01-07-2022
Utrecht University,
Nelen & Schuurmans,
KNMI

Validating a Deep Generative Precipitation Nowcasting Model on the Netherlands

Yoram Frenkiel[1]

Supervisor: Prof. dr. Derek Karssenberg[2]
External Supervisor: Thomas Berends[3]
Second Reader: Dr. Menno Straatsma[4]

Abstract
Precipitation nowcasting tries to predict the intensity of rainfall in the near future. Due to the dependency of many industries on accurate predictions of nowcasting methods, the development of such methods has increased in recent years. In this research, we validated a Deep Generative Model of Radar (DGMR) developed by DeepMind on weather data from the Netherlands. The results of the DGMR were compared to a baseline method S-PROG, based on the PySTEPS framework. It was found that the DGMR outperformed the S-PROG method on multiple metrics, scoring significantly higher for Mean Squared Error and Critical Success Index at timestamp t_0 + 60. However, the DGMR model often failed to correctly classify predictions at long lead times. Therefore, it was concluded that this model is capable of making predictions for the Netherlands. However, re-training of the model is required to achieve the full capabilities of the model.

Keywords
Nowcasting — Precipitation — Deep Learning

[1] *Applied Data Science, Utrecht University, Utrecht, The Netherlands*
E-mail: y.j.frenkiel@students.uu.nl, yoram.frenkiel@nelen-schuurmans.nl

[2] *Professor, Geosciences, Utrecht University, Utrecht, The Netherlands*
[3] *Lead Data Science, Nelen & Schuurmans, Utrecht, The Netherlands*
[4] *Professor, Geosciences, Utrecht University, Utrecht, The Netherlands*

Contents

1. Introduction

Precipitation nowcasting tries to accurately predict the intensity of rainfall in the near future (e.g., 0-6 hours) at high spatial resolutions [1]. This type of forecasting informs decision-making in many industries. These include, but are not limited to, the agriculture industry, air and marine traffic control and the entertainment industry. Besides, it plays an important role in flood risk assessment due to the possibility of flooding as a result of heavy rain [2]. Therefore, the safety and well-being of many people depend on accurate predictions from nowcasting models. Especially considering the growing frequency and severity of heavy rain events in Europe and the United States [3].

The development of more accurate precipitation nowcasting models has increased in recent years. These developments have mostly focused on deep learning approaches because they showed to provide more accurate results than earlier, non-machine learning, methods [1, 2]. Besides, the prediction of heavy rain events and precipitation, in general, requires the analysis of vast amounts of data which is a task well suited for deep learning approaches. A non-machine learning approach commonly used for nowcasting is the framework PySTEPS which provides many different prediction methods [4]. For example, the Spectral Prognosis (S-PROG) method (will refer to this method as the PySTEPS method). This method uses motion fields to predict future radar frames. Motion fields convey the advection of precipitation fields. However, this approach lacks the ability for highly accurate predictions. Mainly due to the addition of a blurring effect, where the predictions get smoothed for increasing lead times [5]. Although this method is able to capture large precipitation structures, it is not able to generate realistic nowscasts. Therefore, blurring is seen as a problem for achieving highly accurate predictions [6].

In 2021 Google's DeepMind department launched a new precipitation nowcasting model with a deep learning framework in pursuit of solving the problems of earlier nowcasting models [7]. Here, a Deep Generative Model (DGM) was used to generate predictions based on a series of radar images. This complex model was trained and tested on radar data from the United Kingdom MET office. Results show this model to outperform traditional nowcasting methods, such as S-PROG and STEPS and even perform better than other state-of-the-art machine learning models [7]. Besides, they found that a majority of meteorologists preferred the prediction of this model over those of competing methods after a review of 56 meteorologists from the MET office [7]. This advanced model could positively influence weather-based decision-making and precipitation nowcasting applications in general, by solving the problems of earlier methods.

The results of this deep learning model are promising, however, it is unknown if these results are applicable to areas outside the scope of the original research as of now. Due to the extensive computational cost, time, and research required for the development of complex precipitation nowcasting models, additional validation can help determine the potential of such models in new regions. Deepmind already validated the model once on data from the USA. This research will further investigate the abilities of this model by validating its performance on Dutch weather data. From this, we can gather how well the model generalizes on Dutch weather data and if it is beneficial to implement this model in the Netherlands. This country is well suited for this research due to the availability of high-quality data and its high amount of average yearly rainfall [8].

1.1 Research Questions

In this research, we aim to find the capabilities of the deep learning methods for precipitation nowcasting in the Netherlands. The main question we will try to answer in this study is *Can the pre-trained DeepMind precipitation model be used for successful precipitation nowcasting in the Netherlands?*

Answering this question will be done on the basis of the answers to three sub-questions. The answers found for the first sub-question will aid in the comprehension of the data used in the model and corresponding data preprocessing steps, and reads: *Is high-quality weather data available for the Netherlands that can be used with the pre-trained DGMR model?* The second sub-question we will answer is: *Are alterations to the model's architecture needed for the prediction of rainfall in the Netherlands?* This question will help to get an understanding of the complex deep learning model and its possible problems for use on Dutch weather data. From this, we could also gather if alterations need to be made. Lastly, we will answer the sub-question: *Are the same patterns noticeable between the results on Dutch weather data and the results found by DeepMind?* This question will lead to a better interpretation of the results and their validity. Answering these three sub-questions makes it possible to determine if the pre-trained model is applicable as a nowcasting model in the Netherlands.

1.2 Relevance

This research on precipitation nowcasting will give insight into the possibilities of complex models in solving the precipitation prediction problem. Furthermore, due to the large dependency of many industries on precipitation nowcasting, studies on more accurate methods will have a positive socio-economic impact. Especially, given the increase in flooding due to rainfall in the Netherlands and western Europe in recent years [9]. More accurate nowcasting models could improve detection of and therefore preparation for certain events.

2. Methods

In this section, we will describe the experiment conducted for this thesis. The experiment will be on the validation of the Deep Generative Model of Radar (DGMR) as introduced by DeepMind on weather data from the Netherlands. By conducting this experiment we hope to answer the question if this pre-trained model can be used for successful precipitation nowcasting for the Netherlands. Additionally, we will investigate if the weather data available for this country is of sufficient quality and fits the requirements for use in the model. Furthermore, we will explore the model architecture and evaluate the predictions of the model. A visualization of the full process can be found in figure 1.

2.1 Data Availability

To validate the model, we will be using the Dutch National Rainfallradar (NRR)[1]. This is a product of Nelen & Schuur-

[1] www.nationaleregenradar.nl

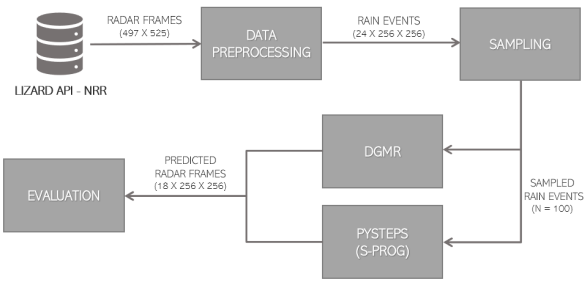

Figure 1. Schematic representation of the full process.

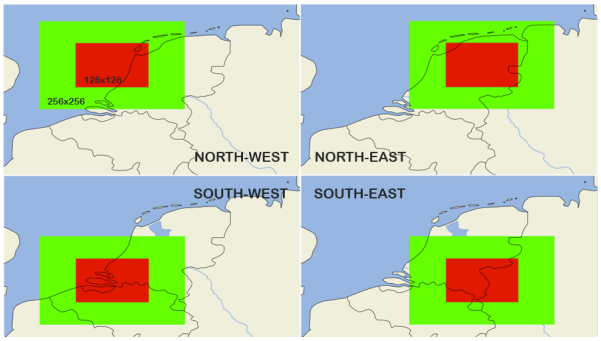

Figure 2. Representation of the four cropped radar frames. The green area indicates the size of the radar frames used as input for the DGMR model. The red area indicates the size of the evaluation area used for both methods.

mans in collaboration with the KNMI (Netherlands), WRD (Germany), and Jabekke (Belgium). Here, images from 6 different radar stations are used in order to create a high-resolution composite radar image representing the current rain situation. The dimensions of the radar image are 497×525 cells, with a cell size of 1km, and cover an area of approximately 261 thousand km^2. Such radar image shows the estimate of precipitation intensity, corrected by ground measurements, over a period of 5 minutes ($mm/5min$)

The main benefits of the NRR are the increase in area and quality compared to the raw radar images from the KNMI (based on only two radar stations). The latter is partly due to the minimization of inconsistencies and missing data points. Such problems can occur for rain data in areas close to (<30 km) or very far away from radar stations. The use of several overlapping radar images mitigates this problem [10]. Besides, the raw radar images are corrected based on more reliable ground stations resulting in a more sound radar image.

The images have a spatial resolution of 1km x 1km and a latency of 5 minutes. These qualities make this unique dataset very well suited for precipitation nowcasting applications. Furthermore, the spatial resolution and latency of the radar images of the NRR are equal to those of the MET office used by DeepMind. The quality and characteristics are therefore in line with the data used in the original research. Data is available from 01-2010 until 05-2022, on the day of writing[2].

2.2 Sampling and Preprocessing

Radar images are extracted using the Lizard API [3]. Here, radar frames from the last full year of available data, in our case 2021 were extracted. From this year data was extracted for days on which 5mm or more rain was measured in de Bilt, the Netherlands[4]. Meaning days on which 5 millimetres of rain was measured over the whole day, certain moments of the day will have no or little rain. This step is implemented to reduce processing times [5]. This resulted in a set of 63 days, spread across the year (see Appendix A). For every day, radar frames were extracted from 00:00 until 23:55, with a temporal offset of 5 minutes. Resulting in 288 frames per day and 18.144 in total.

For pre-processing of these radar images we have used the Python library Rasterio [6] First, we changed the temporal support size of the radar frames from mm/5min to mm/h, also known as resampling. In this way, the temporal support size is equal to that used by DeepMind for training and commonly used in nowcasting applications in general [11, 12]. To achieve this we calculated the moving sum over 12 radar frames, resulting in one radar frame representing the rainfall intensity over a 1-hour period. From now on, the term radar frame will be used for frames representing mm/h. For the second preprocessing step, multiple consecutive radar frames need to be grouped together to form a *rain event*. A rain event consists of 24 consecutive radar frames, covering 120 minutes and is used as input for the nowcasting models. In total 1421 rain events were created with a temporal offset of 1 hour. For the final step of pre-processing, the full radar frame was cropped into smaller frames. This was done to adhere to the strict size limitations for the snapshot of the DGMR model made available [7]. For every rain event four, partially overlapping, crops were made with a dimension of 256×256. Together these four crops cover the entire landmass of the Netherlands and a total area of 147.456km^2, see figure 2.

To create the final test set, we will sample the dataset based on the intensity of rainfall. Here, rain events showing heavy rain (5 mm h^{-1}) will have a higher sampling chance than

[2]Due to the correction of the data a slight delay occurs for data availability. Uncorrected data is also available, but will not be used for this research.

[3]Product of Nelen & Schuurmans, `https://lizard.net/`

[4]Based on measurements from the KNMI.

[5]Extraction of the subset took approximately 10 hours

[6]`https://rasterio.readthedocs.io/en/latest/index.html`

[7]Two models, with different sized input, were made publicly available. One model was used to predict precipitation for the whole of the UK (xxxx×xxxx) and a model to make a prediction on an area of 256×256. The latter was used during training and validation of the model as it meant one full radar image could be cropped and used multiple times. Adding to the size of the dataset

events showing light (1 mm h^{-1}) or even no rain. This step is implemented because most (cropped) radar frames contain no rain. These frames will contribute little to the overall results. By implementing a sampling scheme we can reduce the computational cost needed to make predictions over such frames, without losing statistical power. Computing the sampling probability (q_n for cropped rain event n) was done following the sampling scheme used by DeepMind in the original research, in this way results of our validation will be more comparable to those of DeepMind. Here the following equation was used:

$$q_n = 10^{-3} + \frac{2.2}{24 \times 256 \times 256} \times x_n \quad (1)$$

In this equation x_n is equal to $x_n = \sum_i (1 - exp(-v/1))$, with v representing the intensity of rainfall in mm for grid cell i, when indexing all $24 \times 256 \times 256$ grid cells of a cropped rain event. Furthermore, 10^{-3} is the minimum probability of inclusion and 2.2 is a multiplier used to control the overall inclusion rate. These values are equal to those used in the original research. On the bases of the sampling probability, a test set of size N has been created. The test set consists of 100 rain events spread across the year with a good balance between samples from all four crops. For full details on the test, set used see Appendix B. This sampling probability will also be used during evaluation to correct for the bias introduces by using a dataset that favours radar frames with heavier rainfall. In section 2.4 this will be further explained.

2.3 Deep Generative Model of Radar (DGMR)

For this research, we will be validating a deep generative nowcasting model (DGMR) that is introduced and trained by DeepMind [7]. This model predicts N future radar frames based on M past radar frames. These radar frames convey estimates of precipitation intensity [8]. We will explain this model on the basis of the schematic as seen in figure 3.

We see that the input of the mode is split into context and observations, combined these are equal to 22 radar frames of a rain event. Firstly, the context, i.e. the input frames, is equal to four consecutive radar frames (the previous 20 minutes). The observations, i.e. target frames, are equal to the following 18 frames (the 'future' 90 minutes)[9]. The latter is only used during training, to adjust the parameters of the model and thereby guide learning.

The four input frames are used as input for the conditional generative adversarial network (GAN). This network is specialized in the prediction of precipitation and therefore often used in nowcasting applications [13]. A radar generator is used to generate multiple different future predictions of length 90 minutes (18 frames) based on the four input frames, guided by latent random vector Z and parameters. During training, these parameters are adjusted guided by two loss functions, i.e. error terms, and a regularisation term. These influence parameter adjustments by comparing the generated radar samples to real radar observations and play a significant role in achieving more accurate and realistic predictions.

The input of the first loss function is a random crop of both observed and generated radar frame sequences. This temporal discriminator ensures temporal consistency by classifying real and fake radar frames using a three-dimensional convolutional neural network (CNN). In this way, predictions that are inconsistent in time, so-called jumpy predictions, are penalized. The second loss function is defined by a CNN trained to classify real and generated radar frames. This classification is made on 8 random frames. Hereby, this loss function tries to establish spatial consistency and non-blurry predictions. The latter has shown problematic for earlier nowcasting methods [6]. Accuracy is further improved by introducing a regularization term that imposes a penalty for differences at a per-grid-cell level between predicted and observed radar frames. Here the mean over the N-generated samples is used. The addition of this term improves location-accurate predictions and overall performance.

The model was trained on a large collection of rain events, consisting of cropped radar frames of size 256×256[10] made available by the MET office. A snapshot of the pre-trained model is made available publicly and will be used for this research.

2.4 Evaluation Metrics

For evaluation of the model three metrics will be used: Root Mean Squared Error (RMSE), Critical Succes Index (CSI) and Fraction Skill Score (FSS). These metrics are computed over the central 128×128 grid cells, for each of the predicted frames (see figure 2). This central square is used to prevent the results to be influenced by boundary effects. Such effects can have a negative effect on the model's results due to the model having no information about the area outside its scope, i.e. outside the 256×256 grid[11]. For example, when looking at observations in figure 7, we see a rain structure coming in from the northwest. The DGMR model is unable to predict this precipitation as no information was available regarding this structure.

For evaluation, a weight will be used to correct for the bias introduced by the use of a semi-random sampling scheme.

[8] Note that our data will slightly differ due to correction of ground-level measurements as discussed in section 2.1

[9] Note that not the first two frames of a rain event are not used. The model only uses 22 frames as input for training and 4 frames as input for testing/validation. The reason for these numbers is not mentioned in the original paper but is in line with other nowcasting methods.

[10] The model is capable of predicting precipitation over larger areas, however, this will not be used for this study.

[11] Because the PySTEPS method makes prediction over the full radar frame (of size 497×525), this effect is less prevalent. However, the same approach is used for a better comparison of the results.

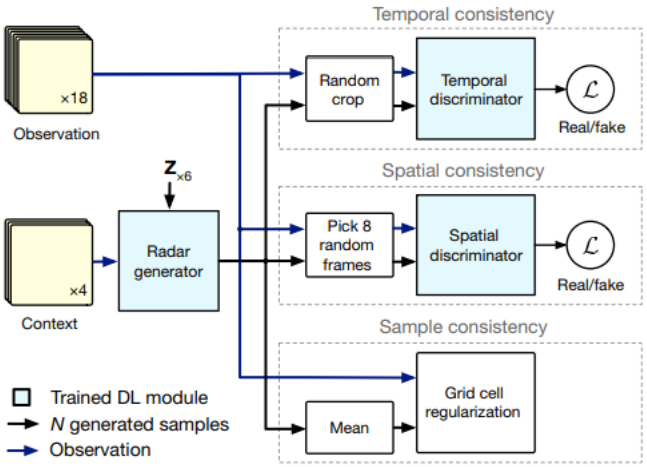

Figure 3. Schematic representation of the DGMR model, showing the four main components of the model. Note the distinction between the black arrows, representing the N generated samples, i.e. N different predictions, and the blue arrows, representing the observation, i.e., the observed ground truth. The latter is only used in the model during training.

For every rain event this weight is equal to $w_n = q_n^{-1}$, with q_n equal to the sampling probability (see equation 1). In this way, rain events with a lower sampling probability, i.e. rain events showing less rain, have a higher weight in the evaluation compensating for the fact they are underrepresented in the test set.

2.4.1 Root Mean Squared Error (RMSE)

RMSE gives a continuous measure of the accuracy of the predictions based on the difference on a per-grid-cell level. We index each grid cell of a predicted radar frame with i and write P_i for the model's prediction for grid cell i, and O_i for the corresponding observed ground truth. This measure is defined as:

$$RMSE = \sqrt{\sum_i \hat{w_n}(P_i - O_i)^2} \tag{2}$$

Here, $\hat{w_n}$ is equal to the normalized weight[12] for event n. Lower is better for RMSE. By addition of the square root RMSE is measured in the same units as the target variable. Thus a RMSE of 1, indicates an average difference of 1mm/h over all grid cells in the radar frame.

2.4.2 Critical Succes Index (CSI)

CSI is used for a binary classification of the predicted frames [7]. This metric evaluates whether or not rainfall exceeds a threshold t, for example low rain ($t = 1mm/h$) or medium rain ($t = 4mm/h$). It is defined as:

$$CSI = \frac{w_n * TP}{w_n * TP + w_n * FP + w_n * FN} \tag{3}$$

[12]The weight for event n is normalized for the sum of all *sampled* events

TP, FP and FN are defined as a true positive ($P_i \leq t, O_i \leq t$), false positive ($P_i \leq t, O_i < t$) and false negative ($P_i < t, O_i \leq t$), respectively. CSI is popular in the nowcasting community due to evaluating the model on both precision and recall in a single measure [7].

2.4.3 Fraction Skill Score (FSS)

FSS is a spatial verification score that gives a direct error measure for the placement of rain. This measure has been shown to give a valid assessment of the performance of precipitation nowcasting. This score is defined as the fraction between correctly classified grid cells and incorrectly classified grid cells in an area of size s^2, multiplied by w_n. With s representing a variable scale in kilometres. The correctness of the prediction of a grid cell is determined in the same way as for CSI, here we used a constant threshold of 1mm/h. For further details on this measure we refer to Skok and Roberts, 2016 [14].

2.5 Validation

The experiment consists of validating the DGMR model developed and trained by DeepMind on rain data from the Netherlands. For every event the DGMR model is used to predict 18 frames, representing the next 90 minutes of rain activity. These 18 frames are then compared to the observed ground truth (i.e. target frames) to compute the RMSE, CSI and FSS for every timestamp. The average of these measurements over the whole test set will be computed.

This procedure is repeated using the nowcasting method S-PROG, to provide a baseline. Here, we use the Lukas-Kanade local feature tracking module to extrapolate a motion field, default parameters are used. For this model, predictions, are made for the full, non-cropped, radar frames. However, computation of the evaluation metrics was done over the same areas as the DGMR model, for a valid comparison. Paired t-tests will be performed to establish significance between the results of both methods, with $\alpha = 0.05$. Here we will compare the results found at timestamp $t_0 + 60$ minutes for all metrics. The validation will be run over 100 rain events. This number is chosen because it provides meaningful results without making the run time unnecessarily long.

3. Results

In this section, the results of the DGRM model and the baseline PySTEPS method will be shown. For all three metrics, the mean is presented for timestamp $t_0 + 30$ minutes, $t_0 + 60$ minutes, and $t_0 + 90$ minutes, with t_0 equal to the timestamp of the final input frame[13].

Tables 1, 2, and 3 show the mean RMSE, CSI and FSS score over all 100 rain events for both precipitation nowcasting methods used. After performing a paired t-test we found a significant difference between the mean RMSE, t(99) = -2.128, p = 0.035. The mean RMSE was lower (i.e. preferred) for

[13]For event 2021-01-01T12:10:00, t_0 is at 2021-01-01T12:30:00.

the predictions by the DGMR model ($\mu = 0.799 \pm 0.677$) compared to the PySTEPS method ($\mu = 0.897 \pm 0.912$). The same paired t-test was performed for the mean CSI (with $t = 1mm$) which also found a significant difference, t(99) = 4.300, p < 0.001. The mean CSI was higher (i.e. preferred) for the DGMR model ($\mu = 0.824 \pm 0.158$) compared to the results from the PySTEPS method ($\mu = 0.778 \pm 0.227$). Lastly, the paired t-test performed for the mean FSS (with $t = 1mm$, $s = 8km$) found no significant difference, t(99) = 1.229, p = 0.221. The mean FSS was higher (i.e. preferred) for the DGMR model ($\mu = 0.579 \pm 0.310$) compared to the PySTEPS method ($\mu = 0.557 \pm 0.338$).

Table 1

RMSE

Method	+30	+60	+90
DGMR	0.519	0.799	0.923
PySTEPS	0.541	0.897	1.097

Table 2

CSI (1mm)

Method	+30	+60	+90
DGMR	0.895	0.824	0.780
PySTEPS	0.862	0.778	0.732

Table 3

FSS (8km, 1mm)

Method	+30	+60	+90
DGMR	0.788	0.579	0.487
PySTEPS	0.726	0.556	0.439

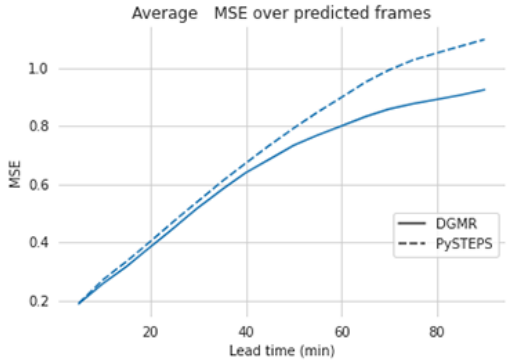

Figure 4. Plot showing average Root Mean Squared Error over test set for every timestamp. An increase in RMSE is visible for leading timestamp for both methods used.

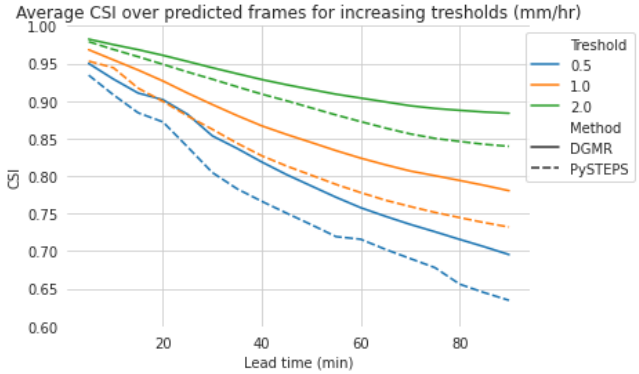

Figure 5. Plot showing average Critical Success Index over test set for every timestamp and different thresholds set at 0.5mm/h (blue), 1mm/h (red) and 2mm/h (green).

3.1 Analysis

The results show the DGMR model to outperform the PySTEPS method for all three metrics. This is further emphasised by the results of the paired t-test, showing a significant difference at $t_0 + 60$ for these two metrics. However, the results from the FSS do not follow the same trend. Not only were the differences found not significant at $t_0 + 60$, but for the lowest scale used (2km), we see the score of the DGMR model to even dip below that of the PySTEPS method. Furthermore, figures 4, 5, and 6 show a clear decline off prediction accuracy with increasing lead times. For both RMSE and CSI we also see a bigger difference in prediction accuracy for increasing lead times, in favour of the DGMR model.

These three metrics help us identify the strengths and weaknesses of the Deep Generative Model. First RMSE, this measure indicates the absolute error. This includes both the intensity and placement of predicted precipitation. We see both methods to start equally strong with a difference of only 0.022

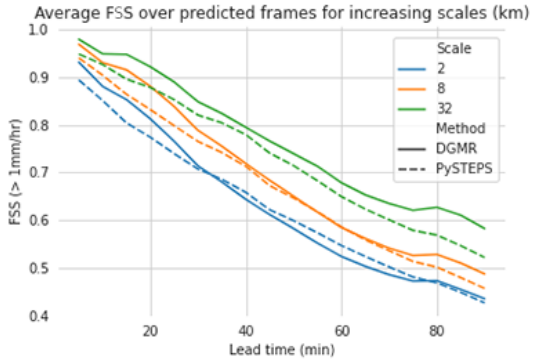

Figure 6. Plot showing average Fraction Skill Score over test set for every timestamp and different scales set at 2km (blue), 8km (red) and 32km (green)

at timestamp +30. However, after 40 minutes the DGMR shows its strengths compared to the PySTEPS method. This difference is most likely due to the blurring effect that is visible for the PySTEPS method as can be seen in figures 7, 8, and 9. The predictions of the DGMR model look more realistic compared to the predictions of the PySTEPS method, especially at increasing lead times. This seems a direct result of the addition of the spatial discriminator to the model to prevent blurring. However, we do see predictions where the DGMR model incorrectly combines multiple smaller structures, as seen in figure 9.

In figure 2 we see the DGMR model outperform the PySTEPS method for CSI on multiple thresholds (0.5mm, 1mm, and 2mm). This is an indication that the DGMR model can more precisely predict the intensity of rainfall. A visual analysis of the prediction in figure 7 indicates this as well. Here, we see the predictions of the PySTEPS method to keep the same (or even slightly increase) the intensity of rainfall over the predicted 90 minutes, whereas the prediction of the DGMR model shows a decrease in rainfall intensity over the predicted place. The latter is more in line with the observed ground truth. However, in figure 8, we see the DGMR is unable to predict an increase in rainfall intensity. Contrary, in some cases, it was found that the DGMR underestimated the intensity of rainfall, with precipitation structures completely disappearing over time for rain events showing light rain. This could be a result of an over-representation of events showing this phenomenon (disappearing rain structures) during training.

Lastly, the FSS metric indicates the spatial accuracy of the prediction. Here the results of both methods show negligible (and non-significant) differences for all scales. This seems to indicate that the use of Deep Learning to estimate the movement of precipitation structures is equally valid as the use of a motion field. However, both methods struggle with capturing the movement of rain structures. Often the DGMR model was able to correctly classify the overall movement of a rain event but failed to identify the movement of smaller precipitation structures.

As discussed before, the addition of the spatial and temporal loss functions does significantly improve the predicting accuracy. The complexity of the model can solve clear problems of the PySTEPS method such as blurring and overestimation of rain intensity. Nonetheless, the predictions of the DGMR model show clear deviations from the ground truth, especially at long lead times. DeepMind noted similar problems with the DGMR model in the original research. However, the problems of underestimating rainfall intensity and the grouping of multiple smaller precipitation structures were not encountered by DeepMind. We assume this to be a direct result of not retraining the model for our research.

4. Discussion

In this chapter, we will discuss the research conducted for this thesis. Additionally, we will evaluate our limitations for this research and propose possible alterations for further studies on this topic.

4.1 Implications

The results have shown that the complex DGMR model is able to make fairly accurate predictions of precipitation for the Netherlands. This model was able to outperform the commonly used PySTEPS method S-PROG on multiple metrics. These results are in line with earlier work on the use of Deep Learning in solving the precipitation prediction problem and emphasise the power of these techniques [1, 7].

Furthermore, in this research, we have been able to successfully apply the model to radar data from the Dutch National Rainfall radar. Not only might this indicate that the available data is of sufficient quality for validating this model, but it could also be an indication of a high generalisability of the model.
Even though the predictions are not flawless at long lead times, the model showed greatly improved results over earlier methods. Especially in solving the problem of blurry predictions. This leap in improvement also has positive implications for the many industries that are dependent on nowcasting techniques. The results from the original research in combination with the results found in this study illustrate a positive socio-economic impact of the use and development of complex deep learning precipitation nowcasting models. However, this positive impact diminishes for increasing lead times.

4.2 Limitations

For this thesis, the reader should consider some limitations.

For one, we validated the DGMR model on Dutch weather data without re-training the model first as this option was not provided publicly by the DeepMind team. This most likely lead to not achieving the full capabilities of the model. Therefore, the differences in prediction accuracy between the two nowcasting methods found could be larger when using validation consisting of both training and testing. This becomes even more prevalent when considering that by using a self-trained model, input size limitations could be avoided. However, the original paper mentioned that training required enormous amounts of processing power and time.

Secondly, when evaluating the models no ensemble metrics were used. Ensemble metrics are computed over multiple predicted samples for every rain event. These metrics provide a statistically stronger evaluation, by, for example, mitigating the influence of random factors. Besides, by using the ensemble technique a larger variety of metrics can be computed. However, due to a significant increase in processing time, this was beyond the scope of this thesis. Besides, the evaluation

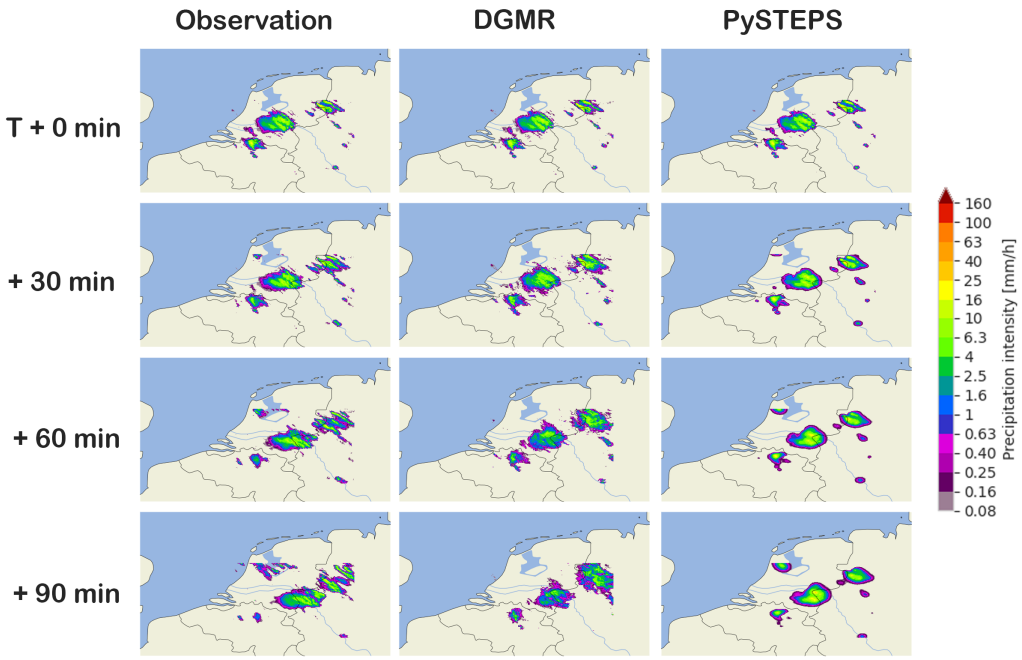

Figure 7. Predictions from both models for precipitation event starting at T=2021-07-27 at 12:00 UTC +1, showing three separate structures of heavy rainfall moving from west to east over the Netherlands. Note, when visualizing precipitation the colour purple is often used to indicate heavy rain, however, in the visualizations in this paper the colour purple is used for light rain.

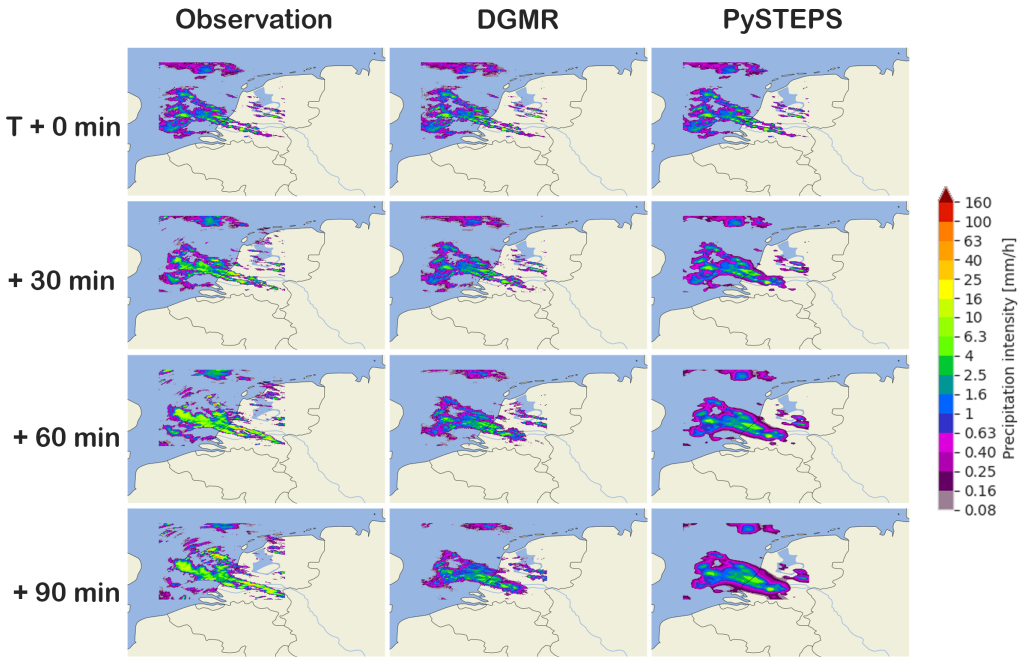

Figure 8. Predictions from both models and the corresponding ground truth for precipitation event starting at T=2021-08-22 at 10:00 UTC +1, showing a large structure of medium to heavy rainfall moving very slightly to the east over the north west of the Netherlands.

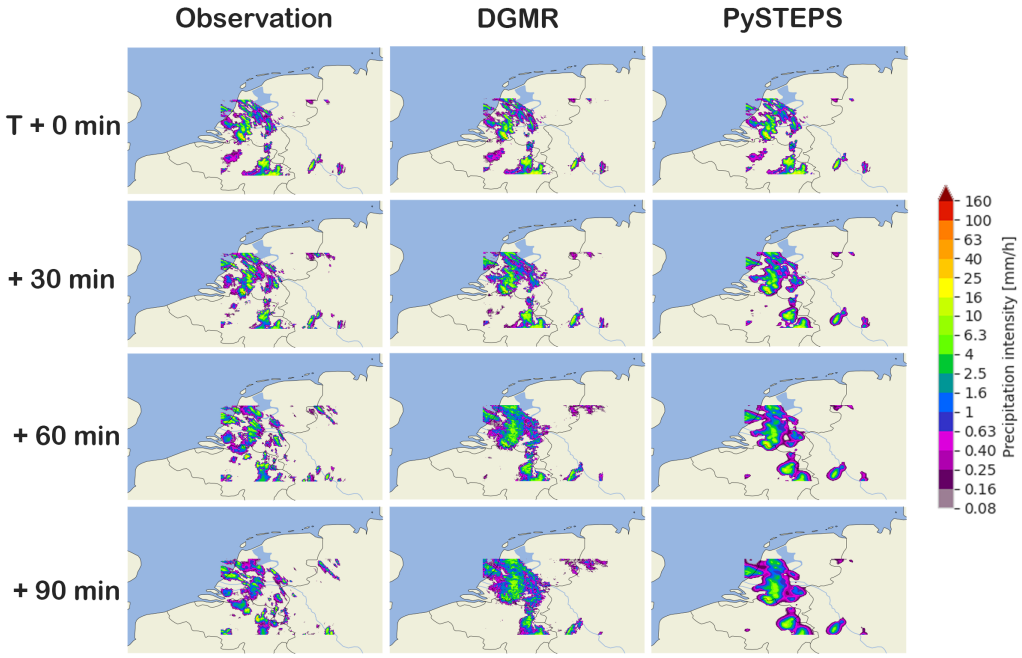

Figure 9. Predictions from both models and the corresponding ground truth for precipitation event starting at T=2021-07-26 at 14:00 UTC +1, showing multiple smaller structures rainfall over the centre of the Netherlands.

metrics used provided a solid assessment of the predictive qualities.

4.3 Further Studies

Due to the aforementioned limitations, there are multiple ways in which the current study can be improved upon. These could be alterations upon the conducted research or research on new topics arising from this research.

For one, this research could be repeated for different countries. Preferably researchers could use rain data from a country with a vastly different climate to that of the United Kingdom and the Netherlands. Given that generalisability found in this study could very well be a result of the similarity of climates between these two countries.

Furthermore, researchers could focus on variables in the data. This could include a study on the difference in predictive quality between events from different periods (summer/winter) or areas (north/south). This could also be an investigation into the predictions of extreme rain events.

5. Conclusion

In this chapter, we will answer the main research question. This will be done by first answering the three sub-questions.

The first sub-question asked if high-quality weather data available for the Netherlands that can be used with the pre-trained DGMR model. After pre-processing the raw radar images extracted from the RNN. We successfully made predictions for multiple rain events. It was found that the quality, and the characteristics, of the data, were sufficient for use in the model. Only the dimensions of the raw radar images were found to be inconsistent, however, this problem can be solved by re-training the model. Still, we conclude the available data to be of satisfactory quality.

The second sub-question *'Are alterations to the models' architecture needed for the prediction of rainfall in the Netherlands?'* was answered by an overview of the models' architecture and an evaluation of the results. It was found that the choices made by DeepMind to solve problems of earlier models, also saw positive effects in this study. However, the prediction still encountered some problems not identified by DeepMind. Most likely this is a result of not retraining the model and not from elements of the model itself. This claim is emphasised by the results of validation (consisting of both training and testing) by DeepMind on the United States. From this, we can conclude that the architecture in its current state can be directly implemented for use in the Netherlands, but re-training is needed.

Thirdly in this study, we tried to answer the question *'Are the same patterns noticeable between the results on Dutch*

weather data and the results found by DeepMind?' by analysis of the results on three different metrics. From this analysis, we can conclude that the results found are in line with those found by DeepMind. Here, the complex DGMR model equally outperformed the PySTEPS method and was able to mitigate problems of earlier nowcasting methods in a similar fashion.

With the use of the answers found for the sub-questions, we can answer the main research question of this thesis: *Can the pre-trained DeepMind precipitation model be used for successful precipitation nowcasting in the Netherlands?* This thesis showed that the pre-trained DGMR model introduced by Google DeepMind was able to predict precipitation events for 90 minutes into the future better than the PySTEPS method used as a baseline. Although this is an important step, the predictions still showed room for improvement, especially at long lead times. However, the intrigues and results of the model in combination with the availability and quality of the data give enough reason to assume that this model, after retraining, can be used for successful precipitation nowcasting in the Netherlands.

References

[1] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017.

[2] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[3] Erich M Fischer and Reto Knutti. Observed heavy precipitation increase confirms theory and early models. *Nature Climate Change*, 6(11):986–991, 2016.

[4] S Pulkkinen, Daniele Nerini, and Loris Foresti. pysteps: an open-source community for radar-based ensemble precipitation nowcasting. In *Rainfall Monitoring, Modelling and Forecasting in Urban Environment. UrbanRain18: 11th International Workshop on Precipitation in Urban Areas. Conference Proceedings*, pages 94–94. ETH Zurich, Institute of Environmental Engineering, 2019.

[5] Marc Berenguer Ferrer, Carles Corral Alexandri, Daniel Sempere Torres, and Alan W Seed. Validation of a nowcasting technique from a hydrological perspective. In *6th International Symposium on Hydrological Applications of Weather Radar*, pages 1–8, 2004.

[6] Koert Schreurs, Yuliya Shapovalova, Maurice Schmeits, Kiri Whan, and Tom Heskes. Precipitation nowcasting using generative adversarial networks. 2021.

[7] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

[8] MF Rios Gaona, Aart Overeem, Hidde Leijnse, and Remko Uijlenhoet. First-year evaluation of gpm rainfall over the netherlands: Imerg day 1 final run (v03d). *Journal of Hydrometeorology*, 17(11):2799–2814, 2016.

[9] Frank Kreienkamp, Sjoukje Y Philip, Jordis S Tradowsky, Sarah F Kew, Philip Lorenz, Julie Arrighi, Alexandre Belleflamme, Thomas Bettmann, Steven Caluwaerts, Steven C Chan, et al. Rapid attribution of heavy rainfall events leading to the severe flooding in western europe during july 2021. 2021.

[10] Hanneke Schuurmans and Jojanneke van Vossen. Nationale regenradar toelichting operationele neerslagproducten. 2013.

[11] Zhihan Gao, Xingjian Shi, Hao Wang, Dit-Yan Yeung, Wang-chun Woo, and Wai-Kin Wong. Deep learning and the weather forecasting problem: Precipitation nowcasting. *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, pages 218–239, 2021.

[12] Kevin Trebing, Tomasz Staczyk, and Siamak Mehrkanoon. Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. *Pattern Recognition Letters*, 145:178–186, 2021.

[13] Chuyao Luo, Xutao Li, Yunming Ye, Shanshan Feng, and Michael K Ng. Experimental study on generative adversarial network for precipitation nowcasting. *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[14] Gregor Skok and Nigel Roberts. Analysis of fractions skill score properties for random precipitation fields and ecmwf forecasts. *Quarterly Journal of the Royal Meteorological Society*, 142(700):2599–2610, 2016.

Appendix

A

Days included for radar image extraction based on measurements of rainfall in De Bilt, Netherlands.

Month	Days
January	7, 12, 19, 21, 28, 29
February	2, 3, 18
March	13, 16
April	7, 10, 29, 30
May	4, 13, 16, 17, 19, 22, 24, 26
June	18, 19, 21, 27
July	3, 4, 15, 25, 26, 27, 31
August	3, 7, 8, 9, 16, 21, 22
September	10, 27, 29
October	1, 2, 3, 6, 12, 20, 21, 30, 31
November	13, 26, 27, 30
December	1, 2, 6, 24, 25, 29

B

Overview of the used test set.

Month	Amount
January	11
February	5
March	3
April	7
May	15
June	7
July	9
August	8
September	6
October	18
November	3
December	8

Position	Amount
North West	23
North East	16
South West	31
South East	30

C

The data and Google Colab Notebook containing the code used for this research can be found here: Data and Code