and discussed. Subsequently three basic image processing techniques are proposed and finally a ML detection and characterisation method is proposed. At the end of the Chapter, the method on the performance evaluation is discussed.**

### 4.1. Melting layer boundary definition

After understanding the ML and its characteristics in the previous Chapter a definition of the melting layer boundaries should be created. This definition becomes the standard for testing the detection algorithms.

The definition of the melting layer on the one hand should have a connection to the physical process within the ML, on the other hand the purpose of this thesis is to select the radar data which belongs to the melting layer. The first statement is supported with the results gained from section 3.5 where the radiosonde temperature profiles are compared with the radar measurements. From Figure 3.21 it is noticed that the 0°C isotherm lies just in the transition between non-ML and ML signature of the radar observables  $L_{DR}$ ,  $Z_{DR}$ ,  $\rho_{hv}$ ,  $Z_{e,hh}$  and  $Z_{e,hv}$ . By "transition" the transition phase between the linear and non-linear part of the radar observables is meant. Therefore, the middle of the transition area is suggested as the start of the melting layer. However, it is questionable if all the radar observables show their transition phase on the same altitude as is suggested by Figure 3.21. If Figure 4.9a and Figure 3.16 are recalled a different conclusion may be drawn. The second statement indicates that the melting layer detection algorithm should be able to select data which encompasses the entire ML signature of all radar observables. This statement is supported by the relative height analysis of section 3.3. The upper boundary of the ML is chosen to be the transition between the non-linear and the linear part of the radar observable which exhibits the highest ML signature. The lower boundary of the ML can then be defined to be the transition between the linear and non-linear part of the radar observable which exhibits the lowest ML signature. Such a definition ensures that it covers all the data belonging to the ML, is independent of available radar observables and the upper boundary is close to the true 0°C isotherm.

The method to determine the boundaries is based on the coordinate transformation method developed in Rico-Ramirez and Cluckie (2007) and described in section 2.2.1.1. Their method generates a boundary which lies on the same position as is meant by the middle of the transition. They only applied their method to the reflectivity profile for detecting the bright band, but the use of the method is not limited to reflectivity profiles only. When their method is applied to the melting layer definition as described above, a coordinate transformation is applied to retrieve the upper boundary of the ML using the reflectivity or the next observable to come. Another coordinate transformation is applied on the lower part of the  $Z_{DR}$  curve (or the next observable if  $Z_{DR}$  is unavailable) to retrieve the lower boundary of the ML. Their method of coordinate transformation can thus be applied to all other radar observables. It can even be applied to the Doppler based observables, although they do not exhibit a sharp transition and as a consequence the estimation of the boundary is less accurate. An image of the ML definition is shown in Figure 4.1. In the image, the x-axis

has been removed for clarity. The application of the coordinate transformation method to acquire the top and bottom boundary of the ML is shown in the image. The top of the melting layer is defined as the mid-

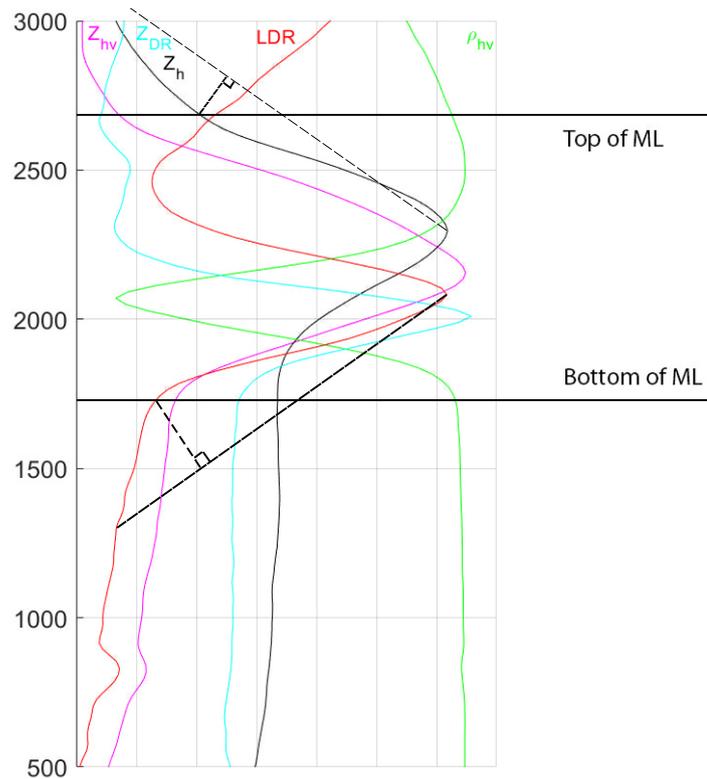


Figure 4.1: The definition of the ML. This graph shows the average ML signatures of five different radar observables. The ML top is defined to be the transition given by the coordinate transformation method of Rico-Ramirez and Cluckie on the  $Z_{e,hh}$ . The bottom of the ML is defined by the same method, but now applied to the  $Z_{DR}$ .

dle of the upper transition phase of  $Z_{e,hh}$ . The bottom of the melting layer is defined as the middle of lower transition phase of the  $Z_{DR}$ . This definition only holds if these radar observables are available. Whenever one of the observables is not available, the definition uses the radar observable which is closest to the one which was not available. If for instance  $Z_{DR}$  is not available,  $\rho_{hv}$  or  $L_{DR}$  will be used for determining the bottom of the ML. Now that a definition is available, the ML detection and characterisation methods can be developed starting with the reference method.

## 4.2. Reference ML characterisation method

To have a baseline to compare the developed algorithms against, a reference existing ML detection method is required. The detection method of Giangrande et al. (2008) is selected in section 2.4 as the reference ML detection algorithm since it is being used in multiple publications and it is easy to implement. Their method consists of using multiple thresholds on different radar observables. The method summarised:

1. Check for  $\rho_{hv}$  between 0.90 and 0.97 below an altitude of 6 km. Mark all the cells which comply to this threshold;
2. For every marked cell, check if a maximum of  $Z_{e,hh}$  and  $Z_{DR}$  occurs within 500 m above the marked cells;
3. The maximum of  $Z_{e,hh}$  falls between 30 and 47 dBZ and the maximum of  $Z_{DR}$  falls between 0.8 to 2.5 dB.

According to Giangrande et al. (2008) pre-processing is performed on the data and consists of averaging  $Z_{e,hh}$  with a window of 3 cells,  $Z_{DR}$  and  $\rho_{hv}$  with 5 cells. If all of the above requirements are met, a cell is considered a ML cell.

When the described procedure is applied to the study case, the output of the method can be shown as an overlay on the image of the  $\rho_{hv}$ . This is shown in Figure 4.2. From Figure 4.2 it is clear that the method

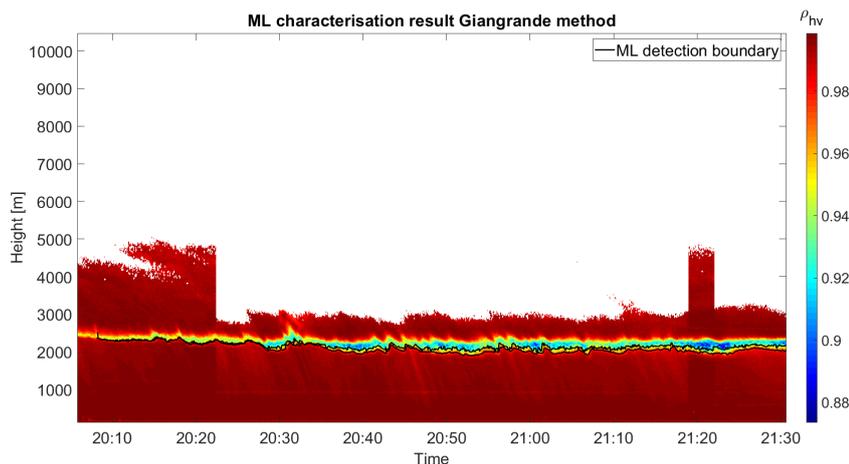


Figure 4.2: The result of the Giangrande et al. method shown as overlay to the copolar correlation coefficient.

succeeds in detecting the ML, but fails to fully capture the boundaries of the ML. This is caused by the second step being too restrictive. Only cells where the maximum  $Z_{e,hh}$  and  $Z_{DR}$  occurs above the marked cells could become ML cells. This means that only  $\rho_{hv}$  cells between 0.90 and 0.97 below the mentioned maxima can become ML cells. Also, the threshold of the  $\rho_{hv}$  is too restrictive on TARA data. A few adjustments were created to overcome most of the problems: lower the threshold for  $\rho_{hv}$ , increase the threshold for  $Z_{e,hh}$  and not only search above marked cells for the maxima, but also below marked cells. Usually the  $Z_{DR}$  peak is positioned below most of the  $\rho_{hv}$  marked cells and it is thus limiting the operation of the method.

- $\rho_{hv}$  between 0.85 and 0.97;
- $Z_{e,hh}$  between 30 and 49 dBZ;
- Search for maximum  $Z_{e,hh}$  and  $Z_{DR}$  500 m above and 200 m below marked cells.

When the suggested improvements from section 4.2 are implemented, the method improved in characterising the ML. This is shown in Figure 4.3. Now at least the ML portion which is related to the decrease of the

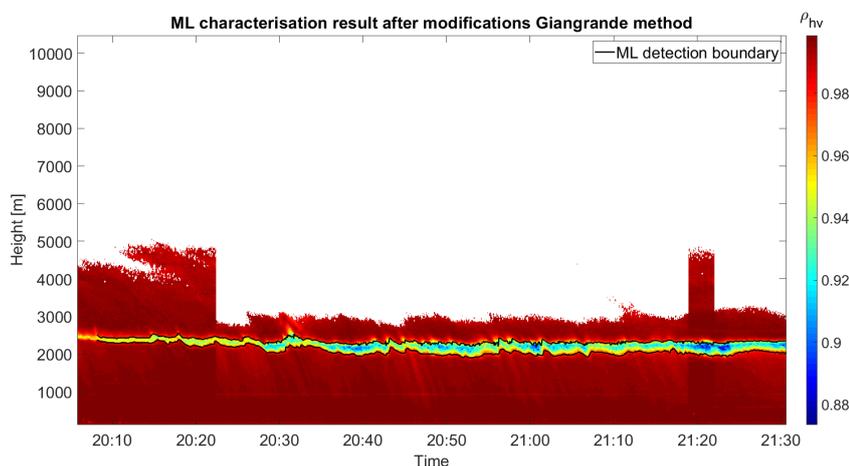


Figure 4.3: The result of the adjusted Giangrande et al. method shown as overlay to the copolar correlation coefficient.

$\rho_{hv}$  is fully captured. Still it does not detect the entire ML since on the left side of the figure (until 20:08) one can notice an undetected part, while it is clear that there is a ML. That part is not detected because of the maximum  $Z_{e,hh}$  being lower than the threshold of 30 dBZ. When the result is shown as an overlay over  $Z_{e,hh}$  in Figure 4.4 we see a result which was expected. Due to the height differences of the radar observables, not the entire ML signature of all radar observables is captured. In Figure 4.4 it can be seen that the detected ML thickness is not covering the upper part of the ML. While this already improved method is not perfect, it

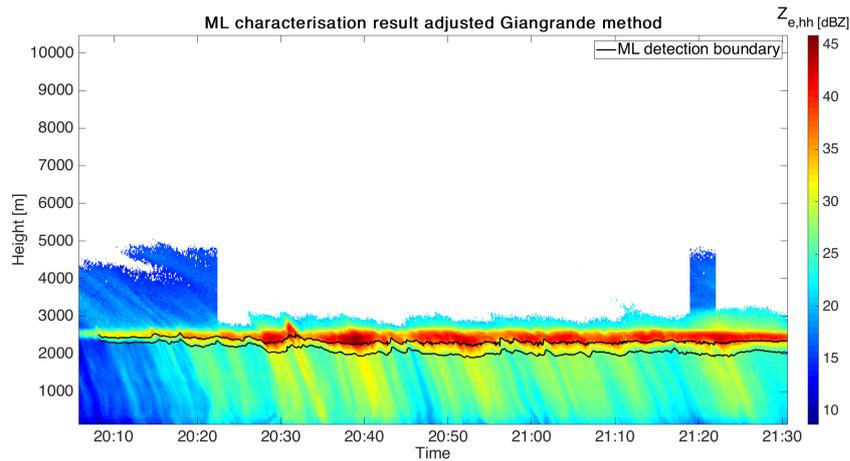


Figure 4.4: The result of the adjusted Giangrande et al. method shown as overlay to the horizontal equivalent reflectivity factor.

becomes the reference method. The reference method still has a few downsides:

- The thresholds are now suitable for our set up, but are not generally usable for others;
- The thresholds might still work limiting in detecting the ML;
- The method fails to capture the entire thickness of the ML;
- The method is dependent on the availability of radar observables.

The proposed reference method for this thesis becomes:

1. Average  $Z_{e,hh}$ ,  $Z_{DR}$  and  $\rho_{hv}$  data only in time with a window of five vertical profiles;
2. Check for  $\rho_{hv}$  between 0.85 and 0.97 below an altitude of 6 km. Mark all the cells which comply to this threshold;
3. For every marked cell, check if a maximum of  $Z_{e,hh}$  and  $Z_{DR}$  occurs within 500 m above and 200 m below the marked cells;
4. The maximum of  $Z_{e,hh}$  falls between 30 and 49 dBZ and the maximum of  $Z_{DR}$  falls between 0.8 to 2.5 dB.

### 4.3. Image processing ML characterisation methods

The second approach for ML detection is purely based on the images which are produced using the data. There are numerous approaches available, while this thesis will discuss three:

- Use the horizontal feature of the ML;
  - Hough transform for detecting horizontal features;
  - Radon transform for detecting horizontal features;
- Connected pixels (areas).

#### 4.3.1. Hough transform

The Hough transform can be used to detect lines in images. Since the ML could be considered as a noisy line, the Hough transform might be able to at least detect if there is a ML or not. A next step could be to detect the upper and lower boundary of the ML, since they both can be considered as noisy lines. The Hough transform for detecting lines in images was developed by Duda and Hart (1972). The basic idea is that every point in an image can be represented by a combination of angle ( $\theta$ ) and distance ( $r$ ) from a datum which is defined as the center of the image. The principle is shown in Figure 4.5. In Figure 4.5 it is shown that every pixel can be represented by an angle and by a distance. When points are lying on the same line, they have the same combination of angle and distance. Every point in an image is plotted in the parameter space where angle is on the x-axis and distance is on the y-axis. Every point in an image can be associated with an infinite amount of combinations of  $\theta$  and  $r$ .  $\theta$  and  $r$  are related by  $r = x \cos \theta + y \sin \theta$ . A matrix of  $\theta$  ranging from 0-360 (or

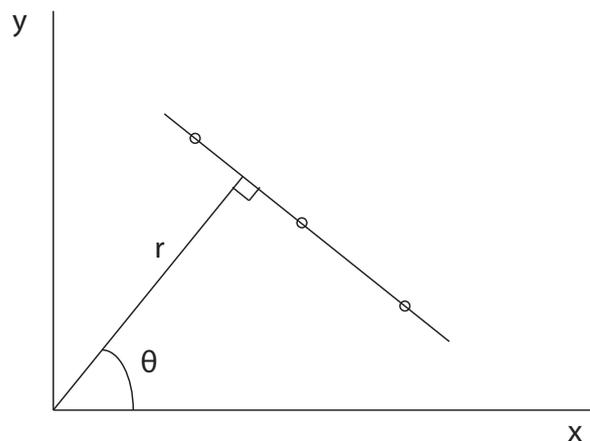


Figure 4.5: Principle of the Hough transform. Every point in the image can be represented by a combination of angle  $\theta$  and distance  $r$ .

$0-2\pi$ ) is generated with accordingly the calculated distances  $r$ . When this is plotted in the parameter space it looks like Figure 4.6. In this figure you can notice that three lines are drawn, which represent the three points from Figure 4.5. Each point is represented by a  $\theta$  and  $r$  relation, thus three lines are visible. Where the three lines coincide (magenta cross in Figure 4.6) a combination of  $\theta$  and  $r$  is found which is true for all three points from Figure 4.5. This gives the  $\theta$  and  $r$  values for the line passing through all three points. When this

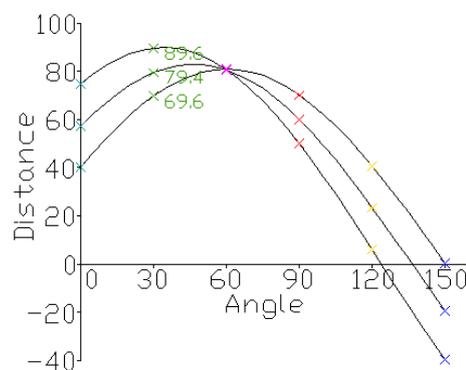


Figure 4.6: Principle of the Hough transform. The three lines are representing the three point of Figure 4.5. Where the lines intersect (at magenta cross) all the three points have one angle distance pair and as a result a line with that angle and distance pair can be drawn through the three points (image from [https://en.wikipedia.org/wiki/Hough\\_transform](https://en.wikipedia.org/wiki/Hough_transform)).

technique is combined with an edge detector, it could provide valuable results. The steps to be taken are:

- Convert image to grey scale;
- Apply edge detector (Sobel, tuned for horizontal edge detection);
- Transform edge detector result to Hough parameter space;
- Detect peaks in Hough parameter space;
- Draw results on original image.

### 4.3.2. Radon transform

The Radon transform is a similar method as the Hough transform, but this method involves integrating over parallel beams over a specified range of angles ( $\theta$ , as shown in Figure 4.7). This process is repeated for every beam 1 pixel away from the previous beam. It basically comes down to make projections of the image on the x-axis. This can be plotted again in a Radon parameter space where the angle is on the x-axis and the deviation from the middle of image is on the y-axis. Without going into too much detail, the procedure used for detecting lines with the Radon transform is the same as with the Hough transform. The Radon parameter

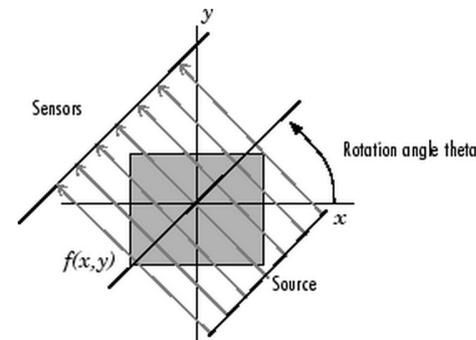


Figure 4.7: Principle of the Radon transform. The image is integrated over parallel beams. The beams are separated by 1 pixel in  $y$  direction. This process is repeated over a multitude of angles  $\theta$  (image from MATLAB 2018a documentation).

space is then analysed using the same method as with the Hough transform and the two parameters ( $\theta$  and vertical distance from center) are extracted from the Radon parameter space.

### 4.3.3. Connected pixels (area size)

Another quite simple method which might prove valuable in detecting the ML could be to just select the pixels which show a similar feature and are close to each other. This is the core principle of this method. The steps to be taken are:

- Create grey scale image from the original image;
- Create black and white picture from grey scale image using a certain threshold;
- Group connected pixels using 'bwconncomp' function in MATLAB;
- Order groups of pixels in pixel count (area);
- Show largest groups of pixels.

This method probably only works correct with the  $\rho_{hv}$ , because it is uniform in almost all areas except in the ML. Thus, the ML is probably the largest area of connected pixels. This is not true for other radar observables. The method produces always an area of largest connected pixels and as a result it also generates a result when there is no ML.

Some basic image processing techniques are explained and next the final ML detection and characterisation method will be introduced.

## 4.4. Machine learning ML characterisation method

In this section a different approach to detect and characterise the ML is taken. The use of machine learning techniques is proposed. The basic principle is to let a learned algorithm decide which data is ML data and which data is not ML data. This learning is done using an annotated or labelled dataset of training data. Once the algorithm has learned from the training data it can be used to classify new data. The goal is not to fully explain and understand the operation of the different classifiers, but to use them in our advantage. MATLAB offers a unique toolbox where a large number of classifiers are readily available, can be trained simultaneously and a performance estimate is given. The most promising methods can then be analysed in more detail to select the best method. One must be careful with the performance estimate, because MATLAB measures the performance of the learned classifier using the same training dataset. The estimate gives an indication of how good the classifiers are at reproducing the training dataset.

The general process of any supervised learning machine learning tool:

1. Select features which describe the phenomena to be found;
2. Create a training dataset with labels which represents the entire dataset;
3. Fit a model to the training data;
4. Evaluate the performance of the model.

The first step in the process is to create features. A feature is an individual measurable property or characteristic of a phenomenon being observed. Feature engineering is one of the most important steps in machine learning.

Then the model needs to be trained using a representative labelled training dataset. This training dataset should contain a very clear situation of a ML. The training data should be separated from the rest of the data, because a model will have higher performance scores if (parts of) the training dataset is used the performance evaluation.

Using the labelled training dataset the model can be trained. From the training dataset, the features are calculated, put in a feature vector space and given to the machine learner.

Once the model is trained, the performance should be evaluated. To evaluate the performance of the model a labelled test dataset is required. The test dataset should be a large labelled dataset taken from the entire ACCEPT campaign comprising all different scenarios (No rain, convective rain, stratiform rain). The performance evaluation is explained further in section 4.5.

Both of these steps are explained next, starting with the ML detector.

In this thesis a two-step approach will be taken to use machine learning for ML detection. The first step, detection, will detect if a ML exists in the vertical profile under consideration. The second step, attribution, only starts if the detector detected a ML.

#### 4.4.1. ML detector

The first step in the process is the ML detector. This detector needs to determine if a vertical profile contains a ML signature or not.

##### Detector features

In this case the features need to be different for a profile with a ML signature and a profile without a ML signature. Additionally, it is preferable that the features do not change significantly due to radar orientation. The machine learner can then learn the optimal boundary between the features with ML signature and without. An initial feature set for the detector is proposed:

- Maximum (or minimum in case of  $\rho_{hv}$ ) - mean of entire profile;
- Maximum (or minimum in case of  $\rho_{hv}$ ) - median of entire profile;
- Altitude difference between maxima (or minimum in case of  $\rho_{hv}$ ) of all radar observables;
- Difference between maxima (or minimum in case of  $\rho_{hv}$ ) and the value of the same radar observable 800m below the maxima;
- Variance of entire profile.

This list of features is calculated for the following radar observables:  $Z_{e,hh}$ ,  $Z_{e,hv}$ ,  $Z_{DR}$ , and  $\rho_{hv}$ .  $L_{DR}$  has been left out initially, because  $L_{DR}$  tends to rapidly increase above the ML as indicated in Figure 3.2.1. The total feature set consists of 22 features. The features which do use  $Z_{DR}$  will change significantly due to changing radar orientation. The usability of this initial set of features is later analysed.

##### Detector training

Training of the detector requires a training dataset which is independent of the rest of the data, but statistically with the same distribution. The entire ACCEPT campaign data is being split up into two separate datasets, the first being the training dataset and the second contains the rest of the ACCEPT campaign data. This training dataset should approach a 50/50 data ratio containing both classes, so the classifier doesn't have

the tendency to over predict one of the classes. The training dataset is created from different parts of the AC-CEPT campaign as follows:

Table 4.1: Training dataset composition for detector

Part	Date	Time start	Time end	Contains ML
1	12-10-2014	15:25	17:18	No
2	13-10-2014	00:04	01:02	No
3	14-10-2014	08:50	09:46	No
4	15-10-2014	20:37	21:12	Yes
5	24-10-2014	08:02	08:25	Yes
6	24-10-2014	20:05	20:27	Yes
7	29-10-2014	09:37	10:13	No
8	02-11-2014	23:33	23:40	Yes

The features are calculated for this training dataset. The resulting training dataset has the structure as shown in Table 4.2. The last column represents the class of data, where a *one* means that the row of data is representing features with a ML and a *zero* means that the corresponding row is representing features which are not part of a ML. The annotation of the class is carried out manually. The training array consist of two parts, the first part contains all ML data and runs from row one to N and the second part contains no ML data and runs from row N+1 to N+M, where N is 2481 and M is 6575. One can notice that the number of profiles containing no ML is much larger, but a large part of it contains no data (only NaN, no precipitation). As a result the data ratio is about 40/60. The trained model is stored in a structure so it can be used for performance evaluation or classification of new data.

Table 4.2: Training array format for detector

VPR nr.	Max-mean				Max-median				Peak height difference				Max-800m below max				Variance				Class		
	$Z_{e,hh}$	$Z_{DR}$	$Z_{e,hv}$	$\rho_{hv}$	$Z_{e,hh}$	$Z_{DR}$	$Z_{e,hv}$	$\rho_{hv}$	$Z_{e,hh}^-$	$Z_{e,hh}^-$	$Z_{e,hh}^-$	$Z_{DR}^-$	$Z_{DR}^-$	$Z_{e,hv}^-$	$Z_{e,hh}$	$Z_{DR}$	$Z_{e,hv}$	$\rho_{hv}$	$Z_{e,hh}$	$Z_{DR}$		$Z_{e,hv}$	$\rho_{hv}$
1	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	1
2	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	1
3	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	1
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
N	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	1
N+1	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	0
N+2	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	0
N+3	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	0
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
N+M	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	0

Now the training data can be generated, a selection of machine learning algorithms is necessary to create the models.

### Detector machine learning algorithms

The ML detector can be created using different types of machine learning algorithms. The ML detection problem is essentially a two-class classification problem where the training is done in a supervised way. The

two classes are: there is a ML in a profile, or there is not. Used machine learning algorithms for such a problem can be (composed from <https://elitedatascience.com/machine-learning-algorithms> and Bhavsar and Ganatra (2012)):

- Logistic regression models
  - Advantage: Outputs have a nice probabilistic interpretation and the algorithm can be regularised to avoid overfitting;
  - Disadvantage: Tends to underperform when there are multiple or non-linear decision boundaries.
- Classification trees (ensembles)
  - Advantage: Classification tree ensembles perform very well in practice. They are robust to outliers, scalable, able to model non-linear decision boundaries, easy to interpret and visualise;
  - Disadvantage: If unconstrained, individual trees are prone to overfitting, but this can be prevented using ensembles.
- Support vector machines
  - Advantage: SVM's can model non-linear decision boundaries, are robust against overfitting and require only a few training samples;
  - Disadvantage: SVM's are memory intensive and do not scale well to large datasets. Training data over 10000 examples makes it very slow. They are difficult to interpret.
- K-nearest Neighbour
  - Advantage: Relatively fast in training the algorithm, easy to understand the operating principle and perform well on applications where a sample can have many class labels;
  - Disadvantage: Relatively slow in classifying new data and require large amounts of storage space.
- Deep learning
  - Advantage: Deep learning performs very well in classifying audio, text and images. There is no need for feature engineering, since deep learning provides it for you;
  - Disadvantage: Deep learning requires a vast amount of data to learn a classifier.

All of them have different advantages and disadvantages. Based on the above-mentioned advantages and disadvantages two different algorithms will be tested in the ML detector performance evaluation: SVM and ensemble classification trees. To train the models, the classification learner application of MATLAB is used. All default settings for learning of the models are maintained in first instance. MATLAB offers six types of SVM classifiers. The difference between the classifiers is mainly on what type of method is used to calculate the optimal decision boundaries. To limit the number of used algorithms 3 types are used to compare the difference in detection performance. The following machine learned methods will be used for performance evaluation:

- *Linear SVM;*
- *Quadratic SVM;*
- *Coarse Gaussian SVM;*
- *Ensemble of bagged trees.*

Based on the best performing detector, the next step of the ML characterisation method is initiated.

#### 4.4.2. ML attributer

The second step in the ML classification process involves the identification of range resolution cells which are part of the ML. In this step, it is assumed that the previous step successfully detected a profile with a ML signature. Compared to the detection process, the attribution process requires additional pre- and post-processing steps. The attribution process therefore becomes:

- Attributer features;
  - Pre-processing;
- Attributer training;
  - Pre-processing;

- Attributer machine learning algorithms;
- Post-processing.

### Attributer features

Features are again needed to learn when a range resolution cell belongs to the ML. Since we now are going to classify every individual resolution cell, features which describe the properties or characteristics of the individual cells are required. It makes sense to use the individual cell values of  $Z_{e,hh}$ ,  $Z_{e,hv}$ ,  $Z_{DR}$ ,  $L_{DR}$  and  $\rho_{hv}$  as features. Using the cells values as features gives two issues in the foreseen application:

1. Individual resolution cells have varying values in different datasets;
2. Individual cells of  $Z_{e,hh}$ ,  $Z_{e,hv}$ ,  $Z_{DR}$ ,  $L_{DR}$  and  $\rho_{hv}$  do not have coinciding ML characteristics;

To cope with these issues two pre-processing steps are proposed. The pre-processing step to cope with the first issue is applied before feature extraction and the second is applied before creation of the training dataset.

**Pre-processing before feature extraction** The first issue is related to fact that the ML exists in different scenarios. This issue is shown in Figure 4.8a. The figure shows two vertical  $Z_{e,hh}$  profiles from one dataset. Both have a ML signature, but they have totally different magnitudes. The "low case" has a peak  $Z_{e,hh}$  in the ML close to 19 dBZ, while that same 19 dBZ is less than the  $Z_{e,hh}$  in the rain of the "high case". Thus, the same value of  $Z_{e,hh}$  could be a "ML" cell and a "rain" cell depending on the scenario. It is very difficult for an algorithm to learn the right boundary if a cell value could both be inside and outside the ML. A solution can be found in enforcing thresholds on the data, just as the reference method uses. Another way to overcome this issue is to normalise the individual profiles. The result of a linear normalisation between 0 and 1 of the same example is shown in Figure 4.8b. The result for both profiles is that the peaks now have the same magnitude (e.g. one). Other normalisation schemes might be used, for now linear normalisation is being used. The resulting proposed feature set becomes:

- $\overline{Z_{e,hh}}$
- $\overline{Z_{e,hv}}$
- $\overline{Z_{DR}}$
- $\overline{L_{DR}}$
- $\overline{\rho_{hv}}$

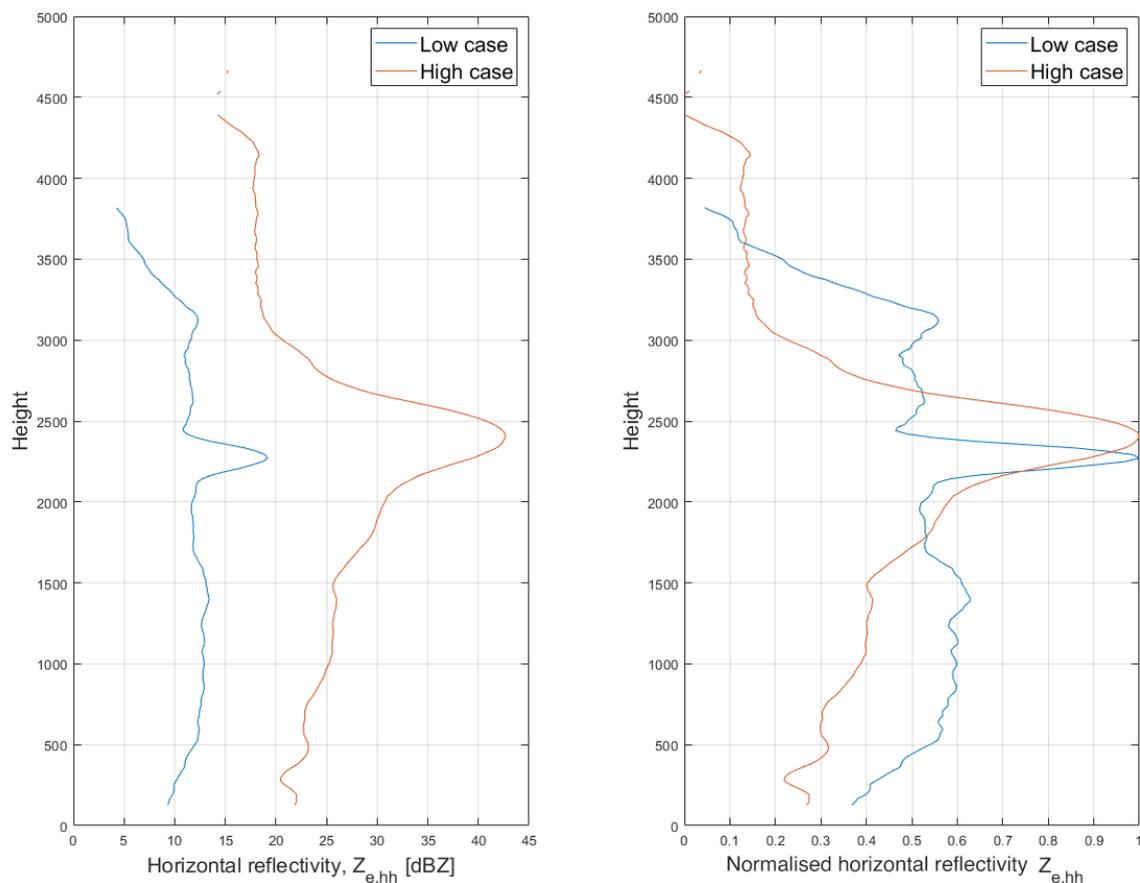
Where the line above the features is indicating the linear normalisation.

The proposed 10 dB SNR threshold in section 3.4 is increased to 15 dB as a result of the unreliable polarimetric measurements in the upper parts of the measurements. The threshold increase improves the removal of those parts of the data.

### Attributer training

The next step is to create a labelled training dataset filled with normalised cell values. The training dataset should contain data which is part of the ML and data outside the ML. For the data which is part of the ML, it is not useful to select all the ML data between the true boundaries (indicated in Figure 4.9a).

**Pre-processing before training data generation** This relates to the second presented issue when the attributer features were introduced. The issue relates to the radar observables reacting to different physical phenomena within the ML. It is shown in Figure 4.9a that the altitude of the ML signature of  $Z_{e,hh}$  is well above the ML signature of  $Z_{DR}$ . This poses a problem for ML detection based purely on the cell data values. One cannot determine a ML cell only based on a slightly higher value of the  $Z_{e,hh}$ . It is the combination of higher values of multiple radar observables which is able to indicate the melting layer. That combination only exists in the centre of the ML (roughly between 2000 m and 2500 m in Figure 4.9a) and not at the (true) boundaries. The example only shows the vertical profiles of  $Z_{e,hh}$  and  $Z_{DR}$ , while we have more observables available. It is to indicate the problem. The selection of a ML cell should preferably be based on at least on two radar observables showing a ML signature. The solution is to use the cell values of all available radar observables of the centre of the ML as shown in Figure 4.9b as the training dataset. The exact difference between the true



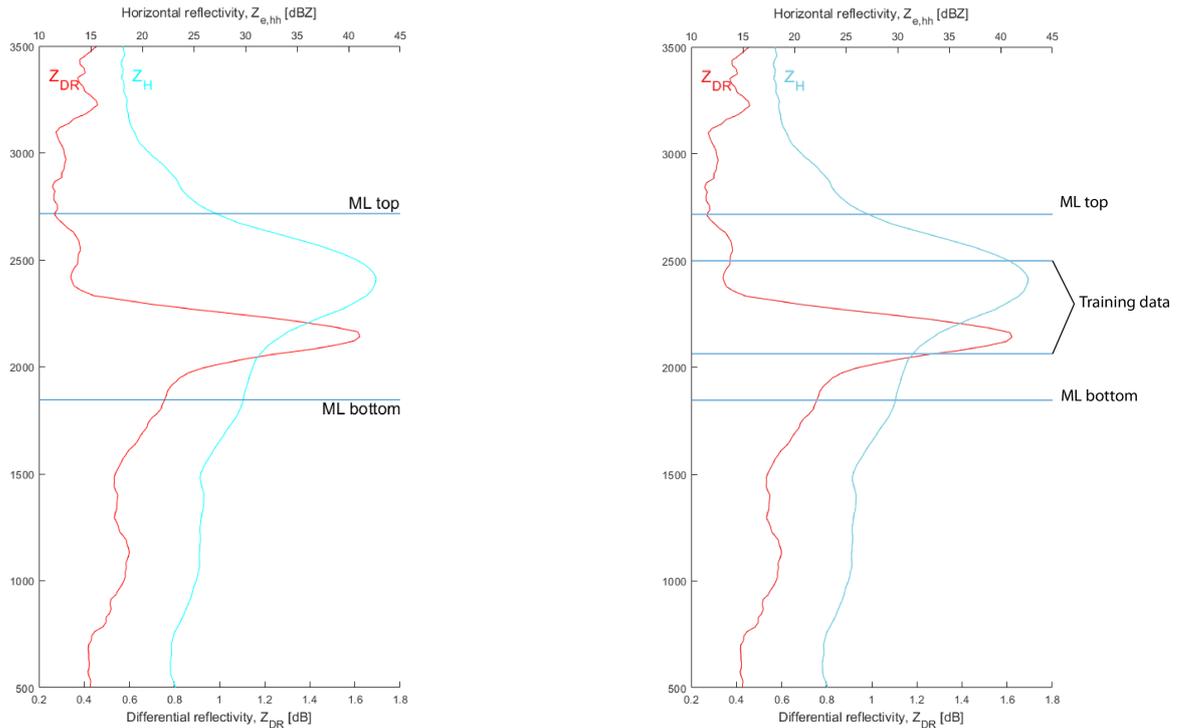
(a) Example of two horizontal equivalent reflectivity profiles in two different data sets. The ML signature in the 'Low' case has a maximum value of 19 dBZ, while this value is not even close to the ML values of the 'High' case.

(b) Example of the same profiles, but now linearly normalised from 0 to 1.

Figure 4.8: A ML signature in two different datasets.

boundaries of the ML and the boundaries of the training dataset will be set to 10 resolution cells (or 212 m). This ensures that, for the training dataset, ML cells show a ML signature of at least 2 radar observables. Lower values result in a ML signature which is not shown by at least two observables and higher values cause the thickness of the centre becoming negative in weak ML scenarios.

The labelled training dataset must contain cells from within the ML and outside the ML. The dataset used to create the data from within the ML is the same as used for training the detector (Table 4.1). The data indicated between the black lines in Figure 4.10a forms the ML data part of the training dataset. For the non-ML data part of the training dataset the dataset is supplemented as is shown in Table 4.3. This is to account for situations where above the ML high values of radar observables are common. The training data outside the ML is composed of cells above and below the ML as is shown in Figure 4.10b. This dataset only contains data with a ML. This might sound odd, but makes sense since the attributer is only applied when a ML has been detected by the ML detector. Not the entire dataset is thus used for training the attributer. This down-selection is done to restrict the amount of data for the learning process. Even with this selection the total amount of data remains 41039 resolution cells. The composition of the training dataset array is schematically shown in Table 4.4. The first set of rows (1 to  $N$ ) contains data from within the ML and thus the class is 1, the second set of rows ( $N+1$  to  $N+M$ ) contains data from outside the ML and thus the class is zero ( $N = 14464$  and  $M = 26575$ ). Note that all the values in the training array vary between 0 and 1 due to linear normalisation.



(a) Example of one profile. The ML signature is not overlapping between the radar observables which might be a problem for detection based on the values of the observables.

(b) Example of the same profiles, but shown with the boundaries of data being used in the training dataset.

Figure 4.9: Example and solution to the problem of non-overlapping features.

Table 4.3: Non-ML training dataset composition for attributer

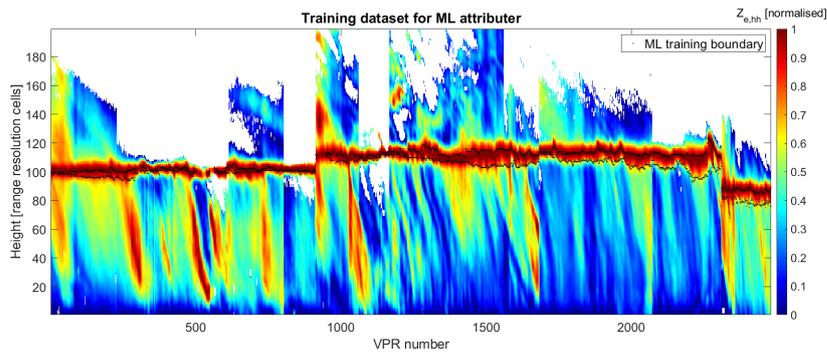
Part	Date	Time start	Time end	Contains ML
1	15-10-2014	20:37	21:12	Yes
2	24-10-2014	08:02	08:25	Yes
3	24-10-2014	20:05	20:27	Yes
4	02-11-2014	21:48	21:53	Yes
5	03-11-2014	07:47	07:58	Yes
6	12-11-2014	10:01	10:06	Yes

Now the training dataset is complete it can be used to train the attributer. The resulting learned model is stored in a structure so it can later be used for performance evaluation. Again, different machine learning algorithms can be used to create a model of the data. This selection is carried out next.

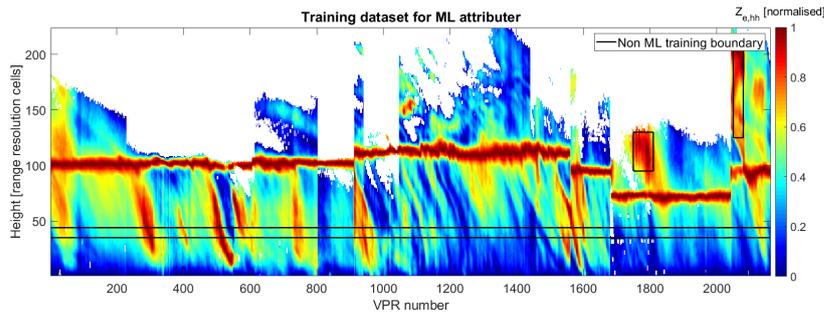
### Attributer machine learning algorithms

Based on the same comparison of machine learning algorithms as previously used in section 4.4.1, the choice of machine learning algorithms is made. For the attributer a K-nearest neighbour (KNN) is included because of the ability to cope with samples which can have labels of multiple classes. An example of this is shown in Appendix A. Based on the amount of training data, SVM's are not considered in this case, which means that for the attribution process two types of machine learning methods will be used where for the KNN method multiple variations are compared:

- *KNN fine*;
- *KNN Medium*;



(a) Data selection of ML data for learning the attributer.



(b) Data selection of Non-ML data for learning the attributer.

Figure 4.10: The composition of training data for the attribution.

Table 4.4: Training array format for attributer

Cell nr.	$\overline{Z_{e,hh}}$	$\overline{Z_{DR}}$	$\overline{L_{DR}}$	$\overline{Z_{e,hv}}$	$\overline{\rho_{hv}}$	Class
1	...	...	...	...	...	1
2	...	...	...	...	...	1
3	...	...	...	...	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	...	...	...	...	...	1
N+1	...	...	...	...	...	0
N+2	...	...	...	...	...	0
N+3	...	...	...	...	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N+M	...	...	...	...	...	0

- *KNN Coarse*;
- *KNN Cosine*;
- *KNN Cubic*;
- *KNN Weighted*;
- *Ensemble of bagged trees*.

The difference between the KNN algorithms is mainly on how many neighbours are required for classification and what the distance metric between the neighbours is. Fine KNN needs only one neighbour, while the coarse KNN needs 100 neighbours. Greater details may be noticed by the algorithm using the Fine KNN method, but there is also a greater risk of over fitting. Again, the classification learner application in MATLAB is used to train the models, while in first instance the default setting for the models are maintained. Only

the default option of standardisation of data is switched off. This setting is not contributing since the data is already normalised. The output of the attributer is essentially a black and white image, where the ML is indicated by ones. Post-processing is next applied to the raw output of the attributer.

**Post-processing** Post processing is necessary for two reasons:

1. *Removal of false positives;*
2. *Stretch the remaining attributions to expected ML thickness.*

It is very likely that some cells do not belong to the ML while they are classified as ML cells (False Positives). A method to remove these false positives is by enforcing a time and thickness threshold on the attributions. A ML generally does not happen for a few seconds and is not just one cell thick, it is thus fair to say that for any attribution being made, the length of the attributed ML cells must be at least one and a half minute and at least 3 cells thick. All attributions with a shorter lifespan than one and a half minute and three cells thick are removed. To perform this operation the same technique as used in section 3.4 is applied to the output of the attributer. The black and white image coming from the attributer is first eroded with a rectangular structuring element (height = 3 cells and length = 30 cells) and then dilated with the same structuring element. The erosion removes all attributions shorter than 30 profiles long and 3 cells thick, while the dilation restores the remaining output to its original size.

Since only the centre of the ML was selected as training data (Figure 4.9b), only the centre of the ML will be the output of the attributer. This will never coincide with the boundaries of ML definition. The solution is to vertically stretch the attributions by the same amount of resolution cells as has been used to define the centre, which was 10 resolution cells. This operation can be performed with image dilation with a vertical line structuring element of 20 resolution cells. After applying post-processing steps to the output of the attributer, a useful result is generated and performance can be calculated with the proposed method of the next section.

## 4.5. Evaluation of identification methods

After implementation of all the detection and characterisation methods, their performance should be computed and compared. The output of the detection and characterisation methods is essentially a black and white image. The output consists of ones (white) where a ML cell has been detected and zeros (black) where no ML cell has been detected. The performance evaluation will consist of two parts:

1. *Calculate and evaluate detection performance;*
2. *Calculate and evaluate differences between ML boundaries.*

The first consists of determining which vertical profiles contains detected ML cells in case of the reference method and take the output of the ML detector from the machine learning method. This can then be compared to the ground truth of the test dataset. In the second step the ML boundaries are determined based on the lowest and highest identified ML cells. The lower and upper boundary are respectively equal to the lowest detected ML cell and highest detected ML cell per vertical profile.

The previous paragraph already spoke about a test dataset. The test dataset is generated from a part of the ACCEPT campaign data which is not part of any of the training datasets. The dataset contains data from the entire ACCEPT campaign and is composed of data from different scenarios. The labelled test dataset is built from the following parts of the ACCEPT campaign: Two elements need to be added to create an annotated

Table 4.5: Test dataset composition

Part	Date	Time start	Time end
1	12-10-2014	19:37	21:46
2	13-10-2014	01:07	02:45
3	15-10-2014	22:44	23:51
4	21-10-2014	05:48	08:26
5	24-10-2014	20:32	21:27
6	03-11-2014	00:04	00:48
7	03-11-2014	01:02	02:03
8	07-11-2014	09:22	11:13

test dataset: true detection labels and true ML boundaries. The first is mostly carried out manually since the definition of the ML does not indicate when a ML exists or not. A ground truth array is formed equal to the length of the test dataset. In the array per vertical profile a one is stored if a ML is present and a zero is stored if no ML is present. The second element is partially automated using the ML definition on the profiles which have a ML detection label. The result of the ML definition is a lower and upper boundary of the ML in that profile. This result is stored in the same ground truth array. As a result, the ground truth array looks like Table 4.6. Note that the unit of the boundaries stored in the array is in range resolution cells, this still has to be converted to meters. Now a test dataset with a ground truth is available, the performance evaluation starts

Table 4.6: Example of part of ground truth array (VP: Vertical profile)

Ground truth item	VP 1	VP 2	VP 3	VP 4	VP 5	VP 6	...	VP N
ML detection label	0	0	0	0	1	1	...	1
Lower boundary	NaN	NaN	NaN	NaN	84	85	...	90
Upper boundary	NaN	NaN	NaN	NaN	110	109	...	115

with the detection performance.

**Detection performance evaluation**

The output of the detector and the (detection) output of the reference method can be compared to the first row of the ground truth array. The outcome of the comparison is plotted in a confusion matrix. A confusion matrix is used to analyse the error types the detector makes. An example of a confusion matrix is shown in Figure 4.11. In a confusion matrix, the output of the classifier is plotted against the true values. Four outcomes are possible:

- *TP: True Positive. The detector reports a ML (one) and the ground truth also reports a ML (one);*
- *TN: True Negative. The detector reports no ML (zero) and the ground truth reports no ML (zero);*
- *FP: False Positive. The detector reports a ML (one) while the ground truth reports no ML (zero);*
- *FN: False negative. The detector reports no ML (zero) while the ground truth reports a ML (one).*

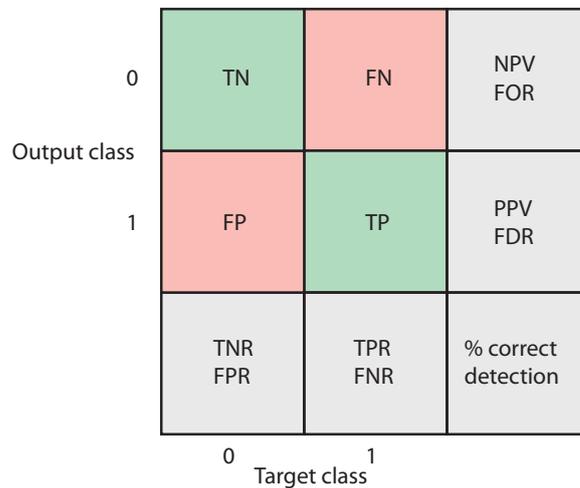


Figure 4.11: Example of a confusion matrix. The outcome of the classifier is indicated as "Output class", while the ground truth is indicated by "True class".

The other fields of the confusion matrix are:

- *FPR: False Positive Rate;*
- *TNR: True Negative Rate;*

- TPR: True Positive Rate;
- *FNR: False Negative Rate;*
- NPV: Negative Predictive Value;
- FOR: False Omission Rate;
- *PPV: Positive Predictive Value;*
- FDR: False Discovery rate.

The confusion matrices are calculated for all proposed detection methods. The reference method is based on a fixed set of rules and thus the output of the detector does not change if the method is run again on the same test dataset. This is not the case with machine learning methods. Training a model based on a training dataset involves using randomness. Even if the machine learning method is re-trained using the same training data it is likely that the output model is slightly different. One cause is the use of a 5-fold cross validation to prevent overfitting of the model. This process cuts the entire training dataset in two, trains the model with one part of the data and uses the rest for validation. This is then repeated 5 times with different cuts of data. The solution is to train the model 100 times sequentially and perform the detection on the test dataset on each of them. The mean and standard deviation are computed which gives a proper understanding of the magnitude of the differences between the trained models.

### **Attribution performance evaluation**

The performance evaluation of the attribution process is similar to the detection process. The attribution process of the reference method is not analysed separately, since the reference method has no separate detection and attribution. However, the attribution processes of different machine learned methods will be analysed. From the test dataset only the data which contains a ML (according to ground truth array) is used to analyse the attribution performance, assuming that the detection process was perfect. The machine learned attributers are then applied to the dataset and performance can be measured and displayed in a confusion matrix. Again, the attribution performance is changing slightly from model to model and as a result, the mean and standard deviation are calculated using a loop which trains and tests the model 100 times. This is only performed for the ensemble of bagged trees, since for a KNN algorithm the output will not differ from each learning process. For evaluating the performance of the attributer, only the raw output (non-post-processed output) of the attributer is used. The output after post processing is only evaluated in combination with the detector. When performing the evaluation of the attributer, we have to take into account that the centre of the ML is used to train the attributer and thus the centre must also be used to base the performance evaluation on.

Once the individual detection and attribution processes have been analysed, the performance of both processes combined shall be assessed. For the reference method, both processes are already combined, but this is not the case for the machine learning process. Now, the actual output of the detector is used to start the attribution process and the total performance is evaluated. Multiple different errors can occur as a result of this combination:

- Detector true positive
  - Attributer true positive
  - Attributer true negative
  - Attributer false positive
  - Attributer false negative
- Detector false positive
  - Attributer false positive
  - Attributer true negative
- Detector true or false negative
  - Attributer not started

This means that the attributer has a chance in repairing the outcome if the detector generated a false positive, but only if the attributer generates true negatives. In case of detector false positives, the attributer can only make two types of outputs, false positives and true negatives. If any positive output is generated it is by

definition a false positive, since no ML is present and if any negative output is generated it must be a true negative, since no ML is present.

The output of both characterisation methods is then compared to the top and bottom boundaries of the labelled test dataset. The RMSe, mean error and correlation coefficient will be calculated for both boundaries to indicate the characterisation performance of the ML boundaries. The mean error is calculated by subtracting the true boundary from the detected boundary and take the mean of the resulting array. This means that a positive error indicates a detected ML boundary above the true boundary and consequently a negative mean error indicates a mean detected ML boundary below the true boundary. The RMSe is calculated by subtracting the true boundary from the detected boundary. The errors are then squared, the mean is calculated and the square root is taken from the result. The RMSe relates to the size of the total error, the mean relates to the mean direction of the error and the correlation coefficient relates to the similarity in shape of the boundaries.



# 5

## Results

This Chapter shows the results which have been obtained with the proposed ML characterisation methods. A performance evaluation of the individual methods is presented, while during the evaluation the results are being discussed.

### 5.1. Reference ML characterisation method

The updated method of Giangrande et al. (2008) is implemented according to section 4.2. As a short summary the following steps are carried out for ML detection. The first step is based on a preselection of data cells which have a copolar correlation coefficient between 0.85 and 0.97. The second step is to determine the maximum  $Z_{e,hh}$  and limit those found maxima between 30 dBZ and 49 dBZ. The third step is to perform the same type of selection procedure on the  $Z_{DR}$ . The maxima of the  $Z_{DR}$  is selected and limited to values between 0.8 dB and 2.5 dB. The final step in the procedure is to select all the cells from the preselection of  $\rho_{hv}$  and check for each cell if the maximum of the  $Z_{e,hh}$  and  $Z_{DR}$  lies between 500 m below or 200 m above the cell in question. When a cell complies to all criteria, a ML cell is registered.

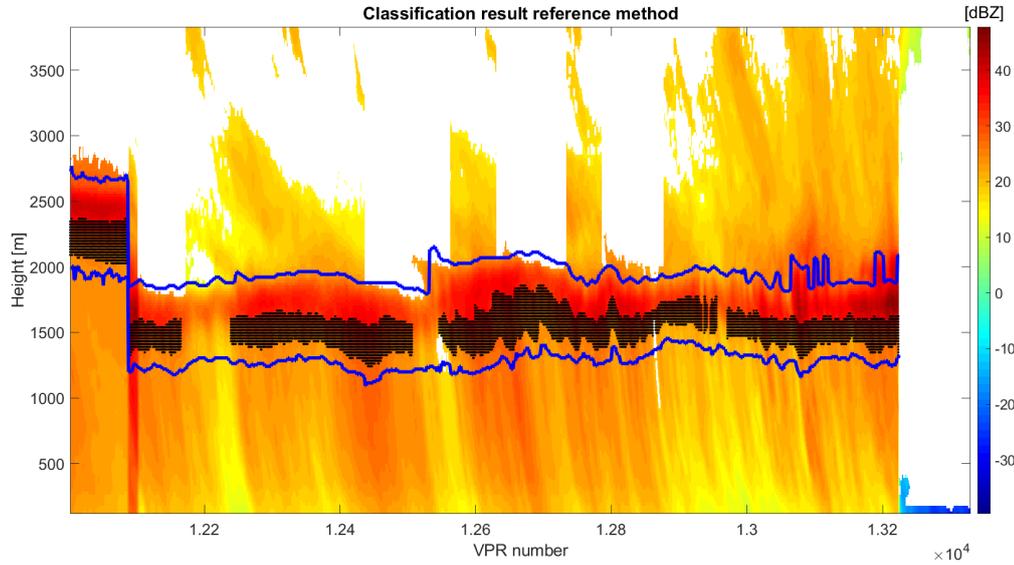
When the reference method is applied to the test dataset, the detection result is shown in Figure 5.1. The overall accuracy of the detection is 80.1%. This means that from all 17414 vertical profiles, 80.1% were correctly classified. The strength of the reference method is that no false positives were registered. This means that by using this method, if there is detection, almost certainly you can assume that there really is a ML. On the other hand, it also misses vertical profiles which do have a ML signature. Almost 3500 (20%) of the 17414 profiles were not detected while they contained a ML signature. Even more indicative is the TPR, only 52.3% is detected. An example of the false negatives is given in Figure 5.2a. In this example some interruptions

Confusion matrix reference method

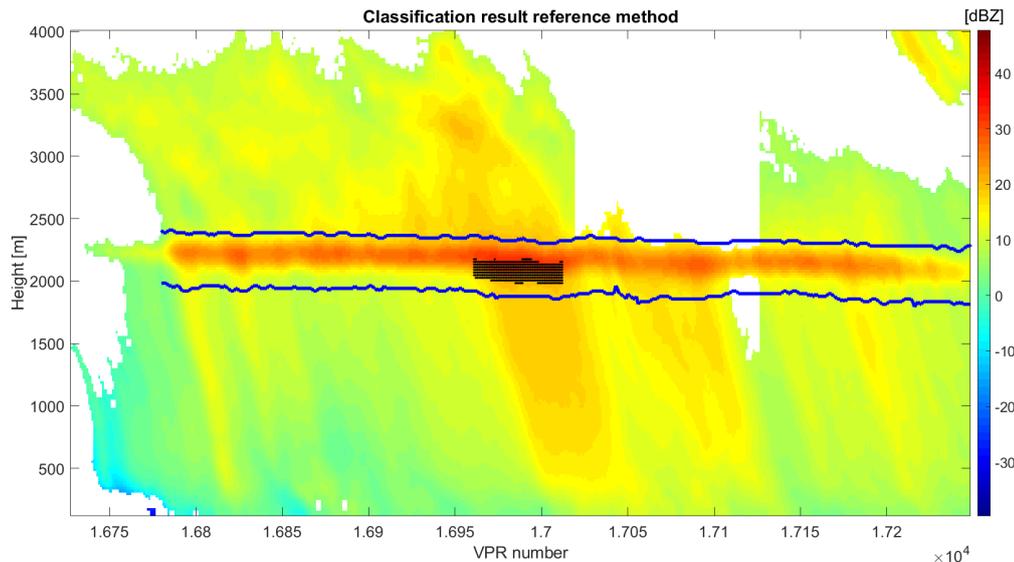
	0	1		
Output class	0	10136 58.2%	3470 19.9%	74.5% 25.5%
	1	0 0.0%	3808 21.9%	100% 0.0%
		100% 0.0%	52.3% 47.7%	80.1% 19.9%
		0	1	
		Target class		

Figure 5.1: Confusion matrix of the reference method. Note that no false positives were initiated by the method.

around VPR numbers  $1.22 \times 10^4$ ,  $1.255 \times 10^4$  and  $1.295 \times 10^4$  in are shown. In all those cases it is clear that the ML itself is not interrupted. The reference method fails to detect the ML here because the peak  $Z_{e,hh}$  in the ML is below the threshold of 30 dBZ. The second image in Figure 5.2 shows a similar result, but now most of



(a) Example of detection result of the reference method on the test dataset at an instant where a ML signature is present. The reference method shows some interruptions around VPR numbers  $1.22 \times 10^4$ ,  $1.255 \times 10^4$  and  $1.295 \times 10^4$ .



(b) Example of detection result of the reference method on the test dataset at an instant where a clear ML signature was present. The method missed most of the ML unless the peak  $Z_{e,hh}$  reached the threshold around VPR numbers  $1.695 \times 10^4$  -  $1.7 \times 10^4$ .

Figure 5.2: Two examples of instants where the reference method did not detect the ML. The blue lines indicate the true ML boundaries according to the definition and the black dots indicate the identified ML cells.

the time the  $Z_{e,hh}$  is lower than the used threshold. Only a small portion of the ML is detected around the VPR numbers  $1.695 \times 10^4$  -  $1.7 \times 10^4$ . Both images in Figure 5.2 illustrate why the confusion matrix contains a high number of false negatives.

A different example of false negatives is shown in Figure 5.3. Here a small portion of the ML has not been detected. The undetected portion is indicated by the upper black ellipse. This missed detection is caused by the fact that the peak  $Z_{DR}$  is far outside the distance thresholds of 200 m and 500 m. In the example the peak  $Z_{DR}$  is indicated by the lower circle around 300 m altitude. This peak  $Z_{DR}$  close to the ground is probably caused by clutter and has not been removed by TARA clutter suppression. This indicates that the reference

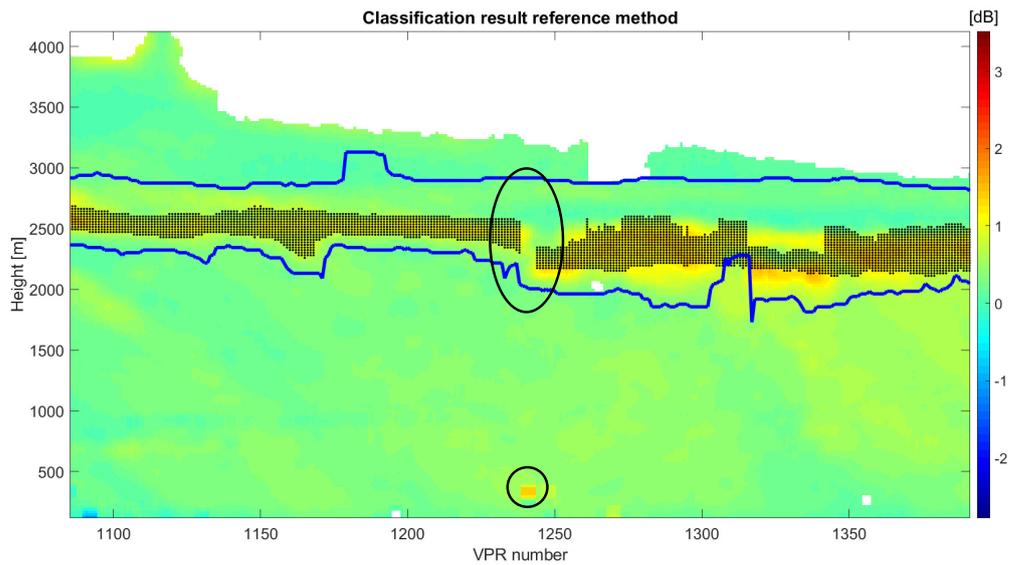


Figure 5.3: Example of detection result of the reference method with the  $Z_{DR}$  as the background image of the test dataset at an instance where a clear ML signature was present. The method resulted in a false negative because the peak  $Z_{DR}$  is much lower than the threshold of 500 m below the cell under investigation.

method is very sensitive to small changes in data consistency. In this case the peak  $Z_{DR}$  is probably caused by a small artefact in the data, this immediately translates to a missed detection by the reference method.

Lastly attention is given to the ML boundary characterisation error of the reference method. When the identified boundaries are considered, we expect, based on the analyses of section 4.2, that the upper ML boundary will be too low. The detection result of the reference method is shown in Figure 5.4, where we see that indeed the identified ML cells are only present in the lower part of the total ML. The fact that almost

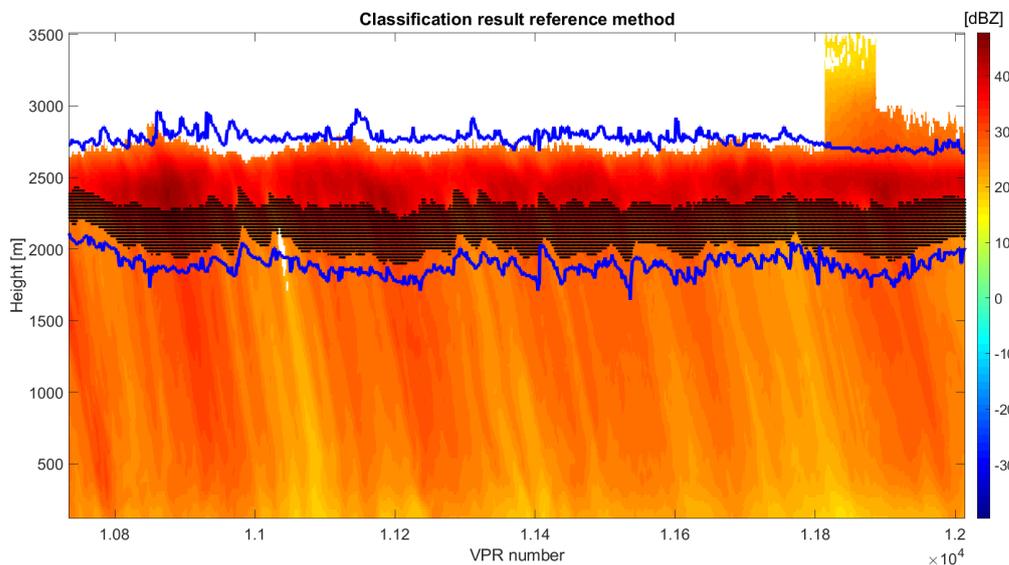


Figure 5.4: Example of detection result of the reference method on the test dataset at an instance where a clear ML signature was present. The blue lines indicate the true ML boundaries according to the definition and the black dots indicate the identified ML cells. The reference method identifies ML cells in the lower part of the total ML, but misses the top part.

the entire top half of the ML missed is a negative point. What also can be noticed from Figure 5.4 is that the lower boundary of the definition and the lower boundary of the detection result very nicely follow each other. There is almost a constant difference of 4 to 6 range resolution cells (85 m- 127 m) even if the boundary

changes rapidly. This method is therefore suitable to analyse the time variability of the lower boundary. A constant difference between the boundaries does not affect the outcome if the time variability of the boundary is analysed.

The performance metrics of the boundary characterisation error is shown in Table 5.1. As expected, the error towards the upper boundary is large compared to the error of the lower boundary. The minus sign is indicating a lower detected boundary compared to the true boundary. Also, a higher correlation with the lower boundary exists compared to the upper boundary.

Table 5.1: Errors w.r.t. the true ML boundaries.

Boundary	RMSe [m]	Mean error [m]	Correlation
Upper	382	-365	0.9654
Lower	149	137	0.9865

To conclude on the reference method: the choice of a high threshold for the ML lowest reflectivity value results in false negatives (ML data missed) while showing no false positives. The lower boundary of the reference method follows the lower boundary of the definition very well, which is very useful in the analysis of the time variability of the lower boundary. The upper boundary has a lower correlation coefficient and has much higher RMSe and mean error. It is shown that most of the upper half of the ML is missed. This result is an undesirable property if one wants to analyse the entire ML. While the method follows the lower boundary very well, it still has a mean error of 137 m above the true boundary. Therefore, the reference method is not very well suited for the purpose of this thesis, but it is able to serve as a baseline method to which the other developed methods can be compared.

## 5.2. Image processing ML characterisation methods

### 5.2.1. Hough transform

The Hough transform method is implemented according to subsection 4.3.1 and tested on the image of the copolar correlation coefficient from the already known clear ML scenario. The Hough transform of the clear ML scenario is shown in Figure 5.5. In the Hough parameter space, it is seen that there is clear convergence of lines around a 90° angle. This makes sense according to the theory (Figure 4.5), since that means that the Hough transform might be able to sense lines in horizontal direction. The result of the Hough transform

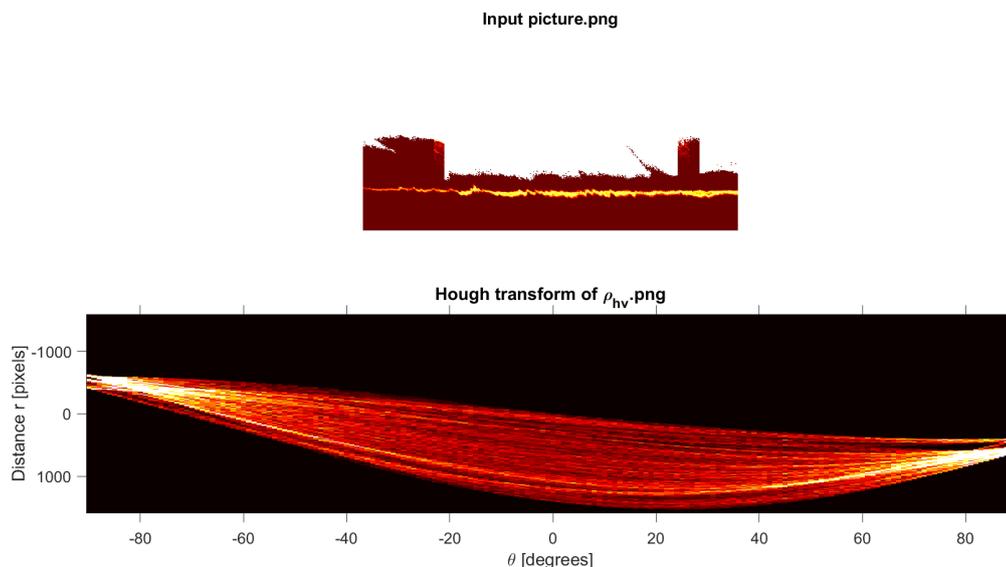


Figure 5.5: Upper picture is the input image, lower picture is the Hough parameter space as explained in subsection 4.3.1.

is shown as overlay on the original image in Figure 5.6. It is clearly visible that the method detects a lot of horizontal line sections on the melting layer itself, but it also detects multiple lines on the transition between

the upper part of the cloud and the part where no data is available. This imposes quite a risk. It is possible to adapt the Hough transform sensitivity and as such reduce the number of detected line sections. This still does not give a very satisfactory result. The method is also tried with various input images and differences in colour scaling. From that experimentation it is clear that the method is very dependent on how the input image is created.

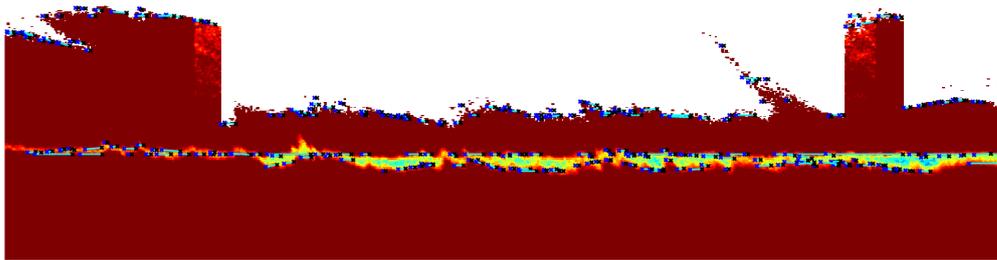


Figure 5.6: Result of horizontal line detection using the Hough transform.

**Conclusion on Hough transform method** From experimentations with the Hough transform method it is clear that this method is not suited for ML detection. Although the method succeeds in detecting the ML in the clear ML scenario, this is not the case for other scenarios and the success of the method is very dependent on colour scaling when the image is created. The fact that it is very depending on colour scaling could be used as well for optimal ML signature display, i.e. one could use the colour scaling to improve Hough line detection method. Since this involves lots of experimentation, while the results are uncertain, this method is not pursued any further.

### 5.2.2. Radon transform

The Radon transform is implemented according to subsection 4.3.2 and again tested on the image of the copolar correlation coefficient from the known study case of ML scenario. The Radon parameter space of the clear ML scenario is shown in Figure 5.7. Also, the three detections can be seen in the figure. The detections are made using the same peak detection technique as used with the Hough transform method. The peak detections lead to three detected lines with an angle ( $\theta$ ) and a vertical distance from the center of the image. The three lines are plotted over the original image and can be seen in Figure 5.8. The three lines coincide very well with the ML. While the result of the Radon transform method proved to be successful in detecting the ML in the clear ML scenario and the method is proved robust against false detections in that scenario, the result is not very successful when there are only very short pieces of ML visible in the image. The method is also very dependent on the creation of the input image and it is therefore decided not to pursue this method any further.

### 5.2.3. Connected pixels

The connected pixel method is implemented as described in subsection 4.3.3. Again, the method is tested using the image of the copolar correlation coefficient from the study case ML scenario. The result can be seen in Figure 5.9. In this figure it is visible that the ML is shown in red, while other connected areas are highlighted in other colours. This method proves to work and selects the ML based on the largest area of connected pixels.

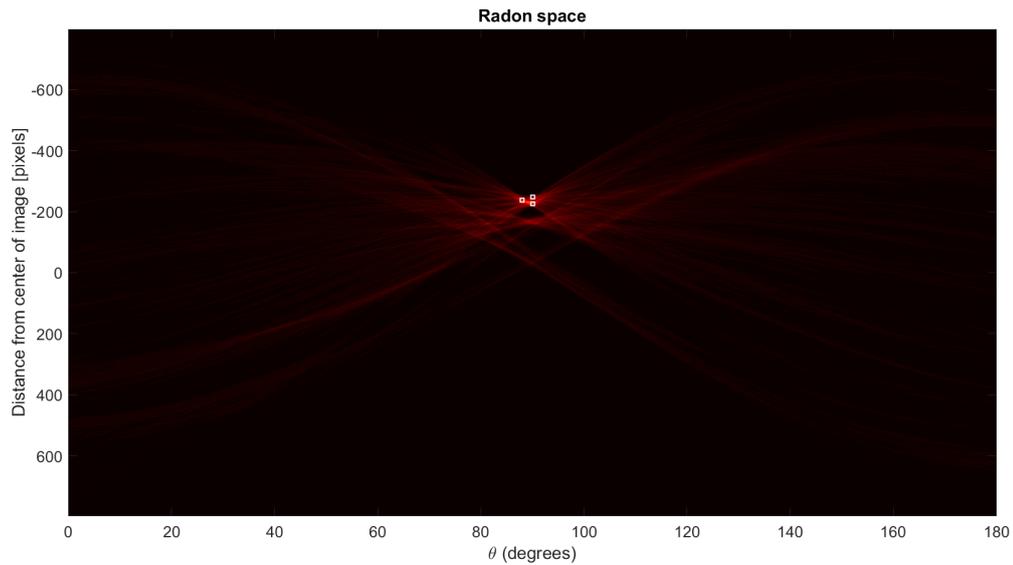


Figure 5.7: Radon transform of the clear ML scenario with three detections indicated by the white squares using peak detection method.

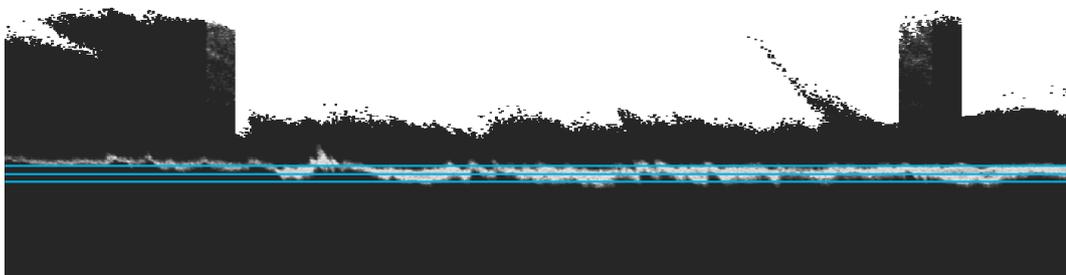


Figure 5.8: Result of horizontal line detection using the Radon transform.

**Conclusion on connected pixel method** This method has an advantage over the previous two methods being that it truly selects the pixels which belong to the ML, including the variability in altitude and intensity. It remains difficult to exactly know which pixels are selected during the transition from grey scale to black and white images. While this method works properly in the chosen study case, it does not perform well in scenarios when edges are connecting more elements together. Also, it now selects the largest pixel area to be the ML, but it might be that other larger areas are created and thus selected as the ML. This method also poses a new challenge, because it selects the largest connected pixels area, which means that it also selects the largest area even when there is no ML. If one wants to use this method for ML detection, some sort of ML detection algorithm needs to pre-select the ML data first. This method leaves much uncertainty about the detection reliability so this method will not be continued.

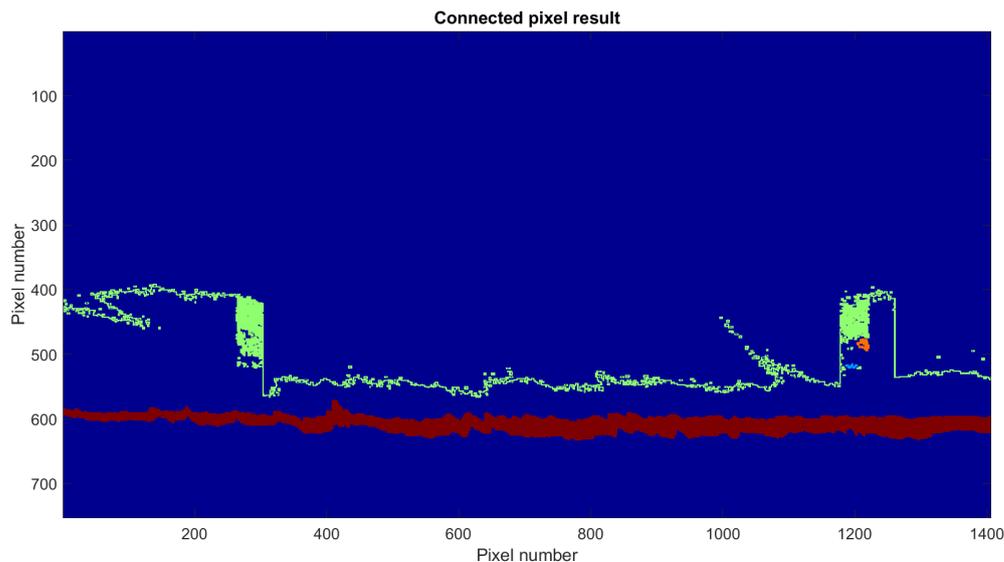


Figure 5.9: Result of connected pixel method.

### 5.3. Machine Learning ML characterisation method

The proposed machine learning characterisation method is implemented and the performance is analysed in this section. The method relies on a two-step procedure, the first being the ML detector and the second the ML attributer. Before the first and before the second step an analysis is carried out on the proposed feature sets. After the detector feature analysis, two different types of machine learning detectors are compared. In the second step of the characterisation method two machine learning attributers are compared.

#### 5.3.1. Machine learning ML detector

The available features are analysed based on the training dataset. Well defined features should have some characteristics:

- Ability to indicate separate classes;
- Spread within classes should be low;
- Should not be redundant.

The features need to be able to separate the classes. The features should have a low spread within the class otherwise the detector only works with a certain dataset and not with others. The feature set also needs to be different for each feature. As a general rule of thumb, one could say that good features are much better than a large number of weak features. A large number of weak features might work confusing, it costs much more processing time to learn a model and not all machine learning techniques can work well with large numbers of features.

A common method to judge the usability of features is to plot individual features in a scatter plot. This is performed for the proposed features. Not all scatter plots are shown, because that would make a total of  $\binom{k+n-1}{k} = \frac{(k+n-1)!}{k!(n-1)!} = 253$  where  $k = 2$  and  $n = 22$  scatter plots according to combinations with repetition. Therefore, only examples of interesting scatter plots will be shown. In Figure 5.10a and Figure 5.10b almost the same scatter plots are shown. The difference is the used feature sets which are max-mean in Figure 5.10a and max-median in Figure 5.10b. The similarity is therefore to be expected. For most machine learning algorithms doubling of features is not a problem, but it also does not contribute to a better detection result. It merely costs more processing resources since more features need to be analysed. Both feature sets have a good separating capability, which is shown by the two different classes mostly situated in a different part of the scatter plots. Also Figure 5.10c shows a bit of a similar scatter plot as the first two. This can be expected, because the features are using the same type of calculation, merely a different measure. Another example of a feature set with good separating capability is the set of variances of the  $Z_{e,hh}$  and  $Z_{e,hv}$ . This feature set is shown in Figure 5.10d. On the other hand, the variance is not always a good choice if the feature set of

Figure 5.10e is chosen. In this image the variances of both features are not able to separate the two classes. A different example is shown in Figure 5.10f. Here the height difference of the peaks is displayed. The height differences are not able to separate the classes very well. It can be noticed that if the height difference is really large, then the data is almost for sure not part of a ML. Based on the feature analysis it is expected that the following features of the initial feature set can be neglected without much impact on the detection result, while saving processing time as a bonus.

- Max-mean or max-median (redundant);
- All peak height differences (non-decisive);
- Variance of  $Z_{DR}$  and  $\rho_{hv}$  (non-decisive).

The remaining 10 features will be used in the second feature set. The resulting two sets of features are:

1. *Feature set 1: All proposed features;*
2. *Feature set 2: Max-mean, Max-800m below, variance of  $Z_{e,hh}$  and  $Z_{e,hv}$ .*

The first used machine learning detector is based on three SVM algorithms and the two defined feature sets. The second machine learning detector is the ensemble of bagged trees, again using both feature sets. The results will be shown in following order:

1. Linear SVM;
  - Feature set 1 (All features);
  - Feature set 2 (Subset of features);
2. Quadratic SVM;
  - Feature set 1 (All features);
  - Feature set 2 (Subset of features);
3. Coarse Gaussian SVM;
  - Feature set 1 (All features);
  - Feature set 2 (Subset of features);
4. Ensemble of decision trees;
  - Feature set 1 (All features);
  - Feature set 2 (Subset of features).

Where in the shown confusion matrices class 1 indicates a ML and class 0 indicates no ML. As can be seen in the confusion matrix of Figure 5.11a the overall accuracy on detection of the linear SVM is in total 92.6% (Figure 5.11a) when using feature set 1 and 93.8% (Figure 5.11b) when using feature set 2. For the Quadratic SVM detectors no difference can be seen in the overall accuracy on detection according to the confusion matrices shown in Figure 5.12a and Figure 5.12b. The quadratic SVM with feature set 2 shows a lower false positive rate compared to feature set 1. The coarse Gaussian SVM detectors show a very minor degradation in performance when looking at the confusion matrices shown in Figure 5.13a and Figure 5.13b. Note that because machine learning algorithms use randomness during the learning procedure, the detection performance numbers can change slightly from the same model learned with the same training dataset. Using the training dataset from this thesis, differences are in the order of 0.1% - 0.9% on the total detection performance. To give a better insight into the variance of the detection result a Monte Carlo run is carried out, which trains the model 100 times and generates a histogram of the detection result. The histograms are shown in Appendix B while the summary (mean and standard deviation) is shown in Table 5.2. The information listed in Table 5.2 indicate that the extra features do not contribute to a better detection performance for the linear SVM, only confusing the classifier. For the quadratic SVM detector, the full feature set shows a very slight increase in total detection performance and a reduction in standard deviation. It is questionable if the performance increase is worth the extra complexity. The choice of features seems to have no effect on the Coarse Gaussian SVM detector. Since it is assumed that for the second step of the ML characterisation method the detector makes no mistakes, a very low false positive rate and a high overall detection accuracy is preferred. The best SVM method for detection is therefore the linear SVM with the feature set 2.

The second machine learning algorithm used for the detection is the ensemble of bagged decision trees. The resulting confusion matrices are shown in Figure 5.14. In Figure 5.14a the ensemble is used with the feature set 1 and in Figure 5.14b, feature set 2 is used. From the confusion matrices we can conclude that

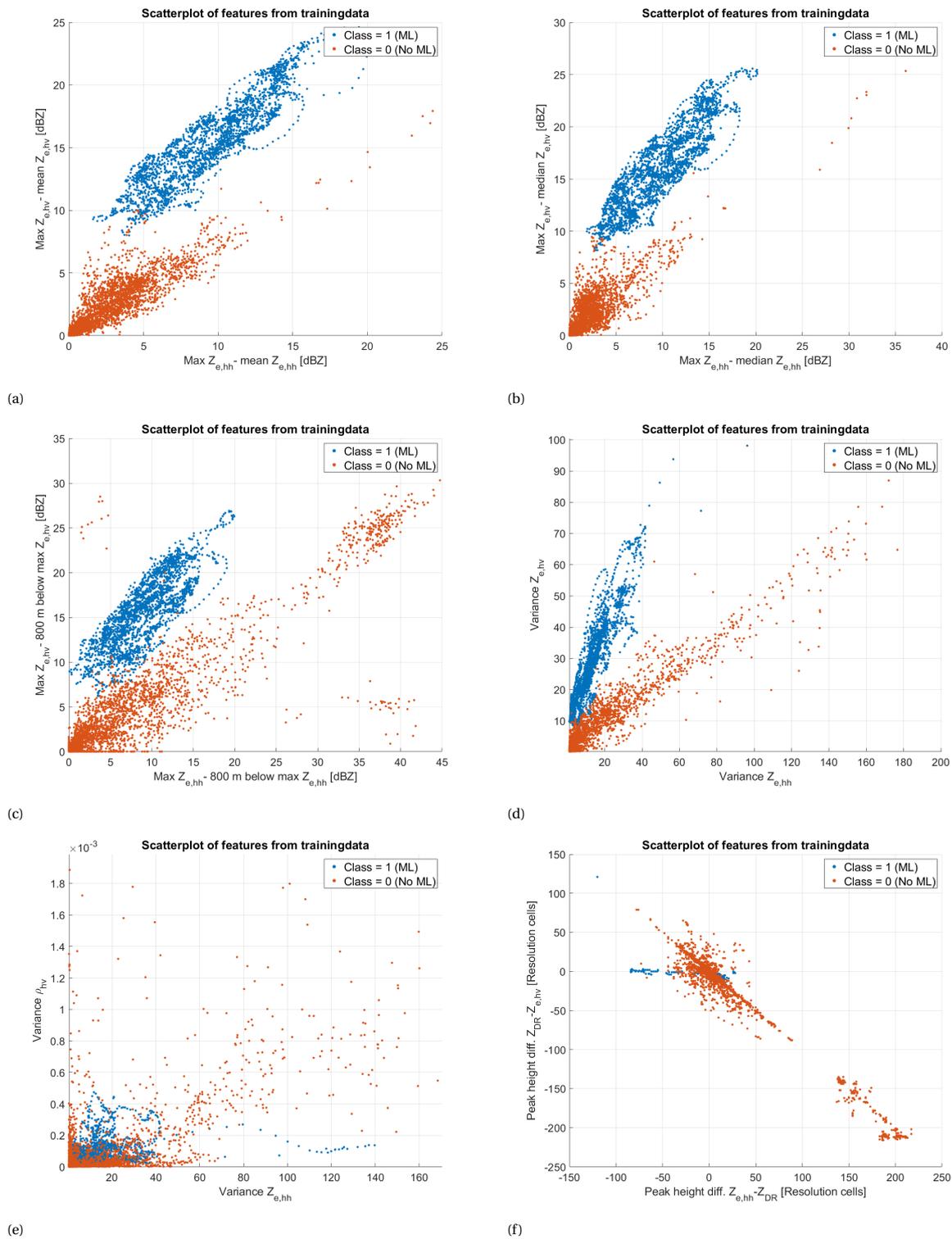


Figure 5.10: a-d: Examples of feature sets with good class separating capabilities. e-f: Examples of feature sets without class separating capabilities.

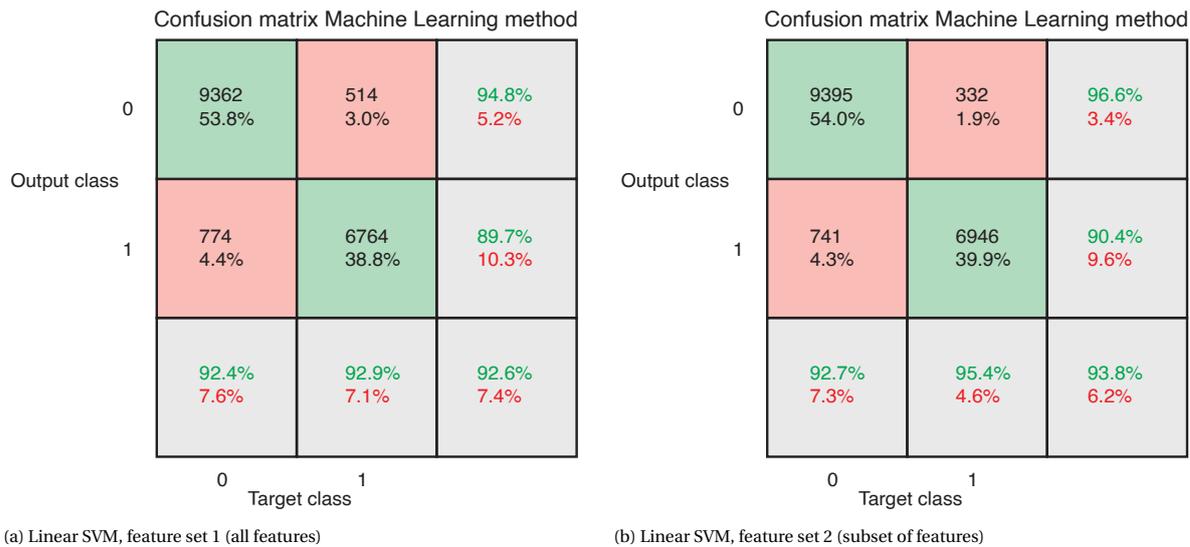


Figure 5.11: Confusion matrices of the detection performance of the linear SVM detectors using the full set of features (a) and the specified subset of features (b).

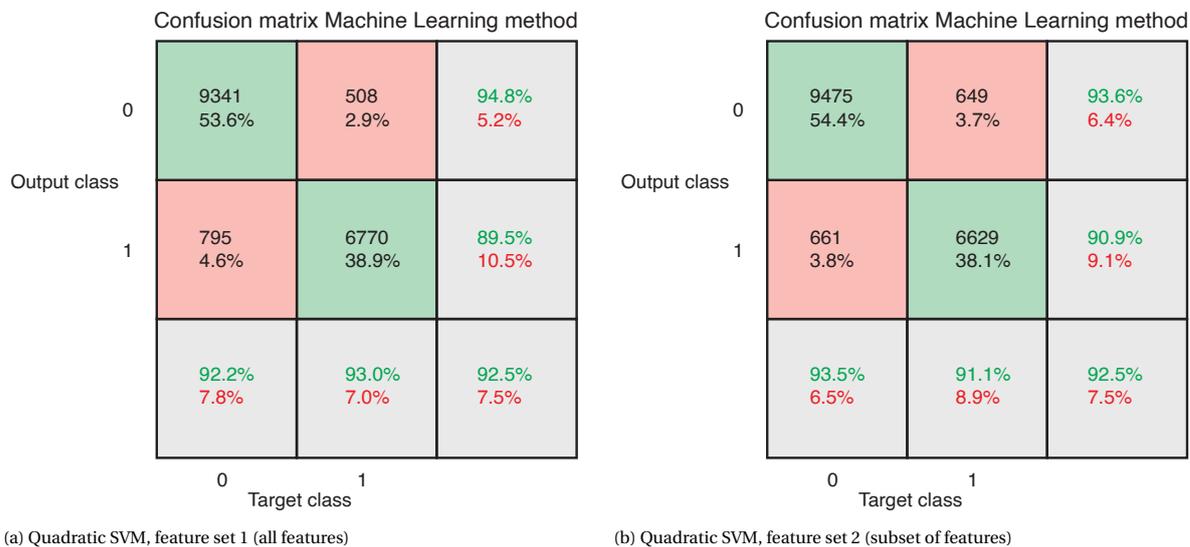


Figure 5.12: Confusion matrices of the detection performance of the Quadratic SVM detectors using the full set of features (a) and the specified subset of features (b).

Table 5.2: Mean and standard deviation of total detection performance.

Algorithm and feature set	$\mu_{DP}$ [%]	$\sigma_{DP}$ [%]
SVM linear, feature set 1 (full)	92.56	0.05
SVM linear, feature set 2 (subset)	93.84	0.05
SVM Quadratic, feature set 1 (full)	92.49	0.03
SVM Quadratic, feature set 2 (subset)	92.2	0.20
SVM Coarse Gaussian, feature set 1 (full)	92.9	0
SVM Coarse Gaussian, feature set 2 (subset)	92.9	0

the detection performance is not changing significantly due to the change in feature set. But, again, before a conclusion is drawn, the standard deviation of the detection result is calculated based on a 100 times Monte Carlo run. The result of the Monte Carlo run is shown in Table 5.3. The accompanying histograms are shown

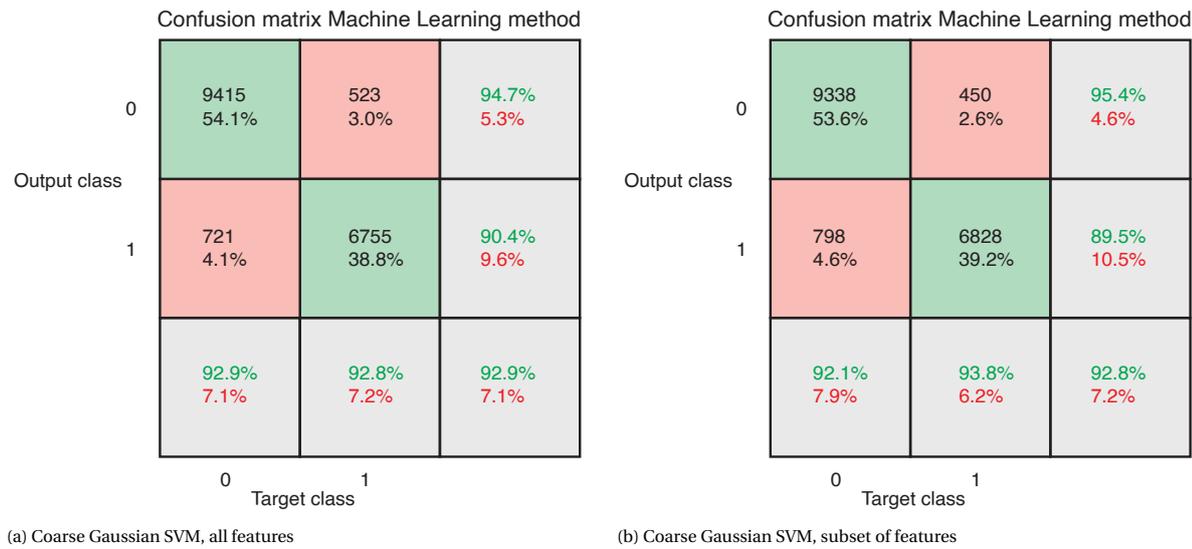


Figure 5.13: Confusion matrices of the detection performance of the coarse Gaussian SVM detectors using the full set of features (a) and the specified subset of features (b).

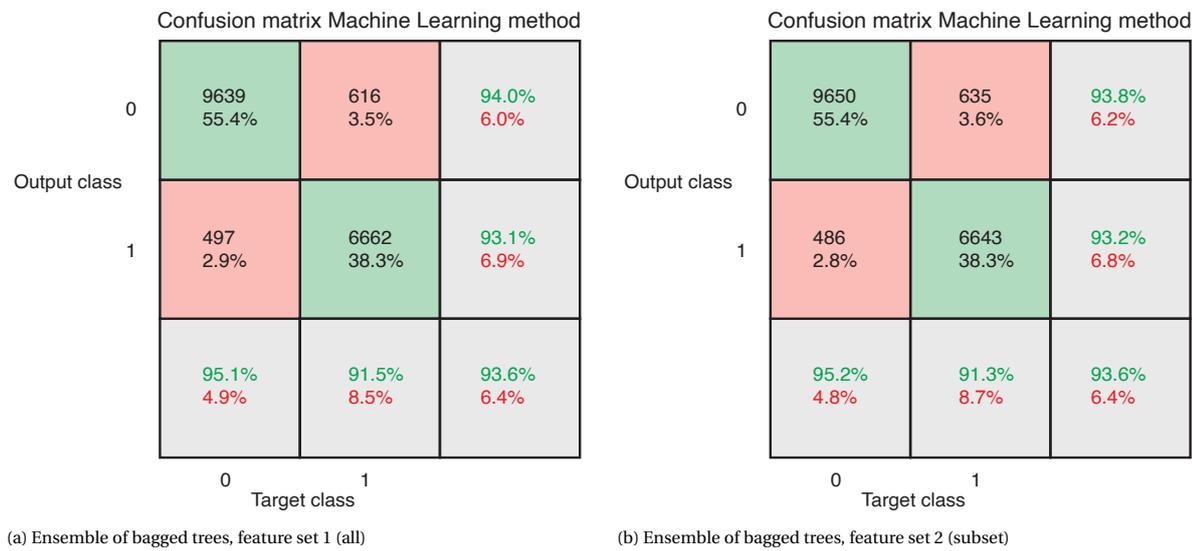


Figure 5.14: Confusion matrices of the detection performance of the ensemble of bagged trees detector using the full set of features (a) and the specified subset of features of the same detector (b)

in Appendix C. With the numbers from Table 5.3 in mind the extra features do not significantly contribute

Table 5.3: Mean and standard deviation of total detection performance.

Algorithm and feature set	$\mu_{DP}$ [%]	$\sigma_{DP}$ [%]
Ensemble, feature set 1 (full)	93.62	0.20
Ensemble, feature set 2 (subset)	93.46	0.46

to improve the overall detection performance and do not confuse the ensemble detector. The larger number of features do cause a more consistent performing ensemble detector, which is seen by the lower standard deviation. This is expected by the increase in features. The increased number of features gives more degrees of freedom and as a result the randomness in the training phase has a lower impact. Processing time has

increased by 37% as a result of the increased number of features. Which might be significant if really large datasets need to be analysed. Otherwise it is recommended to use the full feature set.

Based on the previous results the choice of detector is between the ensemble of bagged trees and the linear SVM. The linear SVM has the advantage that the mean total detection performance is slightly higher (93.8% compared to 93.6% of the ensemble) and the standard deviation of the SVM is much smaller. On the other hand, the number of false positives is higher than the ensemble detector (4.3% for SVM and 2.8% for ensemble). Also, to compare this objectively, Monte Carlo runs are necessary to compare the mean false positive rates and false negative rates and the standard deviation. A table of the two preferred methods and their mean and standard deviation is shown in Table 5.4.

Table 5.4: Mean and standard deviation of total detection performance.

Algorithm and feature set	$\mu_{DP}$ [%]	$\sigma_{DP}$ [%]	$\mu_{FPR}$ [%]	$\sigma_{FPR}$ [%]	$\mu_{FNR}$ [%]	$\sigma_{FNR}$ [%]
Linear SVM, feature set 2 (subset)	93.84	0.06	4.27	0.04	1.92	0.03
Ensemble, feature set 1 (full)	93.62	0.20	2.85	0.19	3.52	0.21

In the detection step it is more important to have a low false positive rate, otherwise data is added which contains no ML. The choice for preferred machine learning algorithm is therefore the ensemble of bagged trees. The linear SVM has a slightly increased overall detection performance and has lower standard deviations for all the parameters. It performs thus a bit more consistent. The feature set used for the ensemble is chosen towards the entire set of features, since the detection performance is higher, standard deviation is lower, while the increase in processing time is limited. The next step is to check how sensitive the chosen method is to the removal of features.

The ensemble of bagged trees with feature set 1 is retrained without the selected feature. The result is shown in Table 5.5 where the left column contains the feature which has been removed and the right columns contain the mean overall detection result, false positive and negative rates as well as the standard deviations. From Table 5.5 we can conclude that the mean detection accuracy, FPR and FNR are almost not influenced

Table 5.5: Sensitivity to removal of single features.

Removed feature	$\mu_{DP}$ [%]	$\sigma_{DP}$ [%]	$\mu_{FPR}$ [%]	$\sigma_{FPR}$ [%]	$\mu_{FNR}$ [%]	$\sigma_{FNR}$ [%]
max - mean $Z_{e,hh}$	93.64	0.22	2.86	0.17	3.49	0.22
max - mean $Z_{DR}$	93.54	0.21	2.75	0.17	3.71	0.20
max - mean $Z_{e,hv}$	93.22	0.23	3.31	0.30	3.47	0.30
max - mean $\rho_{hv}$	93.67	0.21	2.84	0.16	3.49	0.24
max - median $Z_{e,hh}$	93.61	0.21	2.83	0.16	3.56	0.23
max - median $Z_{DR}$	93.84	0.28	2.75	0.18	3.41	0.31
max - median $Z_{e,hv}$	93.89	0.28	3.34	0.28	2.77	0.28
max - median $\rho_{hv}$	93.68	0.26	2.85	0.16	3.47	0.23
Height diff $Z_{e,hh}-Z_{DR}$	93.67	0.19	2.82	0.18	3.51	0.20
Height diff $Z_{e,hh}-Z_{e,hv}$	93.60	0.23	2.93	0.17	3.46	0.25
Height diff $Z_{e,hh}-\rho_{hv}$	93.72	0.25	2.85	0.17	3.44	0.27
Height diff $Z_{DR}-Z_{e,hv}$	93.60	0.20	2.83	0.17	3.57	0.21
Height diff $Z_{DR}-\rho_{hv}$	93.67	0.20	2.82	0.16	3.51	0.22
Height diff $Z_{e,hv}-\rho_{hv}$	93.64	0.19	2.84	0.15	3.53	0.21
max - 800 m below max $Z_{e,hh}$	93.69	0.20	2.90	0.18	3.40	0.21
max - 800 m below max $Z_{DR}$	93.60	0.23	2.81	0.15	3.59	0.20
max - 800 m below max $Z_{e,hv}$	93.60	0.21	2.88	0.19	3.51	0.23
max - 800 m below max $\rho_{hv}$	93.77	0.17	2.76	0.16	3.47	0.21
Var( $Z_{e,hh}$ )	93.67	0.24	2.89	0.15	3.45	0.23
Var( $Z_{DR}$ )	93.70	0.20	2.84	0.16	3.47	0.24
Var( $Z_{e,hv}$ )	93.65	0.22	2.86	0.18	3.48	0.23
Var( $\rho_{hv}$ )	93.64	0.20	2.86	0.18	3.50	0.24

by removal of a single feature. This is expected since there are some redundant features in the feature set. The

only noticeable change occurs when  $max - mean$  of  $Z_{e,hv}$  is removed.

We can also check how important the individual features are within the ensemble of bagged trees. A histogram is shown in Figure 5.15 which indicates the estimated importance of the individual features. From the histogram we can conclude that especially the features which use  $Z_{e,hv}$  and  $Z_{DR}$  are important.

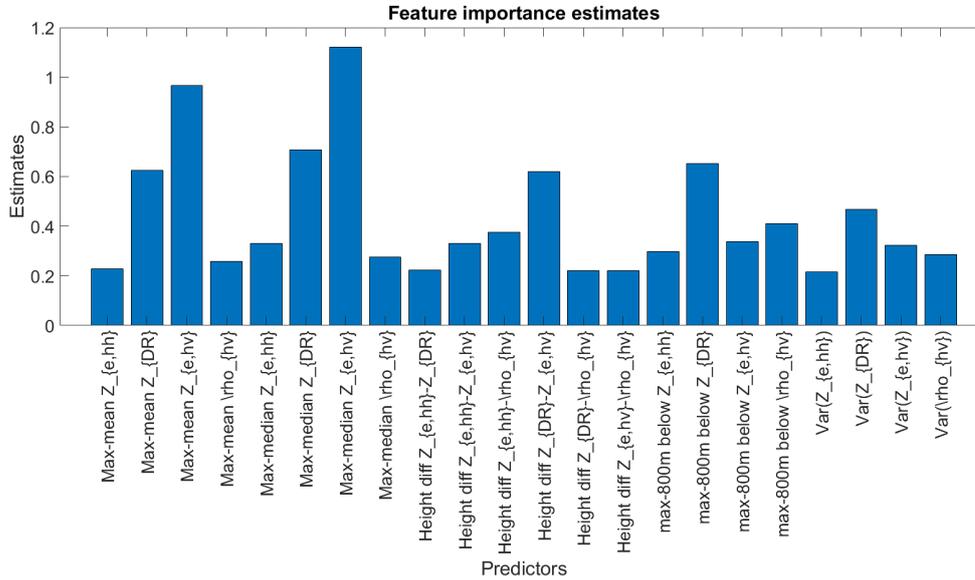


Figure 5.15: Estimate of feature importance.

Two special cases of removing features have also been investigated. The first is the removal of all features which use  $Z_{DR}$ . This would be the case if the radar was pointing at vertical orientation. The second is the removal of all features which are based on  $Z_{e,hv}$  since most radars are not able to make cross polar measurements. For these two cases, the model is retrained without the features which make use of  $Z_{DR}$  or  $Z_{e,hv}$  information. The result is shown in Table 5.6. Removal of  $Z_{DR}$  causes a drop of 1.5% in overall accuracy

Table 5.6: Sensitivity to removal of  $Z_{DR}$  or  $Z_{e,hv}$ .

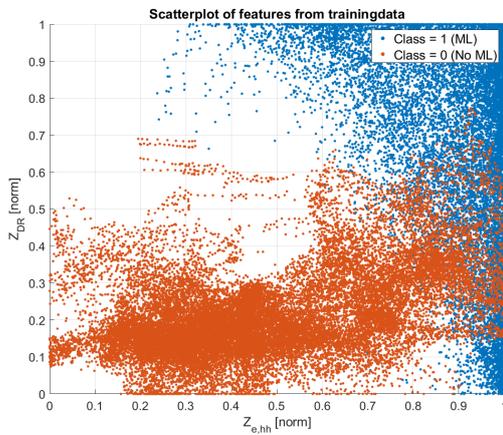
Algorithm and feature set	$\mu_{DP}$ [%]	$\sigma_{DP}$ [%]	$\mu_{FPR}$ [%]	$\sigma_{FPR}$ [%]	$\mu_{FNR}$ [%]	$\sigma_{FNR}$ [%]
Ensemble without $Z_{DR}$	92.14	0.31	2.68	0.17	5.18	0.34
Ensemble without $Z_{e,hv}$	92.60	0.27	4.38	0.15	3.02	0.31

on detection, which is mainly caused by an increase in false negatives. It misses about 1.7% more vertical profiles which do contain a ML signature. The removal of  $Z_{e,hv}$  results in a 1% lower overall accuracy on detection. Mainly the false positive rate has increased. Concluding on these two special cases, it can be said that for both radar observables the detection probability is lowered, but not drastically. The next step in the characterisation process is to select those cells which are part of the ML.

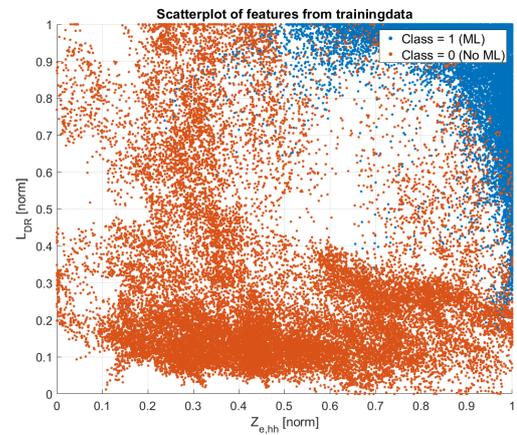
### 5.3.2. Machine learning ML attributer

First a similar feature analysis is carried out on the training dataset. This time it is expected that more overlap between individual features is present due to the problems related to the use of the normalised cell values (ML signature of different features do not entirely overlap). A total of 10 scatter plots can be created using the 5 available features. Again, only interesting scatter plots will be shown. Due to the normalisation procedure it would be ideal to have the ML data in the upper right corner of the scatter plots and the non-ML data in the lower left corner. In Figure 5.16a it is shown that this is not always the case for  $Z_{e,hh}$  and  $Z_{DR}$ . The division between corners is more visible in Figure 5.16b where  $Z_{e,hh}$  and  $L_{DR}$  are shown. In Figure 5.16c one can notice that  $Z_{DR}$  has also some data which belongs to the ML with values close to zero. Luckily, the  $L_{DR}$

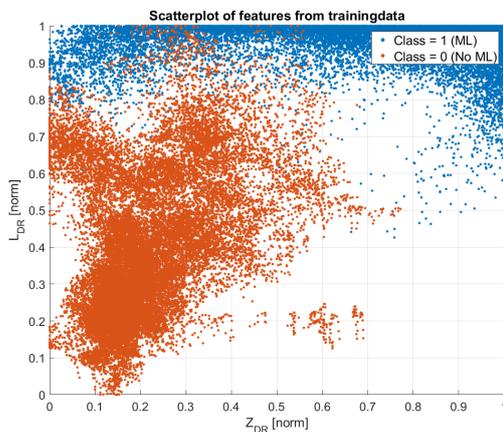
is still able to separate the two classes. In the last image of Figure 5.16 we see a different image because the  $\overline{\rho_{hv}}$  is also normalised between zero and one. The ML is mostly indicated by low  $\overline{\rho_{hv}}$  values, and as a result the scatter plot is mirrored compared to the other scatter plots. The  $\overline{Z_{e,hh}}$  versus  $\overline{\rho_{hv}}$  is able to provide some separation between the classes. Based on the scatter plots we can conclude that perfect separation using the



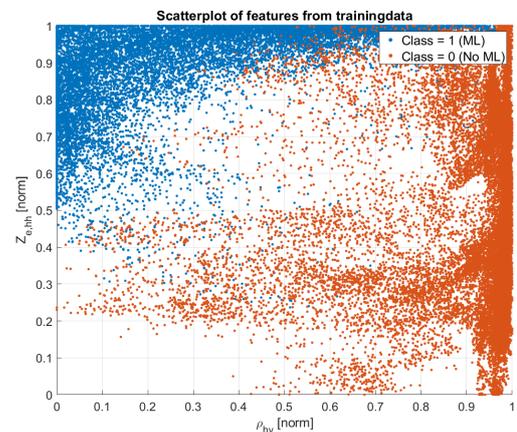
(a) Scatter plot of the normalised  $Z_{e,hh}$  on the x axis and  $Z_{DR}$  on the y axis.



(b) Scatter plot of the normalised  $Z_{e,hh}$  on the x axis and  $L_{DR}$  on the y axis.



(c) Scatter plot of the normalised  $Z_{DR}$  on the x axis and  $L_{DR}$  on the y axis.



(d) Scatter plot of the normalised  $\rho_{hv}$  on the x axis and  $Z_{e,hh}$  on the y axis.

Figure 5.16: Examples of attributer feature sets.

proposed features will probably not be achievable and as a result all proposed features will be used. Following the feature analysis, the selection of machine learning algorithm for attribution is carried out.

This second step in the ML characterisation procedure is attribution based on a machine learned model. Two machine learning algorithms are used for performance comparison. The first algorithm is based on a KNN algorithm, while the second is the already familiar ensemble of bagged trees. MATLAB offers a few different KNN algorithms or settings, and thus the results of the following algorithms will be shown:

- KNN;
  - Fine KNN;
  - Medium KNN;
  - Coarse KNN;
  - Cosine KNN;
  - Cubic KNN;
  - Weighted KNN;
- Ensemble of bagged trees.

The overall attribution performance of the selected algorithms is shown in Table 5.7. From the presented

Table 5.7: Mean and standard deviation of attribution performance.

Algorithm	$\mu_{DP}$ [%]	$\sigma_{DP}$ [%]	$\mu_{FPR}$ [%]	$\sigma_{FPR}$ [%]	$\mu_{FNR}$ [%]	$\sigma_{FNR}$ [%]
KNN Fine	98.18	0	1.64	0	0.17	0
KNN Medium	98.38	0	1.43	0	0.18	0
KNN Coarse	98.56	0	1.26	0	0.18	0
KNN Cosine	98.35	0	1.47	0	0.18	0
KNN Cubic	98.37	0	1.45	0	0.18	0
KNN Weighted	98.31	0	1.52	0	0.17	0
Ensemble of trees	98.30	0.02	1.50	0.02	0.19	0.01

table it can be seen that the overall accuracy on attribution of all algorithms is very close together and most of the KNN algorithms outperform the ensemble of bagged trees by a very small percentage. Since the KNN algorithms are very close together, a closer look is taken to the top three performing algorithms. One can also notice that the standard deviation is zero for all KNN algorithms. This is inherently to the KNN algorithm. It does not actually train a model, so no randomness is used in the training phase. Results will however, depend on the size and choice of training dataset. The confusion matrices of the top three KNN algorithms are shown in Figure 5.17a - Figure 5.17c. Again, a zero indicates the non-ML class and a one indicates the ML class. All KNN algorithms result in a high overall accuracy on detection, while still there is a slight difference. The cubic KNN and medium KNN perform almost the same, no significant difference can be seen here. The coarse KNN algorithm on the other hand performs significantly better in terms of false positives. This gives the idea that the increased number of neighbours has a positive effect on the number of false positives. Therefore, the number of required neighbours is increased to 150. The confusion matrix of this algorithm is shown in Figure 5.17d. No significant increase is demonstrated by further increasing the number of neighbours. It merely costed more memory and more processing time to run the KNN 150 algorithm, while the performance gain is negligible. Therefore, it is chosen for the remainder of the thesis to use the coarse KNN algorithm. In all cases, there is a large number of false positives, almost 40% of the detected ML cells is a false positive. The ML characterisation method relies on just two or three features with a high value. Unfortunately, this is not only related to the ML and as a result false positives occur. This is an issue which is not easily solved. It would help to add features which are able to indicate the ML or are able indicate cells which are not part of the ML. One could think to add Doppler information or the range dependent SNR as features. The result of the false positives will become visible when the detector and attributer are combined in the next section. The next step is to verify the sensitivity to missing features on the overall attribution performance.

To perform this sensitivity analyses, the coarse KNN algorithm is re-trained without the selected feature(s). The result is shown in Table 5.8. In the presented table each row shows the results when one feature has been removed. The bar above the features indicate that the feature has been linearly normalised from zero to one. One special case is also investigated, which is the removal of the entire  $Z_{e,hv}$  measurement. This results in the loss of the  $Z_{e,hv}$  and  $L_{DR}$  features. It can be seen that for most features the overall detection performance drops very slightly compared to the coarse KNN with the full feature set (Table 5.7). Interestingly this does not appear to be the case when  $\overline{Z_{DR}}$  is removed, the overall accuracy on attribution even increases. The number of false positives is significantly lower, while the false negatives almost doubled. The increased accuracy on attribution is a very useful property because it means that the performance is not critically dependant on the  $Z_{DR}$  and as a consequence the  $Z_{DR}$  can be missed. This is the case when the radar is pointing with an elevation angle of 90°. Removal of  $\overline{L_{DR}}$ ,  $\overline{Z_{e,hv}}$ ,  $\overline{\rho_{hv}}$  or both  $\overline{L_{DR}}$  and  $\overline{Z_{e,hv}}$  show almost no change in

Table 5.8: Sensitivity to removal of single features.

Removed feature	Overall $DP$ [%]	FPR [%]	FNR [%]
$\overline{Z_{e,hv}}$	98.46	1.37	0.16
$\overline{Z_{DR}}$	98.73	0.91	0.35
$\overline{L_{DR}}$	98.54	1.28	0.17
$\overline{Z_{e,hv}}$	98.51	1.26	0.22
$\overline{\rho_{hv}}$	98.54	1.27	0.18
$\overline{Z_{e,hv}}$ and $\overline{L_{DR}}$	98.51	1.24	0.23

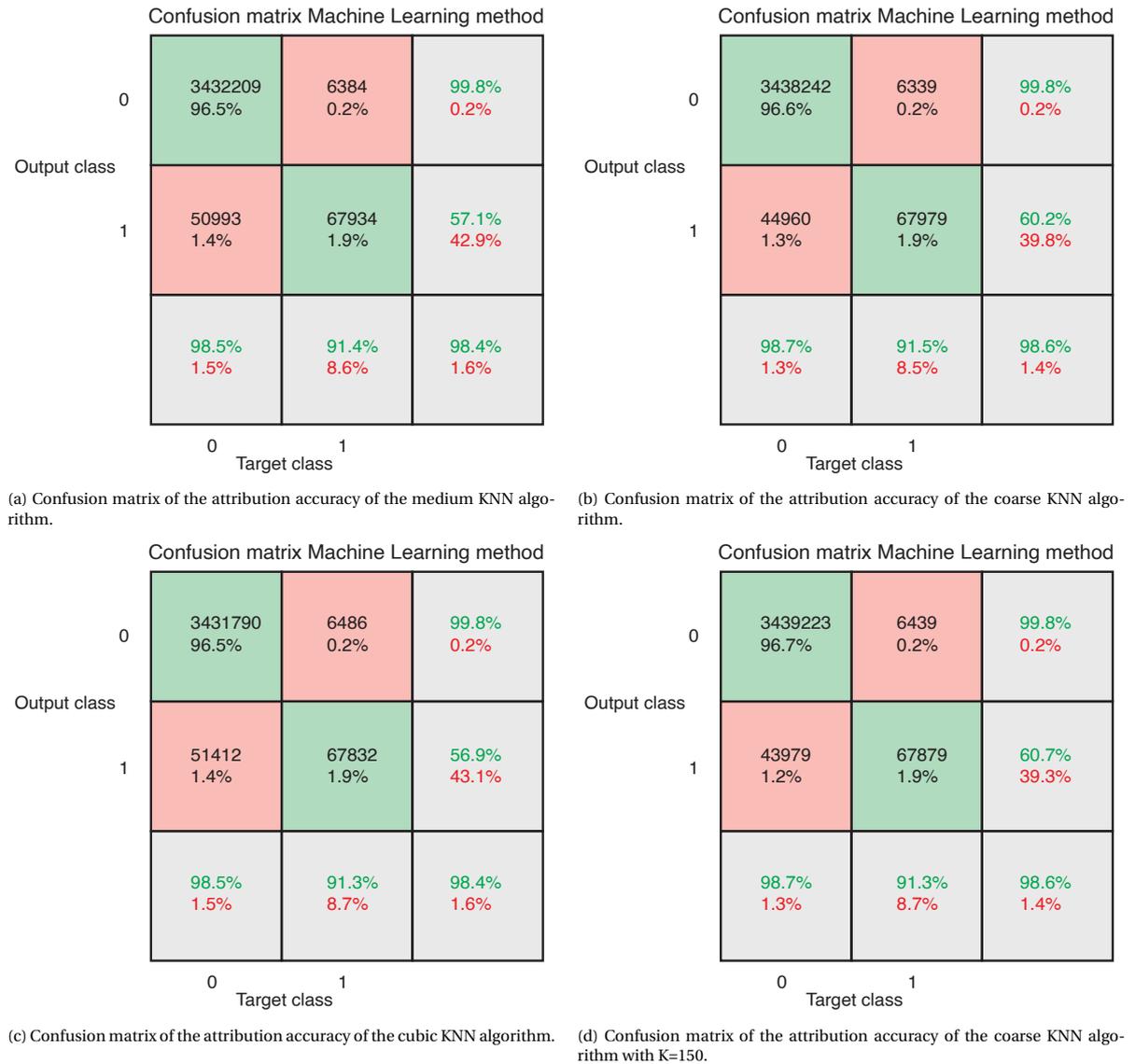


Figure 5.17: The attribution accuracy of the best three performing KNN algorithms and the extended coarse KNN algorithm with K=150.

performance. Removal of  $\overline{Z_{e,hh}}$  has as a consequence that the overall accuracy on attribution drops and the false positive rate increases. It makes sense that the performance is not critically dependent on one or two features, since the ML characteristic is significant in all of the selected features. If more than two features are removed, the attribution performance drops significantly and especially the false positive rate increases.

Unfortunately for KNN algorithms it is not possible to determine the feature importance. To give an indication of feature importance the ensemble of bagged trees is used. The result is shown in Figure 5.18. From the histogram it can be concluded that the ensemble of bagged trees is relying most on the cross polar reflectivity. In the next section the detector and attributer will be combined. First attention is given to the raw output of the combined characterisation procedure and secondly attention is given to the output after post-processing the raw output.

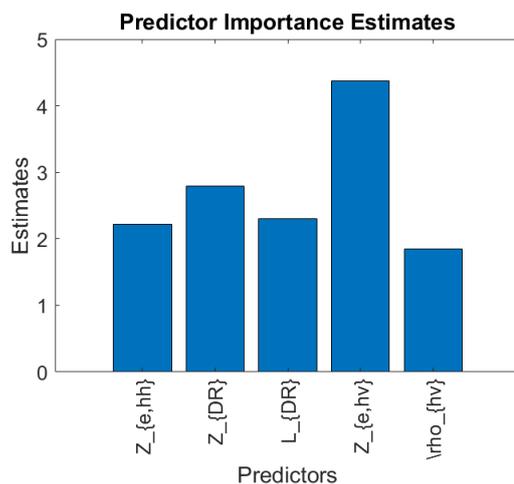


Figure 5.18: Estimate of feature importance based on the ensemble of bagged trees.

### 5.3.3. Combined evaluation of machine learning ML detector and attributer

The machine learning characterisation method is now complete and consists of a detector based on the ensemble of bagged trees with all proposed features and an attributer based on the coarse KNN algorithm with all proposed features. The output of the detector is used to pre-select the vertical profiles which contain a ML. The attributer is applied to the profiles which have been selected by the detector. The raw output of the attributer is shown in Figure 5.19a. Keep in mind that this image is representing the entire test dataset and is thus 17414 profiles long. In the image it can be seen that the raw output is also showing attribution in places where no ML is present. A zoomed example of a specific part is shown in Figure 5.19b. There, in more detail, one can see that false positives are present on a part where no ML is present. The ML characterisation method relies on high values of two or more radar observables. However, this condition is not only related to the ML, which leads to false positives. This is shown for the area within the rectangle of Figure 5.19b in Figure 5.20b to Figure 5.20d. The reason why the attributer has not detected a ML around the VPR number 8120 and an altitude of 4900 m while values of normalised  $Z_{e,hh}$ ,  $Z_{DR}$  and  $Z_{e,hv}$  are high is because the detector in the first place did not detect a ML in those vertical profiles. The attributer was thus never applied to those profiles. In this case the added value of having the two-step procedure becomes clear. The opposite also occurs, if the detector had false positives as output, the attributer is applied to profiles which do not contain a ML and as a result, the attributer might detect ML cells on places where no ML is present. One example is shown in Figure 5.21. The cause of the detection of the ML within the rectangle starts with the detector which outputs false positives. As a result, the profiles are normalised and the attributer finds ML cells.

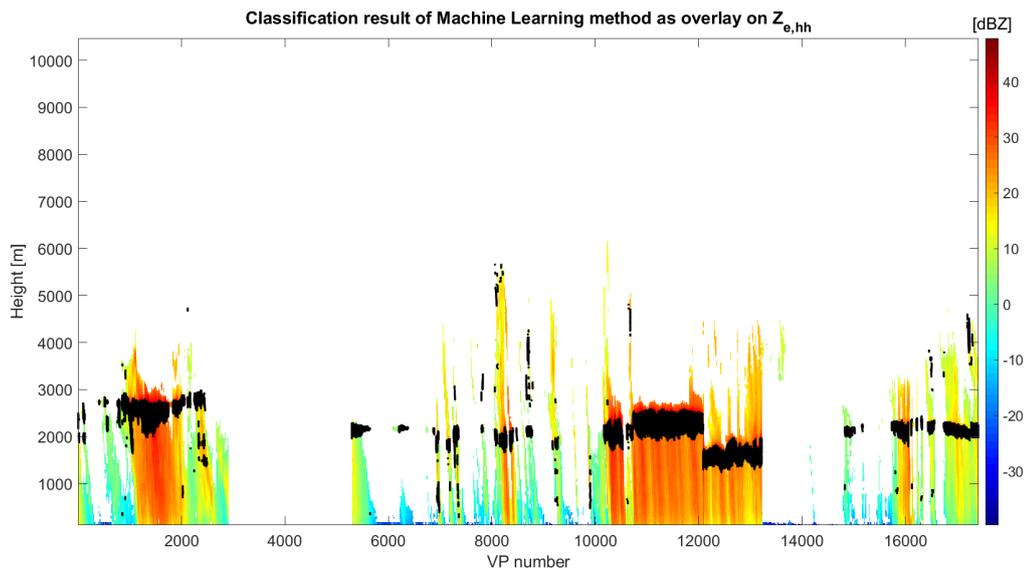
The relatively high number of false positives above the ML might be significantly reduced by introducing the Doppler related radar observables. If Figure 3.10a, Figure 3.12a and Figure 3.14a are recalled, one can see that a large difference exists between above and below the ML. As a result, these observables are capable of indicating when a cell is not part of the ML. This property is very useful in reducing the number of false positives.

Now, let's take a look at a situation with a clear ML scenario and see how well the boundaries of the ML are followed. A situation like this is shown in Figure 5.22. As can be seen there, especially the lower boundary is very well followed, while the top boundary of the ML misses the details. Remember that still the centre of the ML is the correct output at this stage. Post processing needs to stretch the result.

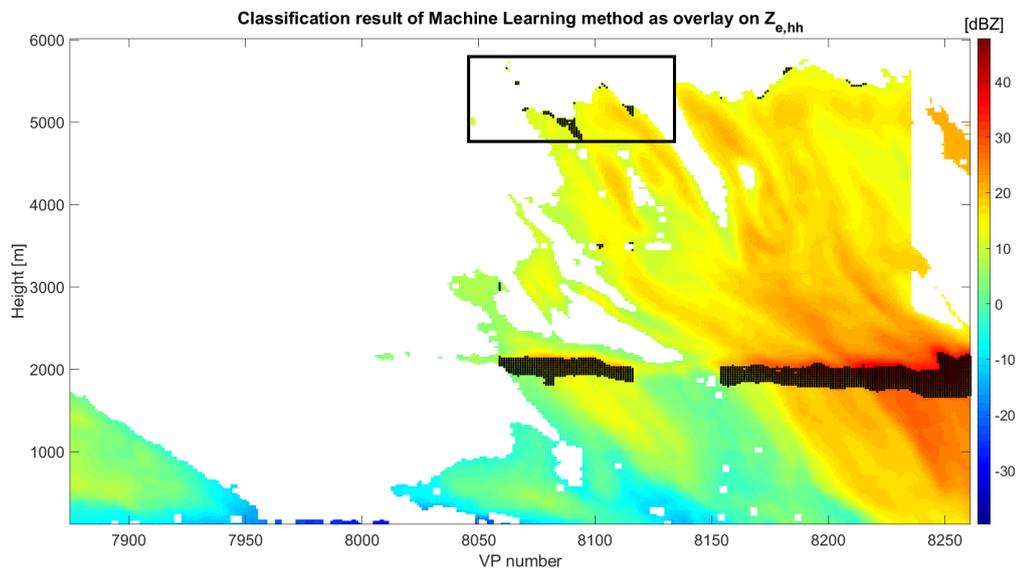
The raw output as shown in Figure 5.19a is not yet very usable. It contains a fair number of false positives and the ML boundaries do not coincide with the true boundaries. Therefore, the post-processing steps proposed in section 4.4.2 are implemented. In summary the steps are:

- Remove detections which are shorter than 1.5 minutes long and three cells thick;
- Stretch detections with 10 range resolution cells.

When these two steps are applied to the raw output the result is given in Figure 5.23. One can already see that compared to Figure 5.19a, the result significantly improved. When the true ML boundaries are added, a proper comparison can be made. The same part of the test data as used in Figure 5.22 is now shown in



(a) Example of the raw output of the machine learning characterisation method. The output of the attribution process is shown in black as an overlay on the  $Z_{e,hh}$ .



(b) Zoomed part of the raw output of the attribution process. False positives are created on the top part of the data. The data within the rectangle is shown in more detail in Figure 5.20a - Figure 5.20d.

Figure 5.19: Raw output of the attributer of the test dataset. Figure 5.19a shows the entire dataset, while Figure 5.19b shows a specific part in more detail to show where false positives occur.

Figure 5.24. The post processed data comes close to the true boundaries, but as a result of the post processing steps some details of the boundaries are lost. Another result of post processing is that parts of previously detected ML are now removed due to the minimum duration of a ML. This can be seen in Figure 5.25a. The blue lines still indicate the true boundaries of the ML, but no black dots representing output of the attributer are present. In this case the duration of the ML was shorter than 1.5 minute and as a result it has been removed. A disadvantage of the vertical stretch procedure is that the detected ML automatically becomes at least 21 resolution cells in altitude (1 remaining detected ML cell  $+2 \times 10$  from the stretching). This is definitely not always the case as is shown in Figure 5.25b. A very thin ML is present there, while the post processed result is much thicker compared to the true boundaries. One can also notice that the imposed SNR threshold of 15 dB might remove ML data of very weak MLs. Advantages of the post processing method lie mostly in the removal of false positives as can be seen in Figure 5.25c. This data is the same part of the test data as shown

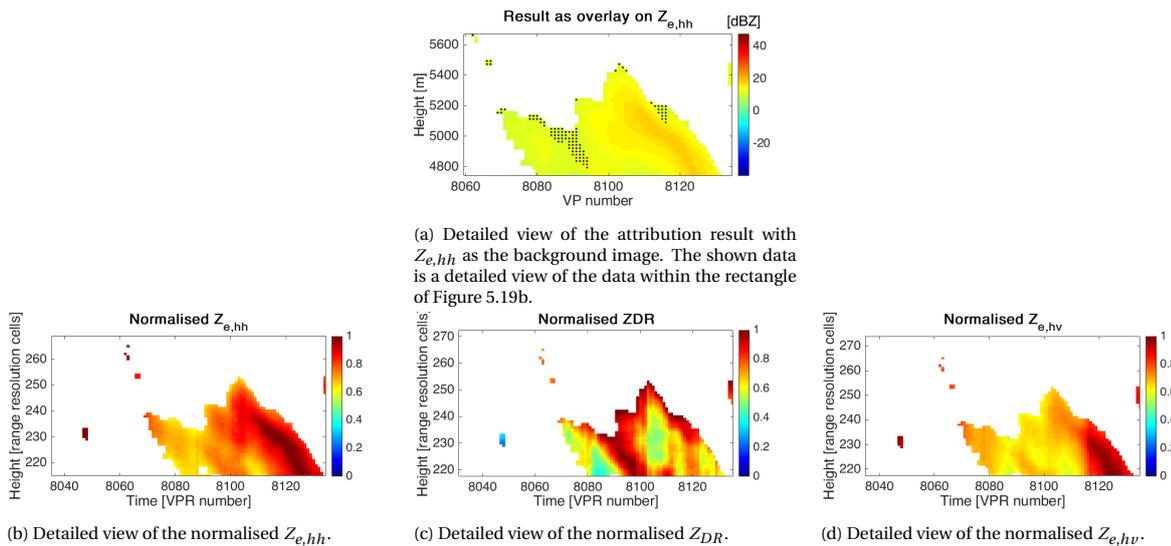


Figure 5.20: Detailed images of  $Z_{e,hh}$ ,  $Z_{DR}$  and  $Z_{e,hv}$  within the rectangle of Figure 5.19b. False positives shown in Figure 5.19b are the result of high values of these three radar observables.

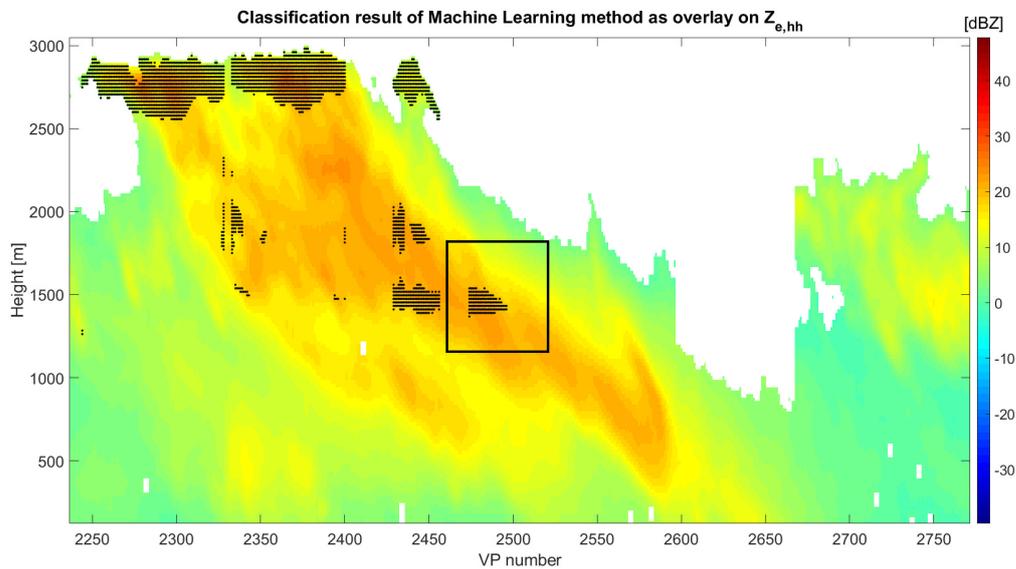


Figure 5.21: Example of the raw output of the machine learning ML characterisation method. Within the rectangle a false detection is made and the attributer also selected ML cells.

in Figure 5.21. No false positives are shown in this case. Overall, very few false positives remain after the post processing steps. One example is shown in Figure 5.25d. This false positive patch of output originated as a false positive by the detector. The profiles got normalised and the attributer positively identified the shown cells as ML cells. The result after post-processing is thus not perfect, but comes very close to the true boundaries of the ML. Now let's see how well the boundaries fit.

The final result of the machine learned method of characterising the ML resulted in the following errors with respect to the true ML boundaries. The mean error shows that the upper boundary is very close to the true boundary, while the lower boundary is slightly lower than the true boundary. Remember that one range resolution cell is about 21 m in altitude which means that the mean result of the upper boundary is within one resolution cell and the lower boundary is about three resolution cells. The fact that the RMSE is significantly higher than the mean error is showing that individual errors made are larger, but are positive and negative, while in the mean they have been averaged out. The correlation between the upper characterised boundary and the true boundary is lower than the correlation between the lower characterised boundary and the true

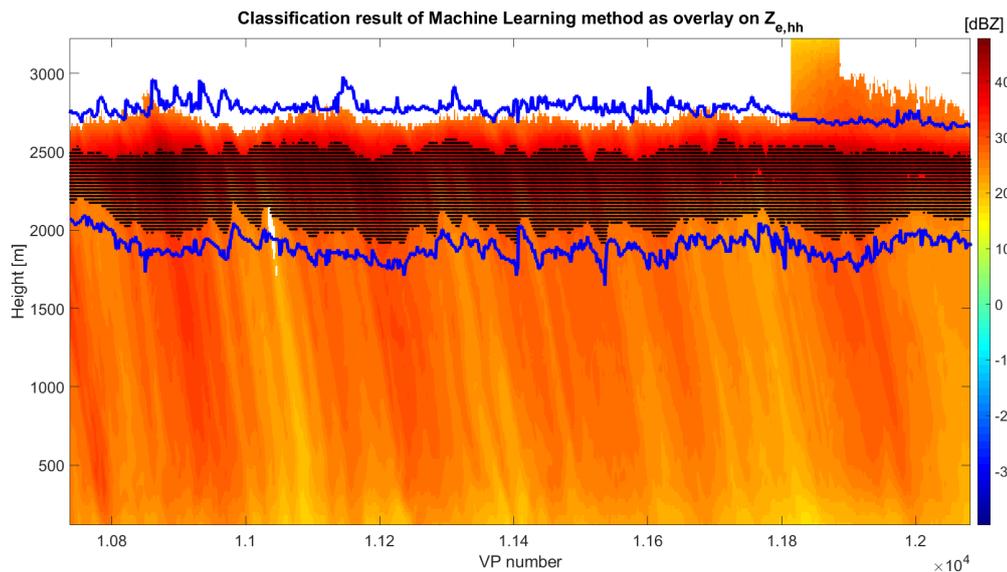


Figure 5.22: Detailed part of the test dataset with a clear ML. The black dots indicate the raw output of the machine learning method and the blue lines indicate the ML boundaries according to the ML definition.

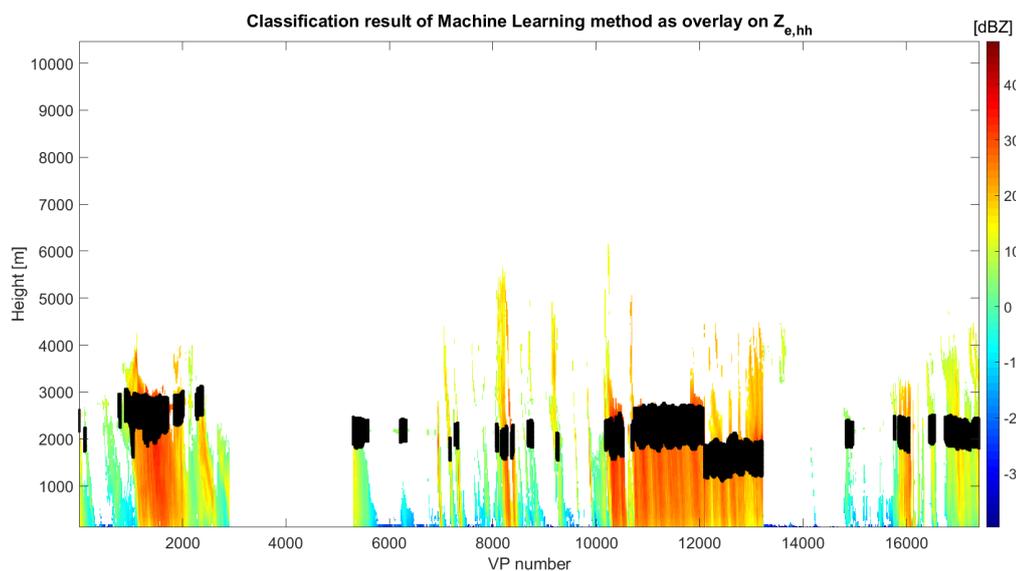


Figure 5.23: Result of post processed machine learned ML characterisation method.

boundary. Still both boundaries have a high correlation.

When the reference method is compared to the machine learned method it can be noticed that the detection probability is significantly higher (80.2% vs. 93.6%). The reference method is more conservative in the detection of a ML and as a result shows no false positives. The machine learned method on the other hand introduces false positives (2.85%). When it comes to the reconstruction of the ML boundaries, the reference method lower boundary follows the shape of the true boundary very well, but it fails to capture the full thickness of the ML. This is shown in RMSe and mean errors of the reference method. The upper boundary has a mean offset of 365 m below the true boundary and the lower boundary has a mean offset of 137 m above the true boundary. The machine learned method on the other hand shows a mean offset of the upper boundary of 20 m above the true boundary and the mean lower boundary is positioned 66 m below the true boundary. The machine learned method has a larger deviation between the RMSe and mean error which indicates

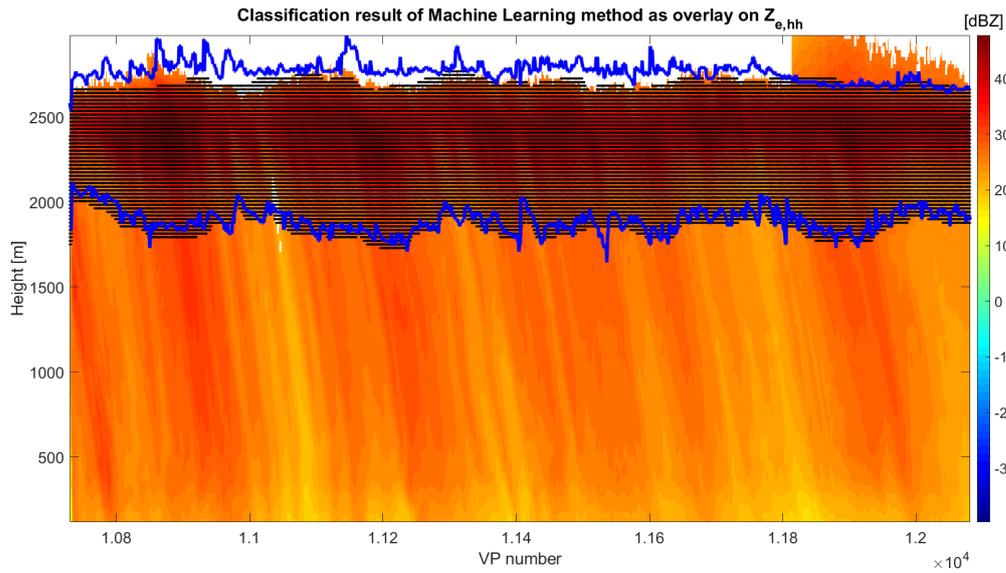
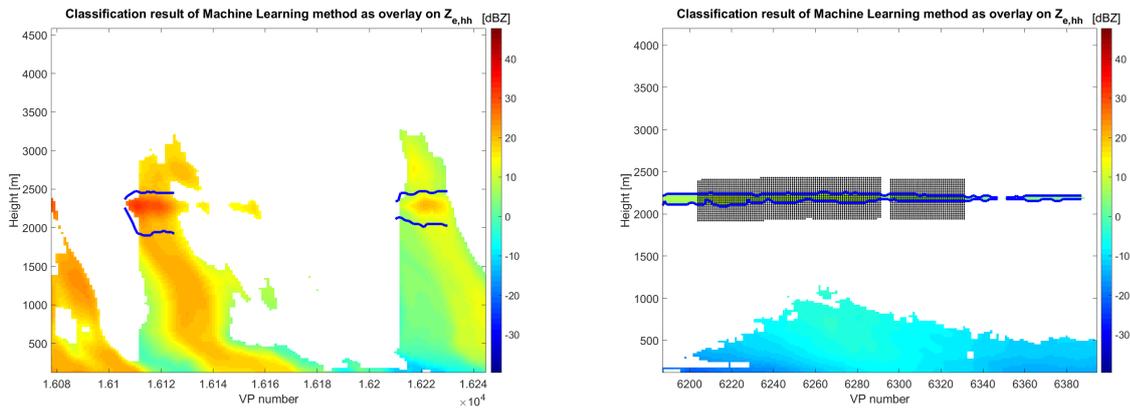
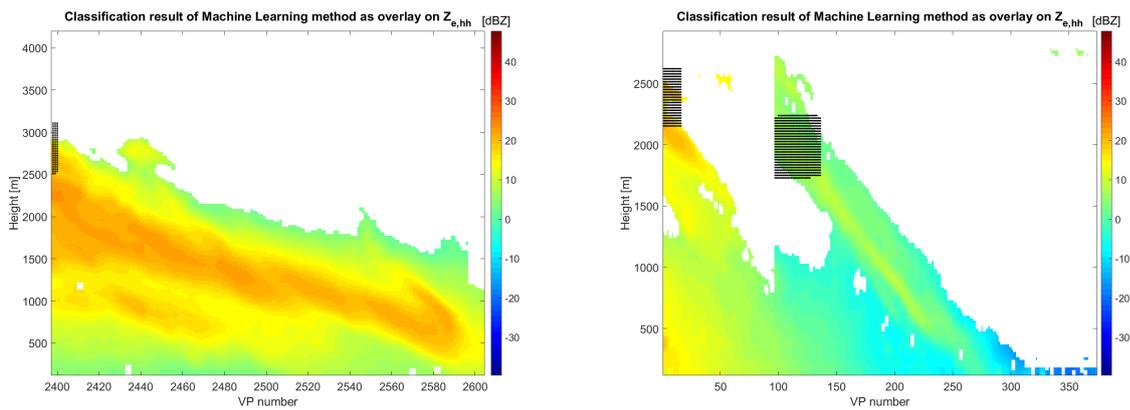


Figure 5.24: Detailed view of the post processed output of the attributer of the same data as shown in Figure 5.22. The detected boundaries follow the true boundaries very well, but some detail is lost due to the image erosion and dilation procedures in post processing.



(a) Parts of previously attributed ML are now removed as a result of the minimum duration of a ML.

(b) Part of the test dataset with a thin ML. Due to the vertical stretch procedure, a minimum thickness is imposed on the output of the attributer.



(c) Same part of the dataset as shown in Figure 5.21. No False positives are seen in this part after post processing.

(d) A few patches of false positives remain after post processing.

Figure 5.25: Detailed images of parts of post processed output of the machine learned ML characterisation method.

Table 5.9: Errors w.r.t. the true ML boundaries.

Boundary	RMSe [m]	Mean error [m]	Correlation
Upper	87	20	0.9656
Lower	95	-66	0.9758

that the machine learned method is less consistent and individual errors are larger, but still not as large as the reference method. When it comes to the correlation of the ML boundaries, the upper boundary for both methods is almost the same, while the lower boundary of the reference method shows a small improvement over the machine learned method. The overall performance of the machine learned method shows improvement over the reference method, it captures almost the entire thickness of the ML, the boundaries are closer to the definition and it is able to select more ML data.

# 6

## Conclusion and discussion

### 6.1. Conclusions

The ML shows itself in radar products as a non-linear increase (or decrease in case of  $\rho_{hv}$ ). This so-called bright band is an indicator of the presence of a ML, while the ML itself is an important indicator of the presence of stratiform precipitation. Stratiform precipitation is responsible for most of the precipitation in the areas around the equator and about 50% in the mid latitudes. Analysis of the ML is therefore deemed important to increase understanding of the ML and the complex physical processes within the ML. The purpose of this thesis is therefore to develop an algorithm which is able to detect and characterise the ML boundaries.

The first part of the thesis started with a literature study on radar observables and their use in ML detection and characterisation algorithms. The first papers on ML detection used only the reflectivity, while later publications used also Doppler information. The most recent papers almost all reside toward polarimetric observables. Most developed methods use between one and three observables and are therefore very dependent on those observables. For researchers it is convenient to have a method or framework which could support the use of as many available radar observables as possible. The method of Giangrande et al. (2008) is selected to form the basis for the reference method, while the method of Rico-Ramirez and Cluckie (2007) is selected to form the basis for the ML definition.

The next step was to examine the available data. First, a procedure is developed to acquire the mean and standard deviation of a typical ML scenario. This procedure is applied to all available radar observables for a radar elevation of 45° and 90°. After this procedure, we could conclude that for 45° elevation all available radar observables except the  $\Psi_{DP}$  are suitable to support a ML detection and characterisation algorithm. In addition to the  $\Psi_{DP}$ ,  $Z_{DR}$  loses the ability to support a ML detection and characterisation at 90° elevation. It is decided that the Doppler related observables are not used in this thesis because they show a higher variability.

Once the mean typical ML signatures were retrieved, they could be compared. Based on the defined reference heights, relative heights between the observables could be determined. Based on the relative heights the conclusion is drawn that the different radar observables are sensitive to different parts of the ML. As a result, a difficulty in ML detection based on multiple radar observables is introduced, because in extreme cases the signature of one observable might already be completed before the signature of the next observable presents itself. This also means that the thickness of the ML cannot be determined based on a single radar observable.

As a result of the presence of outliers in the upper atmospheric data, a method of biased data mitigation based on a SNR threshold of 10 dB has been proposed and later adapted to 15 dB. The imposed SNR threshold greatly reduced the number of outliers in the upper part of the atmospheric data. However, after applying the SNR threshold some outliers remained in lower altitudes. To discard those outliers an image processing technique based on image erosion and dilation is introduced. This technique removed most of the single cell outliers. After both procedures, the quality of the data was considered sufficient for (automatic) ML detection and characterisation.

In the final part of Chapter 3 a comparison between radar observables and the true 0° isotherm was performed to gain understanding of the differences between the two. A procedure was developed to compare the 0° isotherm with the radar observed ML. Unfortunately, only six instances were available to perform this comparison and as a result the conclusion is based on these measurements. The 0° isotherm appears to lie in

the upper transition between linear and non-linear return of the radar observables. This conclusion is further supported by Stewart et al. (1984).

Once knowledge about the data was present and deemed useful for automatic ML detection, a ML definition is developed which indicates the boundaries of the ML and is independent of the availability of specific radar observables. The knowledge about the relation between radar observables and  $0^\circ$  isotherm is also incorporated in the definition. To define the top ML boundary, the method by Rico-Ramirez and Cluckie (2007) is applied on the radar observable which shows the highest ML signature. For the lower boundary the same method is performed on the radar observable with the lowest ML signature.

Based on the literature study the Giangrande et al. (2008) technique is implemented as the reference ML detection and characterisation method. Adaptations to the proposed thresholds of the copolar correlation coefficient and equivalent reflectivity factor were necessary to make the method fit our data. Adaptations to the distance thresholds between the resolution cell under investigation and the maximum of  $Z_{e,hh}$  and  $Z_{DR}$  were necessary to increase the detected thickness of the ML. The reference method succeeded in detecting and characterising the ML. No false positives were reported, which is very desirable, but as a consequence the method generated a large number of false negatives, i.e., it missed ML data. The cause of the high number of false negatives is mostly due to the lower threshold on the  $Z_{e,hh}$  maximum (30 dBZ). Also, false negatives were reported as a result of clutter. The overall detection performance of the reference method was found to be 80.1%. When comparing the attributed boundaries with the true boundaries, the lower boundary of the reference method showed a very high correlation, while the upper boundaries were slightly lower correlated. Even though correlation of the lower detected ML boundary was high, still a mean error of 137 m above the true boundary remained. The mean upper boundary error is calculated to be -365 m. The reference method is thus successful in detecting the ML, but misses ML data and is not able to fully capture the thickness of the ML. This is not a huge problem, since the method only needs to set a baseline.

Next three image processing-based approaches were explored. The first approach is based on the Hough transform which is able to detect lines in images. The idea was to use the horizontal feature of the ML. The Hough transform was able to select horizontal features of the ML, but also detected a large number of horizontal features in other parts of the image. The Hough transform was therefore not a viable solution for a robust ML detection and characterisation method.

The second image processing method is based on the Radon transform which is also able to select lines in images and thus makes use of the horizontal feature of the ML. The method proved to be more robust against false detections, but it proved to be unsuccessful when the ML showed no good horizontal feature. The Radon transform was therefore not a viable solution for ML detection and characterisation.

The last image processing-based technique was based on the use of connected pixels. Pixels with very similar features are connected and form an area. In the study case the ML proved to be the largest area of connected pixels, but this was not true for other cases, especially if no ML was present. The results of this method were unpredictable and therefore not continued. Although the individual methods were not very successful in detecting the ML, a combination of the methods might work. If the Radon transform method could indicate a rough estimate of the ML height, the connected pixel method could be applied to that part of the image. The advantage of using the image processing techniques is that image processing methods do not rely on the actual data, and as a result these methods are not bound or associated to a specific radar.

The last and core method which is designed and developed within the thesis is based on a two-step machine learning ML detector and attributer. The first step is the detector, which detects if a ML exists within the vertical profile under consideration. The second step involves the determination of the cells which are part of the ML. Both steps use a machine learning algorithm to create a model of the data. Multiple detectors and attributers based on different machine learning algorithms have been analysed, while also multiple feature sets have been investigated. The best performing detector was found to be the ensemble of bagged trees because of the lower false positive rate. For the purpose of the thesis a low false positive rate is preferred above a low false negative rate, but preference may change depending on the use of the program. The overall detection performance of the ensemble of bagged trees is 93.6% against 93.8% of the linear SVM. The false positive rate is significantly lower for the ensemble of bagged trees, being: 2.9% against 4.3%. The full feature set is used for the ensemble of bagged trees, while the linear SVM used the predetermined subset of features. The detection performance of the machine learned method is reported to be significantly higher than the reference

method. A difference of 13.5% is shown. A downside of the machine learned method is the introduction of false positives.

The second step of the machine learned method was also tested with two different machine learning algorithms based on linearly normalised profile data. This step assumed a true positive detection from the detector. The best performing algorithm was selected on a marginal improvement of attribution performance. The coarse KNN algorithm performed 0.3% better, which was mainly due to a lower false positive rate. Overall attribution accuracy was found to be 98.6%. From all detected ML cells, almost 40% were false positives. Post processing was therefore necessary to remove the false positives and to increase the thickness of the characterised ML. The high number of false positives shown in the output of the attributer are mostly situated on the upper part of the atmosphere. When post processing is performed, almost all of the false positives are removed and the thickness of the characterised ML is stretched to fit the original thickness. The downside of the used post processing technique is a shown loss of details of the attributed boundaries and a minimum ML thickness is imposed. After post processing the attributed boundaries coincide very well with a mean error of 66 m below the lower boundary and 20 m above the upper boundary.

When the reference method is compared to the machine learned method, the machine learned method performed better on most of the parameters. It has a higher detection probability and approximates the true ML boundaries of the ML much better, while the extra processing time remained limited. The reference method on the other hand proved to have no false positives on our test dataset and showed a slightly higher correlation of the lower boundary. The proposed machine learned method shows to be capable of characterising the ML for ML analysis. Also shown is the methods capability of handling different orientations based on the removal of  $Z_{DR}$  features. It loses some detection performance, but it remains functional after re-training the algorithm. The same can be said about the removal of cross polar measurements. This shows that the developed method is generally applicable and flexible as long as the method is supplied with a proper set of training data and good features.

## 6.2. ML detection and characterisation program

The final result of this thesis was not only the development of the proposed methods, but also included the development of a program for automated ML detection and characterisation. A MATLAB GUI has been produced for a user-friendly interface. The program gives the user a list of functionalities:

- Load TARA pre-processed data;
  - Apply a user defined SNR threshold;
  - Apply a user defined moving average window in both horizontal (time) and vertical (height) directions;
  - Apply linear normalisation on the vertical profiles;
- Display the loaded data;
- Create a training dataset for a machine learning detector;
- Create a training dataset for a machine learning attributer;
- Train a user defined machine learning detection algorithm with the training dataset;
- Train a user defined machine learning attribution algorithm with the training dataset;
- Apply the machine learned models to a pre-defined test dataset;
- Apply the reference method to a pre-defined test dataset;
- Calculate the performance parameters of both machine learned method and reference method;
- Apply the machine learned method to a new dataset and show results;
- Apply the reference method to a new dataset and show results;
- Quickly apply pre-trained (result of this thesis) machine learned models to new dataset;

An image of the GUI is shown in Figure 6.1. All the functionalities can be started using the buttons, radio buttons and user editable fields. The program has not been hardened and as a result users are able to fill anything in the user editable fields and the program will give an error and stops functioning. A list of M-files required to run the program is added in Appendix E. Also, a manual is added to the thesis and can be found

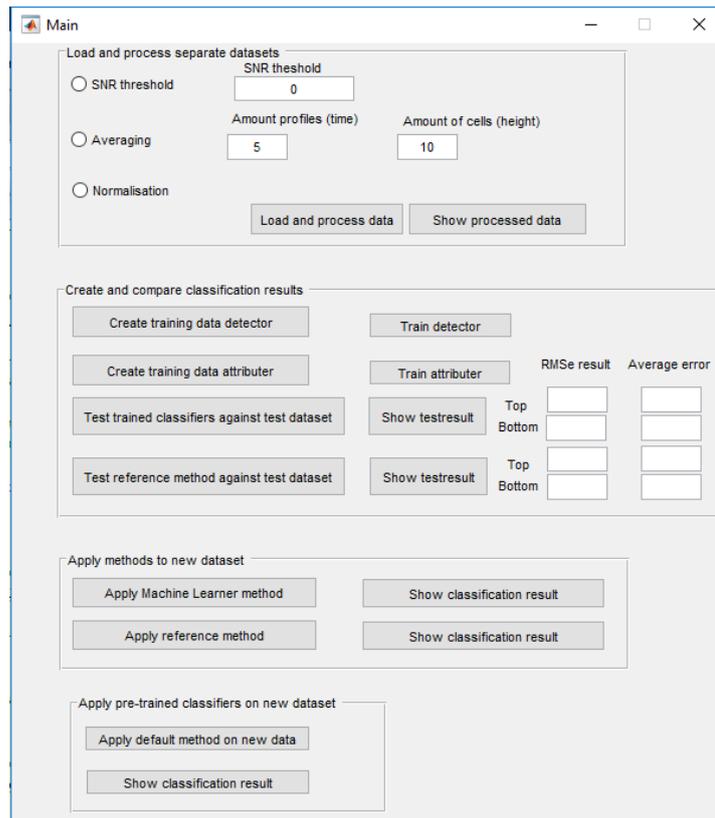


Figure 6.1: Example of the GUI.

in Appendix G. The manual will explain the basic operation of the tool and will explain how to create your own machine learned model and use that with the program.

### 6.3. Recommendations for future work

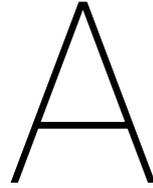
While the result of the proposed machine learning ML characterisation method is very positive, some recommendations for future work are proposed. In this thesis work  $L_{DR}$  is not used in the detection process due to the high  $L_{DR}$  levels in upper parts of the vertical profiles. When some method is introduced to circumvent the high  $L_{DR}$  values in the upper parts of the vertical profiles and  $L_{DR}$  is added to the feature set, it might aid in the detection performance. However, the expected increase in detection performance is limited, because  $L_{DR}$  is directly related to the measurement of  $Z_{e,hv}$ , which is used in the feature set.

Also, the detection process is only based on properties in vertical direction. Since the ML also reveals itself as a horizontal feature, it should improve the detection probability if some features of horizontal properties are added to the feature set of the detector or attributer. This might conflict if in the future the ML detector needs to run in real time, because horizontal features require data in the horizontal (time) plane. Doppler information is also not used, but might prove valuable in ML detection, this is shown in Chapter 3. One or more features based on Doppler information should be able to improve ML detection.

For the attribution part, two main issues arose. The output of the attributer was contaminated with a high number of false positives and the characterised ML required stretching in vertical direction as a result of the limits set on the ML training data. Both issues are handled using post processing steps, but the used post processing steps introduced new issues. To reduce the necessity of the severe post processing steps, main attention should be paid to reduce the false positive output of the attributer. This can include, but is not limited to, adding features which include Doppler information as well as the range dependent SNR. Since there is a distinct difference in Doppler information above and below the ML, it might be the key to suppress the high number of false positives well above the ML. However, the quality of the Doppler measurement well above the ML might become an issue. Secondly, attention should be paid to the stretching method. If a different method can be found which is able to stretch the characterised ML in a smarter way, the imposed minimum thickness of the current method might become obsolete.

Although not been investigated thoroughly, the imposed SNR threshold might remove ML data. Since in this thesis a study case is used to determine the ML characteristic, it might be that there are situations possible where a ML is present below the SNR threshold. The ML in such a case is very weak and probably cannot produce significant rain quantities. If one would focus an analysis entirely on the ML, it might be that some data of a weak ML is removed due to the enforced SNR threshold. If the focus of a ML analysis is to study the relation between ML and the rain below the ML, this is probably not an issue since the weak ML is not producing any significant amounts of rain. A solution could be found in an adaptable SNR threshold programmed within the pre-processing steps.





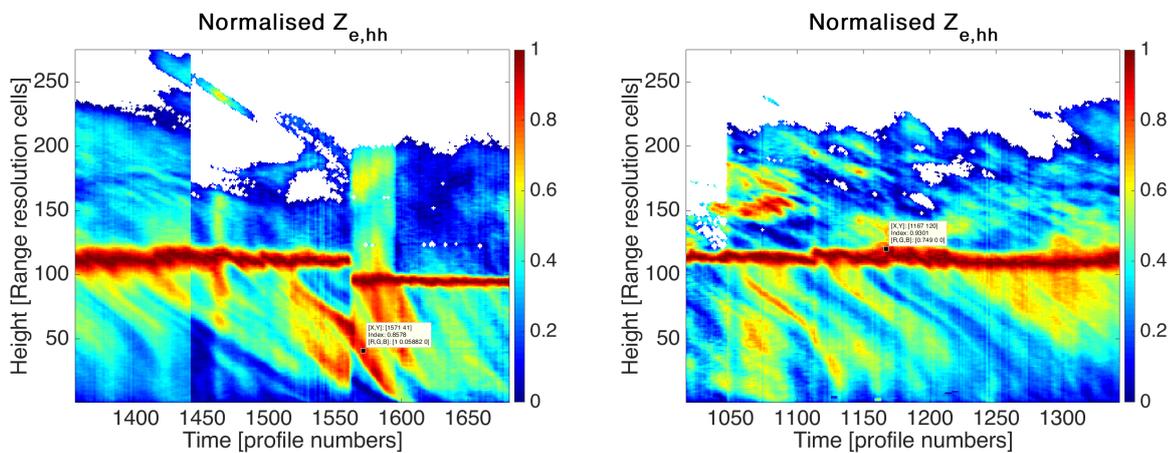
## Cells with similar features

Below is an example given of two resolution cells within one dataset with very similar features. The first row of Table A.1 contains a sample outside the ML, while the second row contains a sample with almost the same values, but is from within the ML. The only significant difference in this case is the  $\overline{L_{DR}}$ , which is slightly lower in case of the ML, while the opposite is expected. Images of both data cells from Table A.1 and their

Table A.1: Example of training data with similar features

Cell coordinate [column row]	$\overline{Z_{e,hh}}$	$\overline{Z_{DR}}$	$\overline{L_{DR}}$	$\overline{Z_{e,hv}}$	$\overline{\rho_{hv}}$	Class
1571 41	0.88	0.48	0.90	0.91	0.49	0
1167 120	0.90	0.49	0.76	0.89	0.49	1

surroundings are shown in Figure A.1. It can be seen that indeed the resolution cell indicated in the first row of Table A.1 and shown in Figure A.1a is not part of a ML, while the second row and Figure A.1b are.



(a) Image of the data cell which is not part of the ML.

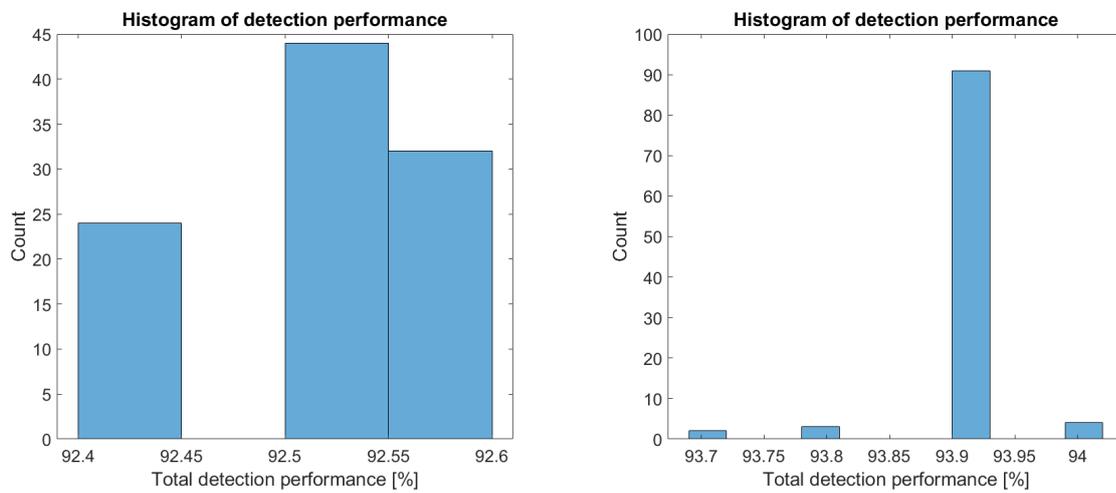
(b) Image of the data cell which is a part of the ML.

Figure A.1: Images show the data cells from Table A.1 and their surrounding data. Both have very similar features, but the left image is clearly not part of a ML while the right image is.



# B

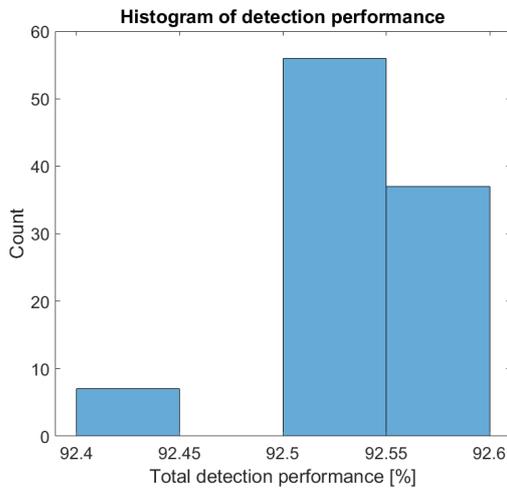
## Histograms SVM detectors



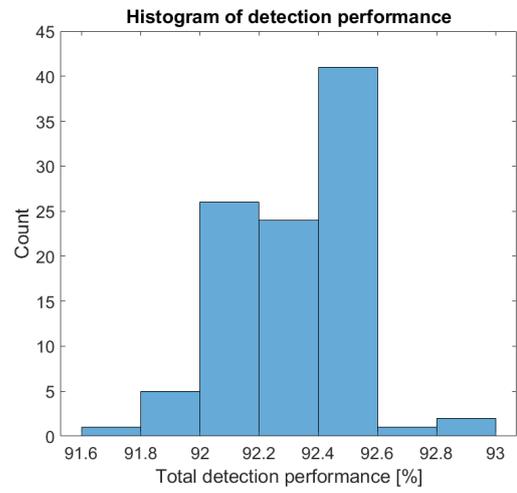
(a) Linear SVM, all features

(b) Linear SVM, subset of features

Figure B.1: Histograms of total detection performance of linear SVM detector with the full feature set (a) and subset of features (b).

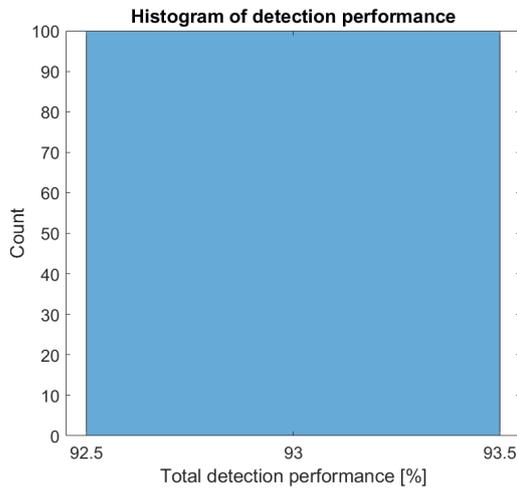


(a) Quadratic SVM, all features

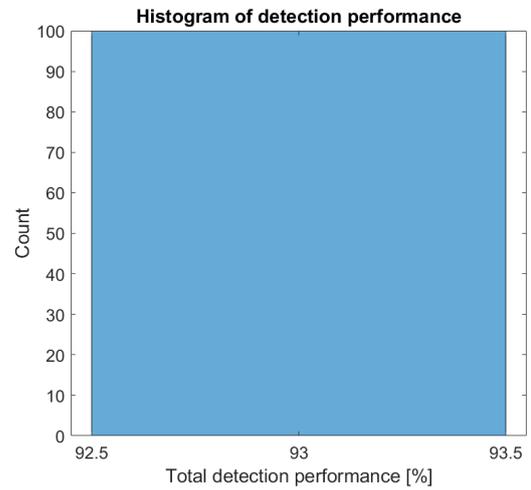


(b) Quadratic SVM, subset of features

Figure B.2: Histograms of total detection performance of Quadratic SVM detector with the full feature set (a) and subset of features (b).



(a) Coarse Gaussian SVM, all features

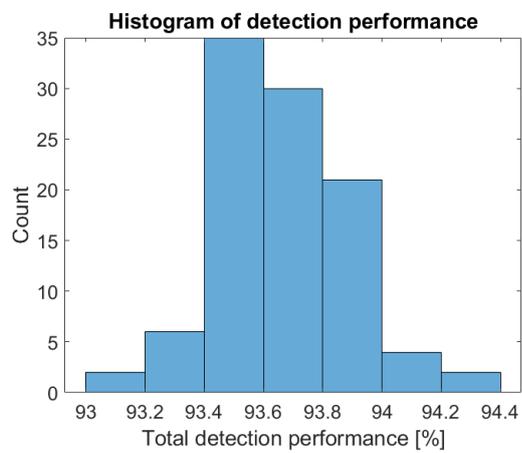


(b) Coarse Gaussian SVM, subset of features

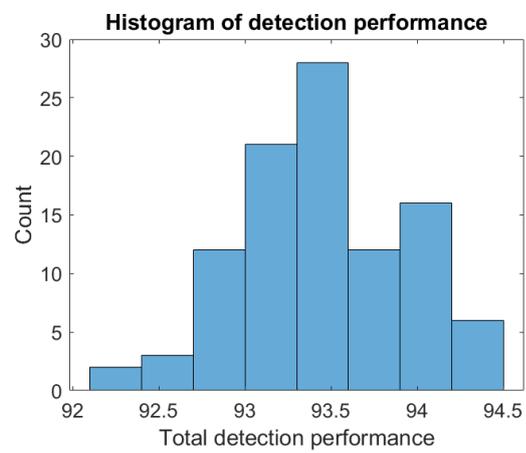
Figure B.3: Histograms of total detection performance of coarse Gaussian SVM detector with the full feature set (a) and subset of features (b).

# C

## Histograms ensemble detector



(a) Ensemble of bagged trees detector, all features



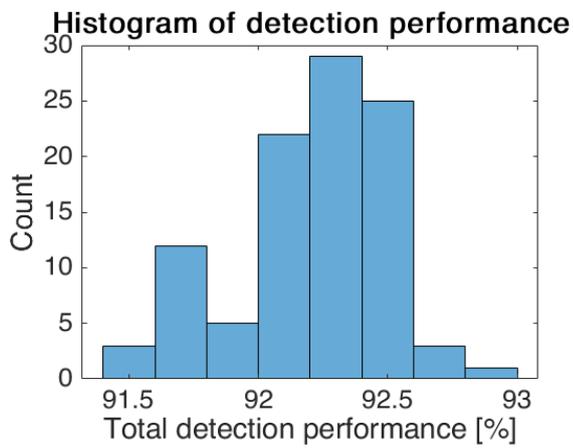
(b) Ensemble of bagged trees detector, subset of features

Figure C.1: Histograms of total detection performance of ensemble of bagged trees detector with the full feature set (a) and subset of features (b).

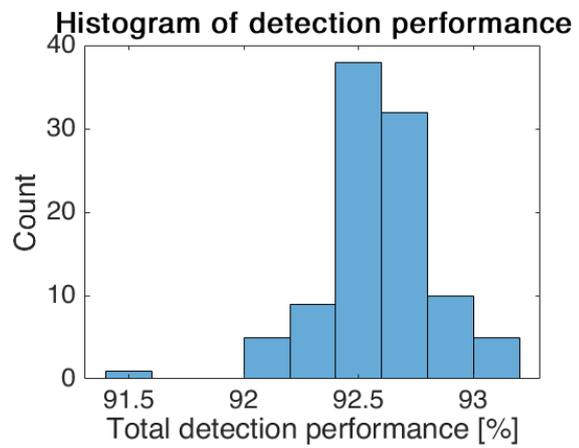




## Histograms special cases ensemble of bagged trees



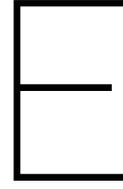
(a) Ensemble of bagged trees detector, all features which use  $Z_{DR}$  have been removed from the feature set.



(b) Ensemble of bagged trees detector, all features which use  $Z_{e,hv}$  have been removed from the feature set.

Figure D.1: Histograms of total detection performance of ensemble of bagged trees detector when on the left all features which use  $Z_{DR}$  are removed and on the right all features which use  $Z_{e,hv}$  are removed.





## List of files required for the GUI

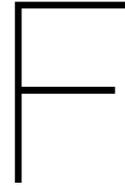
Table E.1: List of files required for the GUI.

Name	Function	Input	Output
Apply_ML_Class_model.m	Applies ML attributer to ML data	trained model, list of ML detections	BW image with attribution result
Apply_model.m	Applies the detector to dataset	Dataset, trained model	List of detections
Calculate_class_performance.m	Calculates the detection performance and boundary errors, plots confusion matrix of detection	list of ML detections, attributer output	RMSe top and bottom, mean error top and bottom
Calculate_reference_performance.m	Calculates performance of reference method, plots confusion matrix of detection	output of reference method	RMSe top and bottom, mean error top and bottom
Calculate_profile_features.m	Calculates the profile features for detector	None	Array with features
Create_train_data_ML_classifier.m	Generates training dataset for attributer	True upper and lower boundaries, distance from boundary to training dataset, toggle for ML or not	Array of training data for attribution learning
Load_data.m	Loads pre processed TARA .mat files	Path	Data structure
Main.m	Main program	None	GUI
ML_definition.m	Applies ML definition to given dataset	None (uses global Processed_data)	Upper and lower boundaries
Plot_bara_data.m	Plots $Z_{e,hh}$ , $Z_{e,hv}$ , $Z_{DR}$ , $L_{DR}$ and $\rho_{hv}$	None (uses global Processed_data)	Figures

Continued on next page

**Table E.1 – continued from previous page**

<b>Name</b>	<b>Function</b>	<b>Input</b>	<b>Output</b>
Process_data.m	Applies specified processing steps to loaded data	Averaging toggle, vertical window size, horizontal window size, SNR threshold toggle, SNR threshold value, normalisation toggle, dataset	Processed data structure
run_reference.m	Applies the reference method to given dataset	none (uses global Processed_data)	BW image of attributed ML cells
TrainClassifier_ebt_det_1.m	Trains the detector	Training data	Trained model
TrainClassifier_class_KNN_coarse.m	Trains attributer	Training data	Trained model
TrainedModel_class.mat	Pre-trained model attributer	Data, detections	attributions
TrainedModel_detector.mat	Pre-trained model detector	Data	detections



## List of files for generating thesis results

Table F.1: List of M-files to generate the results in this thesis.

Name	Function	Input	Output	Reference
Create_scatterplot	Generates scatterplots of training data	Training data, user defined features to compare	Scatter plot	section 4.4
Hough_test.m	Detects lines using Hough transform	Image of study case (defined in m-file)	Hough transform of BW image, detected lines	section 5.2.1
Image_clustering.m	Detects largest area of clustered pixels	Image of study case (defined in m-file)	Image of clustered pixel areas	section 5.2.3
import_radiosonde.m	Load radiosonde data from specified path	path (is asked to user)	structure of radiosonde data	section 3.5
import_RS_Cabauw.m	Load radiosonde data from path given by import_radiosonde.m	path from import_radiosonde.m	Structure RS with radiosonde data	section 3.5
Load_data.m	Loads .mat files with pre-processed TARA data	Path	arrays with data	N.A.
ML_analysis_LDR.m	Loads study case data and shows mean and std dev of $L_{DR}$ ML profile	None	Images	Figure 3.2.1
ML_analysis_LDR_full.m	Calculates mean and std dev of $L_{DR}$ ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_LDR_vert.m	Loads vertical study case data and shows mean and std dev of $L_{DR}$ ML profile	None	Images	section 3.2.2
ML_analysis_Psi.m	Loads study case data and shows mean and std dev of $\Psi_{DP}$ ML profile	None	Images	Figure 3.2.1

Continued on next page

Table F.1 – continued from previous page

Name	Function	Input	Output	reference
ML_analysis_Psi_full.m	Calculates mean and std dev of $\Psi_{DP}$ ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_Psi_vert.m	Loads vertical study case data and shows mean and std dev of $\Psi_{DP}$ ML profile	None	Images	section 3.2.2
ML_analysis_rho.m	Loads study case data and shows mean and std dev of $\rho_{hv}$ ML profile	None	Images	Figure 3.2.1
ML_analysis_rho_full.m	Calculates mean and std dev of $\rho_{hv}$ ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_rho_vert.m	Loads vertical study case data and shows mean and std dev of $\rho_{hv}$ ML profile	None	Images	section 3.2.2
ML_analysis_Skh.m	Loads study case data and shows mean and std dev of Doppler spectral skewness ML profile	None	Images	Figure 3.2.1
ML_analysis_Skh_full.m	Calculates mean and std dev of Doppler spectral skewness ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_Vz.m	Loads study case data and shows mean and std dev of derived vertical velocity ML profile	None	Images	Figure 3.2.1
ML_analysis_Vz_full.m	Calculates mean and std dev of vertical velocity ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_Wh.m	Loads study case data and shows mean and std dev of Doppler spectral width ML profile	None	Images	Figure 3.2.1
ML_analysis_Wh_full.m	Calculates mean and std dev of Doppler spectral width ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_ZDR.m	Loads study case data and shows mean and std dev of $Z_{DR}$ ML profile	None	Images	Figure 3.2.1
ML_analysis_ZDR_full.m	Calculates mean and std dev of $Z_{DR}$ ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_ZDR_vert.m	Loads vertical study case data and shows mean and std dev of $Z_{DR}$ ML profile	None	Images	section 3.2.2
ML_analysis_ZH.m	Loads study case data and shows mean and std dev of $Z_{e,hh}$ ML profile	None	Images	section 3.2.1

Continued on next page

Table E.1 – continued from previous page

Name	Function	Input	Output	reference
ML_analysis_ZH_full.m	Calculates mean and std dev of $Z_{e,hh}$ ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_ZH_vert.m	Loads vertical study case data and shows mean and std dev of $Z_{e,hh}$ ML profile	None	Images	section 3.2.2
ML_analysis_ZHV.m	Loads study case data and shows mean and std dev of $Z_{e,hv}$ ML profile	None	Images	Figure 3.2.1
ML_analysis_ZHV_full.m	Calculates mean and std dev of $Z_{e,hv}$ ML profile specifically for run_all	None	Images	section 3.3
ML_analysis_ZHV_vert.m	Loads vertical study case data and shows mean and std dev of $Z_{e,hv}$ ML profile	None	Images	section 3.2.2
ML_definition.m	applies the ML definition to a dataset	Pre-processed dataset (asked to user)	ML boundaries according to definition	section 4.1
Monte_carlo_class.m	Trains and applies trained attributer to test dataset and determine performance	Training dataset, test dataset, machine learning script, ground truth dataset	Performance parameters	section 5.3.2
Monte_carlo_detector.m	Trains and applies trained detector to test dataset and determine performance	Training dataset, test dataset, machine learning script, ground truth dataset	Performance parameters	section 5.3.1
Output_all.m	Plots all data from Processed_data structure	Processed_data structure	Images	N.A.
Process_data.m	Applies specified processing steps to loaded data	Averaging toggle, vertical window size, horizontal window size, SNR threshold toggle, SNR threshold value, normalisation toggle, dataset	Processed data structure	N.A.
Radon_test.m	Performs radon transform to specified image and performs line detection	Image of study case	Radon parameter space and detections	section 5.2.2

Continued on next page

**Table E.1 – continued from previous page**

<b>Name</b>	<b>Function</b>	<b>Input</b>	<b>Output</b>	<b>reference</b>
RS_profiles.m	Radiosonde comparison with radar observables	User defined radiosonde data, radar measured data around radiosonde launch	Images of radiosonde zero degree isotherm vs. radar measured data	section 3.5
run_all.m	Runs all ML study case analysis and extracts the mean profiles of all radar observables	Study case data	Image with all mean ML profiles	section 3.3



# User manual of GUI

**This manual is solely written for the use of the GUI supplied by the thesis. This manual is divided into three sections. The first section will describe how to load and process data. The second section will explain how to train different machine learning algorithm and how users can create and use their own machine learning algorithms with the GUI. Lastly a short description is given on how to apply default machine learned algorithms to new data and how to define a new default.**

## **G.1. Load and pre-process data**

The top part of the GUI (indicated by the black rectangle in Figure G.1) is meant to load, process and display the data. For pre-processing the user has a few options. The user can select to apply a SNR threshold (in dBZ), averaging with a user defined moving average window and to apply linear normalisation of the profile data. The corresponding radio button needs to be selected to apply the processing step. Any combination of pre-processing steps can be selected. Note that no fault check is performed on the input of the user. Once the pre-processing steps have been selected, click on "Load and process data". This initiates the load procedure. A window will pop up asking for the input data (shown in Figure G.2). Only pre-processed data from TARA can be loaded, but of course when the load process is adapted, any radar data should be able to be loaded. Note that multiple files may be selected at once, the loading procedure will sort the data in chronological order. Even data from multiple days can be selected. After clicking "open", the data will be loaded and processed according to the user defined settings. In the MATLAB command window short messages will show when data is being loaded, being processed and when the procedure is completed. If the loading procedure is aborted by the user, some error messages will show, but the user can still use the GUI and try loading the data again. It will not stop the GUI from functioning. Once the data has been loaded it is stored in a global structure called "Processed\_data". It is a global structure to give the user the ability to call the global structure in his/her own workspace by typing "global Processed\_data" in the command window of MATLAB. The structure contains thirteen fields of data according to Table G.1. As can be seen in Table G.1 more data is available than is used in the thesis work. This gives the opportunity to use the other data as well in the ML detection and characterisation methods. Once the data has been loaded and processed, the user can click on "Show processed data" and the routine "Plot bare data" is started. The plot routine shows the "bare" processed data. This means that the x and y axis are not filled with the height and time. This done to make it easier to find specific parts in the dataset. If one selects a part of the data in the figure, the x and y axis are given according to their corresponding coordinates in the data instead of the accompanying time and height. Of course if the user want to have the accompanying time and height along the axis, this is possible because the time and height vectors are available in the "Processed\_data" structure. This concludes the first section on the GUI. The next section will continue with more depth towards the use of machine learning algorithms and the option to create and apply user defined machine learning algorithms.

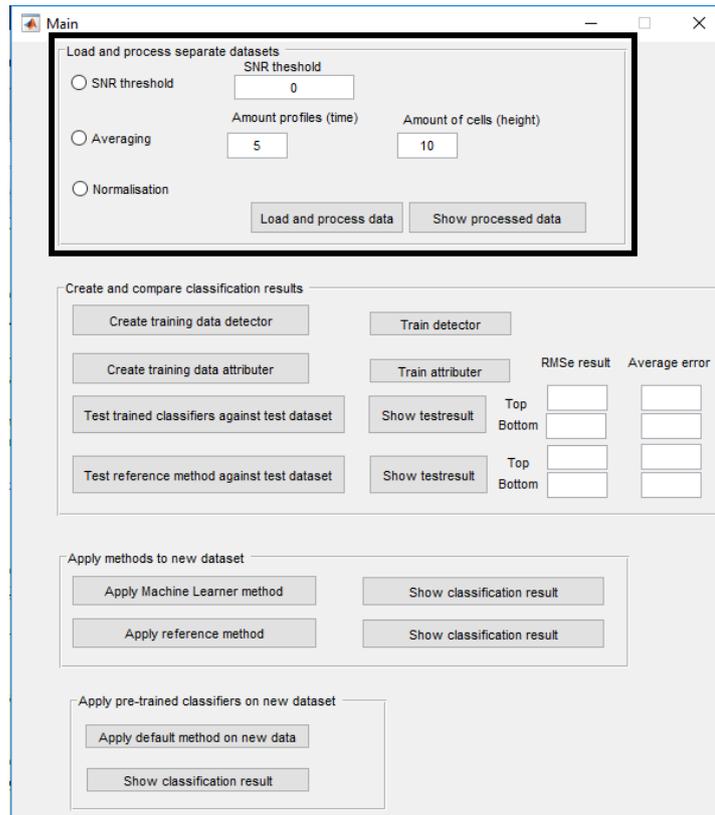


Figure G.1: Example of the GUI, within the black rectangle is the part of the tool situated which focuses on loading, processing and displaying data.

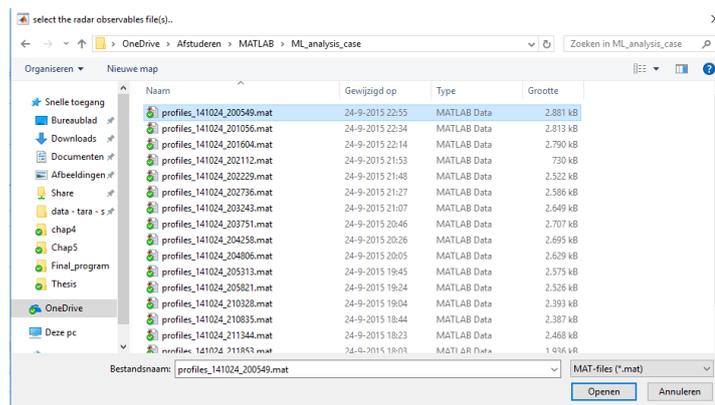


Figure G.2: Load data window.

Table G.1: List of available data after pre-processing

Field name	Contents
ZH	Horizontal equivalent reflectivity ( $Z_{e,hh}$ [dBZ])
ZV	Vertical equivalent reflectivity ( $Z_{e,vv}$ [dBZ])
ZHV	Cross polar equivalent reflectivity ( $Z_{e,hv}$ [dBZ])
Rhohv	Copolar correlation coefficient ( $\rho_{hv}$ )
Vz	Vertical velocity [m/s]
Wh	Doppler spectral width [m/s]
Skh	Doppler skewness
time	Serial data numbers
height	Altitude [m]
noise	Noise matrix [dBZ]
ZDR	Differential reflectivity [dB]
LDR	Linear depolarisation ratio [dB]
Psidp	Differential phase [degrees]

## G.2. Use machine learning part of GUI

The GUI has the possibility to train detectors and attributers with sets of training data. The learned models can be applied to a fixed test dataset for performance comparison or can be applied to a completely new dataset. This section focusses specifically on the part of the GUI indicated by the black rectangle in Figure G.3. To learn a model detector with the GUI, first a training dataset for the detector must be created. Once the

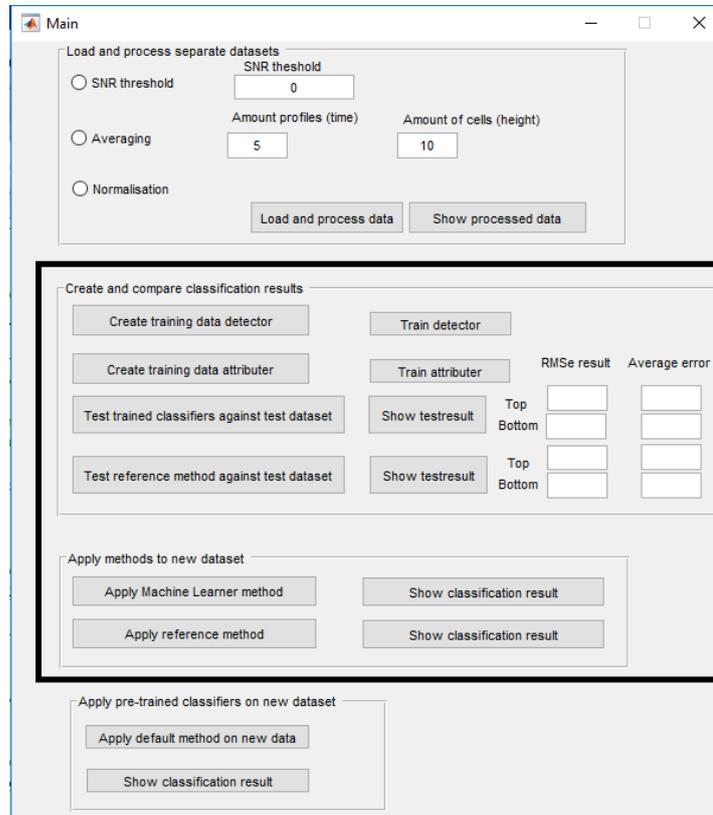


Figure G.3: Example of the GUI, within the black rectangle is the part of the tool situated which focusses on machine learning.

user clicks on "Create training data detector", a routine is started. This routine first loads and pre-processes all the "\*.mat" files in the folder "\root\Data\Training\_data\ML\_data\Detector\". This folder contains all training data which contains a ML. Features are extracted from this dataset and stored. Secondly the routine loads and pre-processes all the "\*.mat" files in the folder "\root\Data\Training\_data\Non\_ML\Detector\". Again features are extracted and stored. Both sets of extracted features are then merged in one array and the classification is added in the last column. The resulting training dataset is again a global variable with the name "Trainingdata\_total". Users can recall the training data array in their own workspace by typing "global Trainingdata\_total" in the MATLAB command window. Now a training dataset has been created for the detector, the detector machine learning algorithm can be trained. This is performed by clicking on "Train detector". The training dataset is then supplied to the training routine and a resulting trained model is the output. The model is stored again in a global structure called "TrainedModel\_profile".

The same procedure needs to be completed for the attributer. The creation of a training dataset is started by clicking "Create training data attributer". This routine is a bit more complex since it uses an algorithm to determine the ML boundaries. First the training dataset from "root\Data\Training\_data\ML\_data\Class\" is loaded. The ML definition algorithm is then applied to this training dataset. This results in a lower and upper boundary of the ML over the entire dataset. The routine "Create\_train\_data\_ML\_classifier" is started which gathers all the data between the boundaries of the ML and stores that in a array. Secondly data is loaded from "root\Data\Training\_data\Non\_ML\Class\" which contains the training data without ML. The exact data patches are extracted again by "Create\_train\_data\_ML\_classifier.m". This data without ML is stored in a separate array. Finally both arrays are merged and supplemented with the classification annotation (1 = ML, 0 = no ML). Now also the completed training dataset is stored in a global variable called "Trainingdata\_ML\_class". This array is supplied to the training algorithm by clicking on "Train attributer". The resulting trained model is

stored in a global structure called "TrainedModel\_class". Now both the detector and attributer have a trained model and can be applied to new datasets.

Two options are available for the trained models: apply the models to new datasets or apply the models to the test dataset. The test dataset is a fixed dataset which is used for performance comparison since the ground truth of this dataset is known. If the first option is required, the user can click on "Apply Machine Learner method" in the button group "Apply methods to new dataset". The just trained methods will be applied to the data selected by the user. Once the classification procedure is completed a notification is displayed in the MATLAB command window. After this message, the results can be shown to the user by clicking "Show classification result" next to the "Apply Machine Learner method" button. The post-processed output of the attributer is also stored in "Class\_result\_Machine\_Learn.mat" as a binary array (0 = no ML, 1 = ML). When a performance measure is required, the method should be applied to the test dataset. The same procedure as before is started, but now the dataset is fixed and as a result the user is not allowed to choose a dataset. The test dataset is loaded from "root\Data\Test\_data\total". Classification may take a while, since the test dataset is rather large (over 17000 profiles). Once completed, a message will appear in the MATLAB command window. After the message has appeared, the user can click on "Show testresult" and the results of the method is shown to user, just as with the previous option. Now the performance parameters are added. A confusion matrix of the detection process is displayed as well as the RMSe and mean error between the detected ML boundaries and true ML boundaries.

### G.2.1. Modify machine learners and training data

It is possible to generate your own machine learned models and data with the GUI, but it involves some extra steps and modification of the code of the GUI. Only proceed if you are familiar with MATLAB code and scripts. This section will be divided into four parts: modifying the detector training data, creation and infusion of own detectors, modifying attributer training data and lastly the use of different machine learned attributers.

#### Modifying detector training data

To change the training dataset for the detector the following steps are necessary. First create two datasets: one containing only data where a ML exists and one containing only data without ML. Place the training dataset with the ML in the folder "\root\Data\Training\_data\ML\_data\Detector\" where root is the folder from which the program is run (MATLAB current folder). Secondly place the training dataset without ML signature in the "\root\Data\Training\_data\Non\_ML\Detector\". When in the program "Create training data detector" is clicked, the program automatically loads all "\*.mat" files from both folders, calculates the features and generates the training data array. The rest of the program can be operated as described previously. If one would like to add or change features for the detector, the m-file "Calculate\_profile\_features.m" needs to be modified to ones liking. As long as the output structure as shown in Table 4.2 is maintained. Of course the amount of columns may be changed as a result of adding or removing features. After modifying the feature set, always a new machine learning algorithm must be trained. How to do this is shown next.

#### Create and apply new detectors

The creation of a new machine learning algorithm is not difficult. After clicking on "Create training data detector" in the GUI, the training array is available, it is merely not visible in the workspace. To make it available in the workspace "global Trainingdata\_total" must be typed in the command window of MATLAB. The training array with the calculated features is now available in the workspace. To easily create a new machine learning algorithm the "Classification learner app" can be opened, this app is available in the top section under the tab "APPS" as shown in Figure G.4 in the red circle. Start the app by clicking on it. The screen as

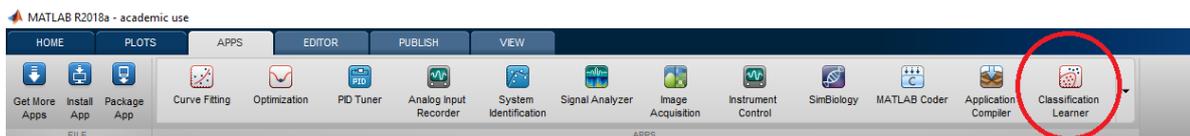


Figure G.4: Location of the classification learner APP.

shown in Figure G.5 opens. After clicking on "New session" (within the red circle), the "New session" screen

opens. Select from the drop down menu under "Workspace Variable" the training data (Trainingdata\_total). One could already pre-select which feature should be included or not and the amount of cross validation can be chosen here. If settings are made click "Start session". The default screen of the classification learner

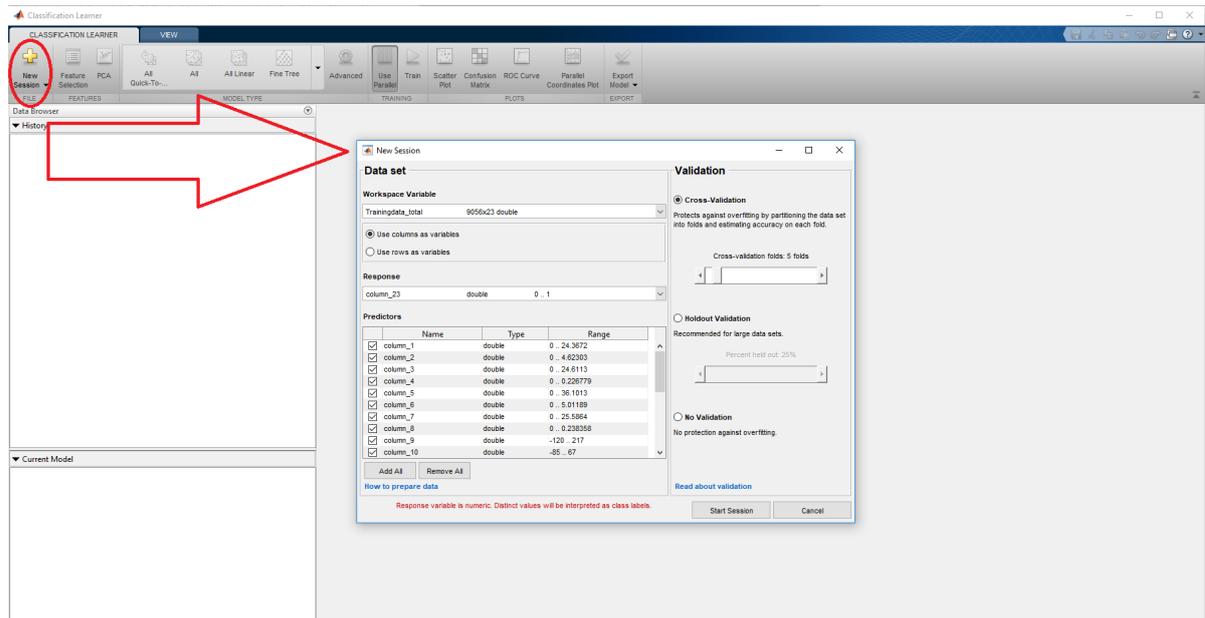


Figure G.5: Opening screen of classification learner.

opens (see Figure G.6) where a multitude of options are available. On the right hand side (within the red area) different features (named Predictors in MALTAB) and different ways to display the features can be selected. On the left side (within the yellow area) the machine learning algorithms can be selected to train. If one clicks the down arrow between "advanced" and the "Fine tree" a complete list of available machine

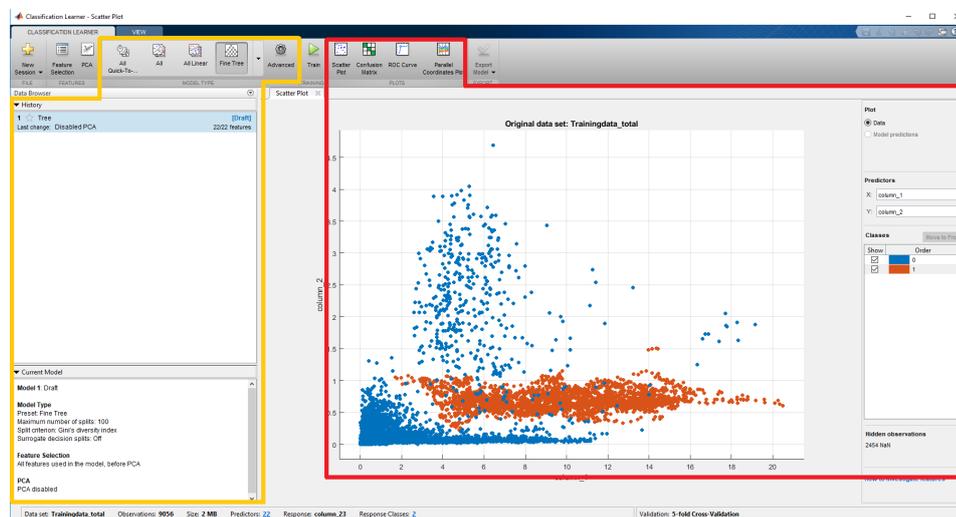


Figure G.6: Main screen of classification learner app.

learning algorithms becomes available. One can select an algorithm and it is added to the history. One can also select a multitude of algorithms by clicking one of the "all" algorithms. All the algorithms of that type are then added to the history. Under "advanced" some extra options are available, which depend on the selected algorithm. Also the user may (de)select features using the "Feature Selection" option in the top bar. When the user selected algorithm(s) are ready for training, one can click "Train". This button start the training procedure for all selected algorithms. If "Use Parallel" is on (by default) the algorithms can be trained in parallel if the used computer has multiple cores. This saves time if more than 2 algorithms are selected

for training. Otherwise I would recommend to switch it off. Once all algorithms are ready with training, one could see the performance of the algorithms. This performance is based on the same training dataset and thus give a good indication of the classification performance, but they are not tested against a separate test dataset. If the user has selected the algorithm which is going to be used in the program, "Export Model" on the far right side of the top bar can be clicked. Then the option "Generate Code" is selected. In the MATLAB editor the generated code appears. To use the code, it must be saved to the root folder of the program. Remember the name you gave it. Now a code is ready it must be used by the program, so open "main.m" in the editor. Scroll down to the part which is initiated after pressing the "Train\_profile\_class" button. Look for the line `[TrainedModel_profile Accuracy_ensemble] = trainClassifier_ebt_det_1(Trainingdata_total);` as shown in Figure G.7. This line initiates the training procedure of the detector. In this example `trainClassifier_ebt_det_1` (in the red rectangle) is the used m-file for training the detector. Replace this by the m-file name you just

```

432 % --- Executes on button press in Train_profile_class button.
433 function Train_profile_class_Callback(hObject, eventdata, handles)
434 % hObject handle to Train_profile_class button (see GCBO)
435 % eventdata reserved - to be defined in a future version of MATLAB
436 % handles structure with handles and user data (see GUIDATA)
437
438 %% This button trains the ML detector
439 global Trainingdata_total
440 global TrainedModel_profile
441
442 % Train the ML detector. ensemble of bagged trees in this case, but any
443 % other classifier can be trained. You just need to create a code with the classification learner app and put it
444 % in an M file. Then change the name of the M file below to learn your own
445 % classifier
446 % That is why Trainingdata_total is a global variable. You can recall the
447 % global variable into your own workspace and use the trainingdata in the
448 % classification learner app.
449 disp('Training ML detector..')
450 [TrainedModel_profile Accuracy_ensemble] = trainClassifier_ebt_det_1(Trainingdata_total);
451 disp('Ready training ML detector')
452

```

Figure G.7: Portion of code of main.m where the training of an algorithm is performed.

created when the generated code was stored. Now save the program and run the main program. When after generating the training data "Train detector" is pressed, the program will use the just generated machine learning method. The rest of the program doesn't need modification, because the trained model is stored as a structure and is used everywhere in the program. Since the trained model is a global variable, the user can also recall the model to his/her workspace.

### Modifying attributer training data

Changing the training dataset for the attributer is a bit more difficult, since it is not just a replacement of datasets. One could keep the existing dataset, but still change the training dataset because only a part of the data within the dataset is used. The ML data is now selected based on the definition, but any user defined mechanism can be created. The ML training data is located in "`\root\Data\Training_data\ML_data\Class\`". One can chose to copy a new dataset here or leave the dataset as it is. The training data without ML is located in "`\root\Data\Training_data\Non_ML\Class\`". Again one can chose to copy new data in this folder or leave the dataset as it is. Note that the data in this folder does contain data with a ML, but parts of the data where the ML is not located is selected to become the training data. Once datasets have been uploaded in the corresponding folders it is time to select the ML from the ML training dataset. The program uses the boundaries according to the definition, but this should be checked manually first. The procedure is as follows: First load the ML training dataset with the GUI option "Load and process data" (note that SNR threshold must be on and set to 10, averaging is on and set to 5 for both windows and normalisation is set to on (1)), then run the ML definition ("`[Boundary_up Boundary_low] = ML_definition`") to generate the boundaries. The boundaries can be made visible using "Show\_definition\_boundaries". Please look if the detected boundaries are within expectations. If some detected boundaries are out of the ordinary, note the VPR numbers, so they can be later excluded from the training dataset. The selection of the ML data is now ready. The same type of procedure must be followed to generate training data outside the ML. As previously mentioned the training dataset is located in "`\root\Data\Training_data\Non_ML\Class\`". A selection must be made otherwise way too much training data is selected. This training dataset can be loaded and displayed using the program. Remember to turn on SNR threshold to 10 dB, averaging to 5 for both windows and normalisation. When the data is plotted with "Plot processed data" one should select which data patches from the displayed dataset should become the training dataset. Note the profile and range resolution cell numbers of the selected data

(use the data cursor in the figure toolbar to show the correct profile and range resolution cell numbers). Once notation of the required data is made, both the ML data and non-ML data have been selected. The next step is to provide this information to the program. First start with the ML data. In principle the ML data is automatically selected just as you have done manually by using the definition. Only exceptions should be provided to the program. Open "main.m" and scroll down to "%% determine ML according to definition". Add your noted profile numbers below "%remove a few errors from the boundaries" for both the upper and lower boundary. Set these values to NaN. Secondly the selection of non-ML training data must become known to the program. Open for this purpose the m-file "Create\_train\_data\_ML\_classifier.m" and scroll down to "%% generate train data outside the ML". In this section the training data is taken from the dataset. The noted patches of data (VPR and range resolution cell numbers) are now required. The user may create his own set of data as long as the output array structure shown in Table 4.4 is maintained. Once datasets have been programmed in the m-file, the rest of the program can be run without modification. The attributer is now trained with a new training dataset. If one would like to add or change features for the attributer, the m-file "Create\_train\_data\_ML\_classifier.m" needs to be modified to ones liking. by default only  $Z_{e,hh}$  (ZH),  $Z_{DR}$  (ZDR),  $L_{DR}$  (LDR),  $Z_{e,hv}$  (ZHV) and  $\rho_{hv}$  (rho hv) are used as features, but one can add or remove columns as long as the structure output of the m-file as shown in Table 4.4 is maintained. After modifying feature(s), always a new machine learning algorithm must be trained. How to do this is shown next.

### Create and apply new attributers

Basically the exact same procedure as the creation of a new machine learning algorithm for the detector applies. First the training data must be generated by clicking on "Create training data attributer". Once the training dataset is generated it can become available for the user by typing "global Trainingdata\_ML\_class" in the MATLAB command window. Next a new machine learning algorithm can be generated by creating a new session in the classification learner app and selecting this training dataset from the workspace. The entire procedure can be read in the part on the creation of a new detector. Once a new code is generated and stored, the user must edit the program "main.m" to be able to use the new attributer. Scroll down to "%% This button trains the ML attributer", shown in Figure G.8. Somewhere close below that line (in the red

```

455 % --- Executes on button press in Train ML class button.
456 function Train_ML_class_button_Callback(hObject, eventdata, handles)
457 % hObject handle to Train_ML_class_button (see GCBO)
458 % eventdata reserved - to be defined in a future version of MATLAB
459 % handles structure with handles and user data (see GUIDATA)
460
461 %% This button trains the ML attributer
462
463 global Trainingdata_ML_class
464 global TrainedModel_class
465
466 % Train the classifier. Coarse KNN in this case, but any
467 % other classifier can be trained. You just need to create a code with the classification learner app and put it
468 % in an M file. Then change the name of the M file below to learn your own
469 % classifier
470 % That is why Trainingdata_ML_class is a global variable. You can recall the
471 % global variable into your own workspace and use the trainingdata in the
472 % classification learner app.
473 disp('Training ML classifier')
474 [TrainedModel_class Accuracy] = trainClassifier_class_KNN_coarse(Trainingdata_ML_class);
475 disp('Ready training ML classifier')
476

```

Figure G.8: Portion of code of main.m where the training of an algorithm is performed.

rectangle) resides the following line of code "[TrainedModel\_class Accuracy] = trainClassifier\_class\_KNN\_coarse(Trainingdata\_ML\_class)". Again replace the part after the "=" symbol by the just generated m-file name. Now the entire program functions with the newly created attributer.

### G.3. Apply default algorithms to new dataset

To prevent the need to learn two algorithms every time the program is used, the program has the option to apply a default machine learned algorithm to new data. This is shown in Figure G.9. The default machine learned algorithms are the outcome of the thesis (ensemble of bagged trees detector and coarse KNN attributer). The user may of course create a new default. There are basically two ways to create a new default: directly from the classification learner app and after training with the program. The first way is to change the

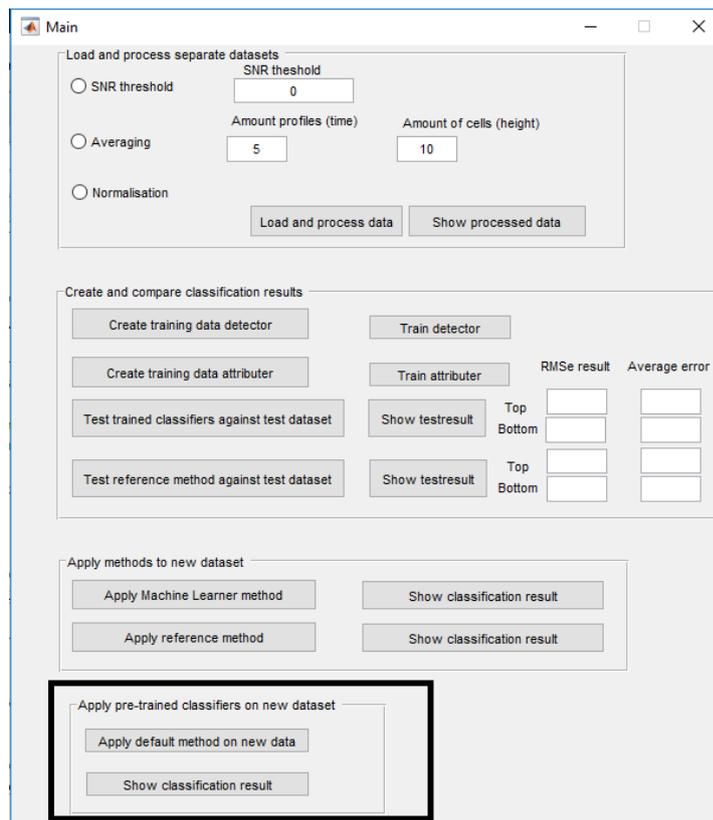


Figure G.9: Example of the GUI, within the black rectangle is the part of the program indicated which focusses on applying a default machine learned method to new data.

export model option in the classification learner. In the previous steps the model was exported as a code, but the model can also be exported as a trained model. Options "export model" and "export compact model" are available. Both options can be chosen. The regular "export model" option also includes the training dataset, which is not necessary for this goal. When either of the two options are chosen a name can be given to the model. The model becomes then available in the workspace under that name. For a smooth implementation of the new default, please use the name "TrainedModel\_profile" for the detector and "TrainedModel\_class" for the attributer. The only thing which needs to be done now to let the new models become the new default is to store both models as a .mat file. Type "save('TrainedModel\_profile.mat', 'TrainedModel\_profile')" for the detector and the similar command "save('TrainedModel\_class.mat', 'TrainedModel\_class')" for the attributer to save the models. By saving them, they automatically become the new default models, since the program loads both specified .mat files.



# Bibliography

- [1] Hervé Andrieu and Jean Dominique Creutin. Identification of vertical profiles of radar reflectivity for hydrological applications using an inverse method. part i: Formulation. *Journal of Applied Meteorology (1988-2005)*, 34(1):225–239, 1995. ISSN 08948763, 15200450. URL <http://www.jstor.org/stable/26187206>.
- [2] L. Baldini and E. Gorgucci. Characteristics of the melting layer in the mediterranean region from dual polarization radar measurements at vertical incidence. 6:4138–4141, July 2005. ISSN 2153-6996. doi: 10.1109/IGARSS.2005.1525826.
- [3] Luca Baldini and Eugenio Gorgucci. Identification of the melting layer through dual-polarization radar measurements at vertical incidence. *Journal of Atmospheric and Oceanic Technology*, 23(6):829–839, 2006. doi: 10.1175/JTECH1884.1. URL <https://doi.org/10.1175/JTECH1884.1>.
- [4] J. Bandera, A. D. Papatsoris, P. A. Watson, J. Tan, and J. W. Goddard. Method for detecting the extent of the melting layer. *Electronics Letters*, 34(22):2104–2105, Oct 1998. ISSN 0013-5194. doi: 10.1049/el:19981462.
- [5] L. J. Battan. Radar observation of the atmosphere. (the university of chicago press) 1973. pp x, 324; 125 figures, 21 tables. £7·15. *Quarterly Journal of the Royal Meteorological Society*, 99(422):793–793, 1973. ISSN 1477-870X. doi: 10.1002/qj.49709942229. URL <http://dx.doi.org/10.1002/qj.49709942229>.
- [6] Hetal Bhavsar and Amit Ganatra. A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, 2, 01 2012.
- [7] S. Boodoo, D. Hudak, N. Donaldson, and M. Leduc. Application of dual-polarization radar melting-layer detection algorithm. *Journal of Applied Meteorology and Climatology*, 49(8):1779–1793, 2010. doi: 10.1175/2010JAMC2421.1. URL <https://doi.org/10.1175/2010JAMC2421.1>.
- [8] Marco Borga, Emmanouil N. Anagnostou, and Witold F. Krajewski. A simulation approach for validation of a brightband correction method. *Journal of Applied Meteorology*, 36(11):1507–1518, 1997. doi: 10.1175/1520-0450(1997)036<1507:ASAFVO>2.0.CO;2. URL [https://doi.org/10.1175/1520-0450\(1997\)036<1507:ASAFVO>2.0.CO;2](https://doi.org/10.1175/1520-0450(1997)036<1507:ASAFVO>2.0.CO;2).
- [9] Edward A. Brandes and Kyoko Ikeda. Freezing-level estimation with polarimetric radar. *Journal of Applied Meteorology*, 43(11):1541–1553, 2004. doi: 10.1175/JAM2155.1. URL <https://doi.org/10.1175/JAM2155.1>.
- [10] M. Cheng and C. G. Collier. An objective method for recognizing and partially correcting bright-band error in radar images. *Journal of Applied Meteorology*, 32(6):1142–1149, 1993. doi: 10.1175/1520-0450(1993)032<1142:AOMFRA>2.0.CO;2. URL [https://doi.org/10.1175/1520-0450\(1993\)032<1142:AOMFRA>2.0.CO;2](https://doi.org/10.1175/1520-0450(1993)032<1142:AOMFRA>2.0.CO;2).
- [11] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972. ISSN 0001-0782. doi: 10.1145/361237.361242. URL <http://doi.acm.org/10.1145/361237.361242>.
- [12] F. Fabry, G. L. Austin, and D. Tees. The accuracy of rainfall estimates by radar as a function of range. *Quarterly Journal of the Royal Meteorological Society*, 118(505):435–453, 1992. ISSN 1477-870X. doi: 10.1002/qj.49711850503. URL <http://dx.doi.org/10.1002/qj.49711850503>.
- [13] Frederic Fabry and Isztar Zawadzki. Long-term radar observations of the melting layer of precipitation and their interpretation. *Journal of the Atmospheric Sciences*, 52(7):838–851, 1995. doi: 10.1175/1520-0469(1995)052<0838:LTR00T>2.0.CO;2. URL [https://doi.org/10.1175/1520-0469\(1995\)052<0838:LTR00T>2.0.CO;2](https://doi.org/10.1175/1520-0469(1995)052<0838:LTR00T>2.0.CO;2).

- [14] Scott E. Giangrande, John M. Krause, and Alexander V. Ryzhkov. Automatic designation of the melting layer with a polarimetric prototype of the wsr-88d radar. *Journal of Applied Meteorology and Climatology*, 47(5):1354–1364, 2008. doi: 10.1175/2007JAMC1634.1. URL <https://doi.org/10.1175/2007JAMC1634.1>.
- [15] Jonathan J. Gourley and Chris M. Calvert. Automated detection of the bright band using wsr-88d data. *Weather and Forecasting*, 18(4):585–599, 2003. doi: 10.1175/1520-0434(2003)018<0585:ADOTBB>2.0.CO;2. URL [https://doi.org/10.1175/1520-0434\(2003\)018<0585:ADOTBB>2.0.CO;2](https://doi.org/10.1175/1520-0434(2003)018<0585:ADOTBB>2.0.CO;2).
- [16] P. J. Hardaker, A. R. Holt, and C. G. Collier. A melting-layer model and its use in correcting for the bright band in single-polarization radar echoes. *Quarterly Journal of the Royal Meteorological Society*, 121(523):495–525, 1995. ISSN 1477-870X. doi: 10.1002/qj.49712152303. URL <http://dx.doi.org/10.1002/qj.49712152303>.
- [17] R. Houze. *Cloud Dynamics*. Academic Press, 1993. ISBN 9780080959955.
- [18] Robert A. Houze. Structures of atmospheric precipitation systems: A global survey. *Radio Science*, 16(5):671–689, 1981. ISSN 1944-799X. doi: 10.1029/RS016i005p00671. URL <http://dx.doi.org/10.1029/RS016i005p00671>.
- [19] Anthony J. Illingworth and Robert W. Thompson. Radar bright band correction using the linear depolarisation ratio. 2011.
- [20] John S. Kain, Stephen M. Goss, and Michael E. Baldwin. The melting effect as a factor in precipitation-type forecasting. *Weather and Forecasting*, 15(6):700–714, 2000. doi: 10.1175/1520-0434(2000)015<0700:TMEAAF>2.0.CO;2. URL [https://doi.org/10.1175/1520-0434\(2000\)015<0700:TMEAAF>2.0.CO;2](https://doi.org/10.1175/1520-0434(2000)015<0700:TMEAAF>2.0.CO;2).
- [21] M. Kitchen, R. Brown, and A. G. Davies. Real-time correction of weather radar data for the effects of bright band, range and orographic growth in widespread precipitation. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1231–1254, 1994. ISSN 1477-870X. doi: 10.1002/qj.49712051906. URL <http://dx.doi.org/10.1002/qj.49712051906>.
- [22] Wim Klaassen. Radar observations and simulation of the melting layer of precipitation. *Journal of the Atmospheric Sciences*, 45(24):3741–3753, 1988. doi: 10.1175/1520-0469(1988)045<3741:ROASOT>2.0.CO;2. URL [https://doi.org/10.1175/1520-0469\(1988\)045<3741:ROASOT>2.0.CO;2](https://doi.org/10.1175/1520-0469(1988)045<3741:ROASOT>2.0.CO;2).
- [23] Sergey Y. Matrosov, Kurt A. Clark, and David E. Kingsmill. A polarimetric radar approach to identify rain, melting-layer, and snow regions for applying corrections to vertical profiles of reflectivity. *Journal of Applied Meteorology and Climatology*, 46(2):154–166, 2007. doi: 10.1175/JAM2508.1. URL <https://doi.org/10.1175/JAM2508.1>.
- [24] P. Meischner. *Weather Radar: Principles and Advanced Applications*. Physics of Earth and Space Environments. Springer Berlin Heidelberg, 2013. ISBN 9783662052020. URL <https://books.google.nl/books?id=S6EyBwAAQBAJ>.
- [25] Thomas Pfaff, Alexander Engelbrecht, and Jochen Seidel. Detection of the bright band with a vertically pointing k-band radar. *Meteorologische Zeitschrift*, 23(5):527–534, 12 2014. doi: 10.1127/metz/2014/0605. URL <http://dx.doi.org/10.1127/metz/2014/0605>.
- [26] M. A. Rico-Ramirez, I. D. Cluckie, and D. Han. Correction of the bright band using dual-polarisation radar. *Atmospheric Science Letters*, 6(1):40–46, 2005. ISSN 1530-261X. doi: 10.1002/asl.89. URL <http://dx.doi.org/10.1002/asl.89>.
- [27] M. A. Rico-Ramirez, I. D. Cluckie, G. Shepherd, and A. Pallot. A high-resolution radar experiment on the island of jersey. *Meteorological Applications*, 14(2):117–129, 2007. ISSN 1469-8080. doi: 10.1002/met.13. URL <http://dx.doi.org/10.1002/met.13>.
- [28] M. A. Rico-Ramirez and I. D. Cluckie. Bright-band detection from radar vertical reflectivity profiles. *International Journal of Remote Sensing*, 28(18):4013–4025, 2007. doi: 10.1080/01431160601047797. URL <https://doi.org/10.1080/01431160601047797>.

- [29] Caroline Sandford, Anthony Illingworth, and Robert Thompson. The potential use of the linear depolarization ratio to distinguish between convective and stratiform rainfall to improve radar rain-rate estimates. *Journal of Applied Meteorology and Climatology*, 56(11):2927–2940, 2017. doi: 10.1175/JAMC-D-17-0014.1. URL <https://doi.org/10.1175/JAMC-D-17-0014.1>.
- [30] Catherine J. Smith. The reduction of errors caused by bright bands in quantitative rainfall measurements made using radar. *Journal of Atmospheric and Oceanic Technology*, 3, 03 1986.
- [31] T. J. Smyth and A. J. Illingworth. Radar estimates of rainfall rates at the ground in bright band and non-bright band events. *Quarterly Journal of the Royal Meteorological Society*, 124(551):2417–2434, 1998. ISSN 1477-870X. doi: 10.1002/qj.49712455112. URL <http://dx.doi.org/10.1002/qj.49712455112>.
- [32] R.E. Stewart, J.D. Marwitz, J.C. Pace, and R.E. Carbone. CHARACTERISTICS THROUGH THE MELTING LAYER OF STRATIFORM CLOUDS. *JOURNAL OF THE ATMOSPHERIC SCIENCES*, 41(22):3227–3237, 1984. ISSN 0022-4928. doi: {10.1175/1520-0469(1984)041<3227:CTTMLO>2.0.CO;2}.
- [33] Pierre Tabary, Alain Le Henaff, Gianfranco Vulpiani, Jacques Parent du Châtelet, and Jonathan J. Gourley. Melting layer characterization and identification with a c-band dual-polarization radar : a long-term analysis. 2006.
- [34] Elizabeth J. Thompson, Steven A. Rutledge, Brenda Dolan, V. Chandrasekar, and Boon Leng Cheong. A dual-polarization radar hydrometeor classification algorithm for winter precipitation. *Journal of Atmospheric and Oceanic Technology*, 31(7):1457–1481, 2014. doi: 10.1175/JTECH-D-13-00119.1. URL <https://doi.org/10.1175/JTECH-D-13-00119.1>.
- [35] K.A. Tilford, I.D. Cluckie, and R.J. Griffith. Vertical reflectivity characteristics and bright band correction. *Radar hydrology for real time flood forecasting. Proceedings of an advanced study course*, pages 47–65, 2001.
- [36] Bertrand Vignal, Hervé Andrieu, and J. Dominique Creutin. Identification of vertical profiles of reflectivity from volume scan radar data. *Journal of Applied Meteorology*, 38(8):1214–1228, 1999. doi: 10.1175/1520-0450(1999)038<1214:IOVPOR>2.0.CO;2. URL [https://doi.org/10.1175/1520-0450\(1999\)038<1214:IOVPOR>2.0.CO;2](https://doi.org/10.1175/1520-0450(1999)038<1214:IOVPOR>2.0.CO;2).
- [37] Bertrand Vignal, Gianmario Galli, Jürg Joss, and Urs Germann. Three methods to determine profiles of reflectivity from volumetric radar data to correct precipitation estimates. *Journal of Applied Meteorology*, 39(10):1715–1726, 2000. doi: 10.1175/1520-0450-39.10.1715. URL <https://doi.org/10.1175/1520-0450-39.10.1715>.
- [38] Allen B. White, Daniel J. Gottas, Eric T. Strem, F. Martin Ralph, and Paul J. Neiman. An automated bright-band height detection algorithm for use with doppler radar spectral moments. *Journal of Atmospheric and Oceanic Technology*, 19(5):687–697, 2002. doi: 10.1175/1520-0426(2002)019<0687:AABHDA>2.0.CO;2. URL [https://doi.org/10.1175/1520-0426\(2002\)019<0687:AABHDA>2.0.CO;2](https://doi.org/10.1175/1520-0426(2002)019<0687:AABHDA>2.0.CO;2).
- [39] Daniel Wolfensberger, Danny Scipion, and Alexis Berne. Detection and characterization of the melting layer based on polarimetric radar scans. *Quarterly Journal of the Royal Meteorological Society*, 142:108–124, 2016. ISSN 1477-870X. doi: 10.1002/qj.2672. URL <http://dx.doi.org/10.1002/qj.2672>.
- [40] M.K. Yau and R.R. Rogers. *A Short Course in Cloud Physics*. Butterworth-Heinemann, 1996. ISBN 9780750632157.
- [41] Jian Zhang, Carrie Langston, and Kenneth Howard. Brightband identification based on vertical profiles of reflectivity from the wsr-88d. *Journal of Atmospheric and Oceanic Technology*, 25(10):1859–1872, 2008. doi: 10.1175/2008JTECHA1039.1. URL <https://doi.org/10.1175/2008JTECHA1039.1>.
- [42] D. S. Zrnić, R. Raghavan, and V. Chandrasekar. Observations of copolar correlation coefficient through a bright band at vertical incidence. *Journal of Applied Meteorology*, 33(1):45–52, 1994. doi: 10.1175/1520-0450(1994)033<0045:OCCCT>2.0.CO;2. URL [https://doi.org/10.1175/1520-0450\(1994\)033<0045:OCCCT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1994)033<0045:OCCCT>2.0.CO;2).